Estruturas, plataformas, protocolos e ferramentas de IA da Agentic no AWS

AWS Recomendações

AWS Recomendações: Estruturas, plataformas, protocolos e ferramentas de IA da Agentic no AWS

Table of Contents

Estruturas, plataformas, protocolos e ferramentas de IA da Agentic no AWS

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

Janeiro de 2026 (histórico do documento)

A IA agente é um paradigma poderoso na interseção entre IA, sistemas distribuídos e engenharia de software. É uma classe de sistemas inteligentes composta por agentes de software autônomos e assíncronos que usam modelos de IA e se integram a ferramentas e recursos. Os agentes demonstram agência, podem perceber o contexto, raciocinar sobre metas, tomar decisões e realizar ações propositais em nome de usuários ou sistemas. Esses agentes operam de forma independente, geralmente de forma colaborativa, em ambientes distribuídos e são projetados para perseguir objetivos delegados com inteligência, memória e intenção incorporadas.

Além AWS disso, as organizações podem aproveitar a IA agente para automatizar fluxos de trabalho complexos, aprimorar os processos de tomada de decisão e criar sistemas mais responsivos. Este guia fornece informações sobre os principais componentes necessários para criar soluções eficazes de IA agêntica:

- O Frameworks traça o perfil das estruturas atuais de IA para agentes, incluindo análises de seus benefícios e casos de uso. Saiba como essas estruturas reduzem o trabalho pesado indiferenciado entre padrões, protocolos e ferramentas. Entenda os principais critérios de seleção para escolher a estrutura certa para suas necessidades.
- As plataformas fornecem uma visão geral das plataformas de IA agentes (agente gerenciado, orquestração de código aberto e híbrida) e considerações para seleção ou design.
- Protocolos explora protocolos de comunicação padronizados essenciais para interações de agentes. Agent-to-agentprotocolos estão surgindo, como o Model Context Protocol (MCP) e o Agent2Agent (A2A) de código aberto, junto com outras implementações proprietárias. Descubra como protocolos comuns permitem que diferentes protocolos interajam perfeitamente.
- As ferramentas fornecem informações sobre ferramentas baseadas em protocolos (como o MCP), ferramentas nativas da estrutura e meta-ferramentas. As organizações podem criar um kit de ferramentas que se integre aos principais sistemas em seus fluxos de trabalho, permitindo fluxos de trabalho agentes baseados no usuário final e no servidor.

Público-alvo

Este guia é para arquitetos, desenvolvedores e líderes de tecnologia que buscam aproveitar o poder dos agentes de software orientados por IA em aplicativos modernos nativos da nuvem.

Objetivos

Este guia ajuda você a:

- Compare diferentes estruturas de IA agente para selecionar a mais adequada para seu caso de uso.
- Saiba mais sobre plataformas de IA agênticas que fornecem recursos para transformar agentes individuais em sistemas coordenados e adaptáveis.
- Entenda as vantagens dos protocolos abertos para criar arquiteturas sustentáveis de IA agente.
- Crie uma estratégia apropriada de integração de ferramentas ao criar sistemas de agentes.

Sobre esta série de conteúdo

Este guia faz parte de uma série sobre IA agente em AWS. Para obter mais informações e ver os outros guias desta série, consulte Agentic AI no site da AWS Prescriptive Guidance.

Estruturas

O Foundations of Agentic AI on AWS examina os principais padrões e fluxos de trabalho que permitem um comportamento autônomo e direcionado a objetivos. No centro da implementação desses padrões está a escolha da estrutura. Uma estrutura é a base de software do código pré-escrito que fornece um ambiente estruturado e funcionalidade comum para criar e gerenciar as ferramentas e os recursos de orquestração necessários para criar agentes de IA autônomos prontos para produção.

Estruturas de IA agênticas eficazes fornecem vários recursos essenciais que transformam as interações brutas do modelo de linguagem grande (LLM) em sistemas coordenados e inteligentes capazes de raciocinar, colaborar e agir:

- A orquestração de agentes coordena o fluxo de informações e a tomada de decisões em um ou vários agentes para atingir metas complexas sem intervenção humana.
- A integração de ferramentas permite que os agentes interajam com sistemas externos, APIs e fontes de dados para ampliar seus recursos além do processamento de linguagem. Para obter mais informações, consulte Visão geral das ferramentas na Strands Agents documentação.
- O gerenciamento de memória fornece um estado persistente ou baseado em sessão para manter o contexto em todas as interações, essencial para tarefas adaptáveis ou de longa duração. Estruturas mais avançadas incorporam memória de longo prazo para armazenar resumos e preferências do usuário, permitindo experiências agentes personalizadas e contextualmente conscientes. Para obter mais informações, consulte Como pensar sobre estruturas de agentes no LangChain blog.
- A definição do fluxo de trabalho suporta padrões estruturados, como cadeias, roteamento, paralelização e ciclos de reflexão, que permitem um raciocínio autônomo sofisticado.
- A implantação e o monitoramento facilitam a transição do desenvolvimento para a produção com observabilidade para sistemas autônomos. Para obter mais informações, consulte o anúncio de disponibilidade AgentCore geral do Amazon Bedrock.

Esses recursos são implementados com abordagens e ênfases variadas em todo o cenário da estrutura, cada uma oferecendo vantagens distintas para diferentes casos de uso de agentes autônomos e contextos organizacionais.

Esta seção traça o perfil e compara as principais estruturas para criar soluções de IA agênticas, com foco em seus pontos fortes, limitações e casos de uso ideais para operação autônoma:

- Agentes de filamentos
- LangChain and LangGraph
- Tripulação AI
- AutoGen
- ???
- Comparando estruturas de IA agênticas

> **Note**
>
> Esta seção aborda as estruturas que apoiam especificamente a agência da IA e não abrange interfaces de front-end ou IA generativa sem agência.

Strands Agents

Strands Agentsé um SDK de código aberto que foi lançado inicialmente pela AWS, conforme descrito no AWS Open Source Blog. Strands Agentsfoi projetado para criar agentes de IA autônomos com uma abordagem que prioriza o modelo. Ele fornece uma estrutura flexível e extensível, projetada para funcionar perfeitamente e, ao Serviços da AWS mesmo tempo, permanecer aberta à integração com componentes de terceiros. O Strands Agents é ideal para criar soluções totalmente autônomas.

Principais características do Strands Agents

Strands Agentsinclui os seguintes recursos principais:

- Model-first design — Construído com base no conceito de que o modelo básico é o núcleo da inteligência do agente, permitindo um raciocínio autônomo sofisticado. Para obter mais informações, consulte Agent Loop na Strands Agents documentação.
- Multi-agent padrões de colaboração — modelos de Built-in coordenação, como padrões Swarm, Graph e Workflow, que permitem colaboração e governança escaláveis em redes de agentes distribuídos. Para obter mais informações, consulte Multi-agent Padrões na documentação do Strands Agents.
- Integração MCP — Suporte nativo para o Model Context Protocol (MCP), permitindo o fornecimento de contexto padronizado para LLMs para operação autônoma consistente.

- AWS service (Serviço da AWS) integração — conexão perfeita com Amazon Bedrock,, AWS Lambda AWS Step Functions, e outros Serviços da AWS para fluxos de trabalho autônomos abrangentes. Para obter mais informações, consulte Resumo AWS semanal (AWS blog).
- Seleção de modelos básicos — Suporta vários modelos básicos, incluindo Anthropic Claude, Amazon Nova (Premier, Pro, Lite e Micro) no Amazon Bedrock e outros, para otimizar diferentes capacidades de raciocínio autônomo. Para obter mais informações, consulte Amazon Bedrock na Strands Agents documentação.
- Integração da API LLM — Integração flexível com diferentes interfaces de serviço LLM, incluindo Amazon Bedrock, OpenAI e outras, para implantação em produção. Para obter mais informações, consulte Uso básico do Amazon Bedrock na Strands Agents documentação.
- Capacidades multimodais — Support para várias modalidades, incluindo processamento de texto, fala e imagem para interações abrangentes com agentes autônomos. Para obter mais informações, consulte Amazon Bedrock Multimodal Support na Strands Agents documentação.
- Ecossistema de ferramentas — Conjunto rico de ferramentas para AWS service (Serviço da AWS) interação, com extensibilidade para ferramentas personalizadas que expandem as capacidades autônomas. Para obter mais informações, consulte Visão geral das ferramentas na Strands Agents documentação.

Quando utilizar Strands Agents

Strands Agentsé particularmente adequado para cenários de agentes autônomos, incluindo:

- Organizações que se baseiam em uma AWS infraestrutura que desejam integração nativa com fluxos Serviços da AWS de trabalho autônomos
- Equipes que exigem recursos de segurança, escalabilidade e conformidade de nível empresarial para sistemas autônomos de produção
- Projetos que precisam de flexibilidade na seleção de modelos em diferentes fornecedores para tarefas autônomas especializadas
- Casos de uso que exigem forte integração com AWS fluxos de trabalho e recursos existentes para processos autônomos de ponta a ponta

Abordagem de implementação para Strands Agents

Strands Agentsfornece uma abordagem de implementação direta para as partes interessadas da empresa, conforme descrito em seu Guia de Início Rápido para Python e Guia de Início Rápido para. Typescript A estrutura permite que as organizações:

- Selecione modelos básicos como o Amazon Nova (Premier, Pro, Lite ou Micro) no Amazon Bedrock com base em requisitos comerciais específicos.
- Defina ferramentas personalizadas que se conectem a sistemas corporativos e fontes de dados.
- Processe várias modalidades, incluindo texto, imagens e fala.
- Implante agentes que possam responder de forma autônoma às consultas comerciais e realizar tarefas.

Essa abordagem de implementação permite que as equipes de negócios desenvolvam e implantem rapidamente agentes autônomos sem profundo conhecimento técnico no desenvolvimento de modelos de IA.

Real-world exemplo de Strands Agents

AWS Transform for .NET usa Strands Agents para potencializar seus recursos de modernização de aplicativos, conforme descrito em AWS Transform for .NET, o primeiro serviço de IA agente para modernizar aplicativos.NET em grande escala (AWS Blog). Este serviço de produção emprega vários agentes autônomos especializados. Os agentes trabalham juntos para analisar aplicativos.NET legados, planejar estratégias de modernização e executar transformações de código em arquiteturas nativas da nuvem sem intervenção humana. AWS Transform for .NET demonstra a prontidão de produção de Strands Agents sistemas autônomos corporativos.

LangChain e LangGraph

LangChainé uma das estruturas mais estabelecidas no ecossistema de IA agente. LangGraphamplia seus recursos para oferecer suporte a fluxos de trabalho de agentes complexos e dinâmicos, conforme descrito no LangChain Blog. Juntos, eles fornecem uma solução abrangente para criar agentes de IA autônomos sofisticados com recursos avançados de orquestração para operação independente.

Principais características de LangChain e LangGraph

LangChaine LangGraph inclua os seguintes recursos principais:

- Ecossistema de componentes — Ampla biblioteca de componentes pré-construídos para vários recursos de agentes autônomos, permitindo o rápido desenvolvimento de agentes especializados. Para obter mais informações, consulte Início rápido na LangChain documentação.
- Seleção de modelos básicos — Suporte para diversos modelos de fundação, incluindo modelos Anthropic Claude, Amazon Nova (Premier, Pro, Lite e Micro) no Amazon Bedrock e outros para diferentes capacidades de raciocínio. Para obter mais informações, consulte Entradas e saídas na LangChain documentação.
- Integração da API LLM — Interfaces padronizadas para vários provedores de serviços de modelo de linguagem grande (LLM), incluindo Amazon Bedrock e outrosOpenAI, para implantação flexível. Para obter mais informações, consulte LLMs na LangChain documentação.
- Processamento multimodal — Built-in suporte para processamento de texto, imagem e áudio para permitir interações ricas com agentes autônomos multimodais. Para obter mais informações, consulte Multimodalidade na LangChain documentação.
- Graph-based fluxos de trabalho — LangGraph permite definir comportamentos complexos de agentes autônomos como máquinas de estado, suportando uma lógica de decisão sofisticada. Para obter mais informações, consulte o anúncio da LangGraphPlatform GA.
- Abstrações de memória — Várias opções para gerenciamento de memória de curto e longo prazo, o que é essencial para agentes autônomos que mantêm o contexto ao longo do tempo. Para obter mais informações, consulte Como adicionar memória aos chatbots na LangChain documentação.
- Integração de ferramentas — Ecossistema rico de integrações de ferramentas em vários serviços e APIs, ampliando as capacidades dos agentes autônomos. Para obter mais informações, consulte Ferramentas na LangChain documentação.
- LangGraph plataforma — Solução gerenciada de implantação e monitoramento para ambientes de produção, oferecendo suporte a agentes autônomos de longa duração. Para obter mais informações, consulte o anúncio da LangGraphPlatform GA.

Quando usar LangChain e LangGraph

LangChaine LangGraph são particularmente adequados para cenários de agentes autônomos, incluindo:

- Fluxos de trabalho complexos de raciocínio em várias etapas que exigem orquestração sofisticada para tomada de decisão autônoma
- Projetos que precisam de acesso a um grande ecossistema de componentes e integrações pré-construídos para diversas capacidades autônomas
- Equipes com infraestrutura e experiência Python em aprendizado de máquina (ML) existentes que desejam criar sistemas autônomos
- Casos de uso que exigem gerenciamento complexo de estados em sessões de agentes autônomos de longa duração

Abordagem de implementação para LangChain e LangGraph

LangChaine LangGraph forneça uma abordagem de implementação estruturada para as partes interessadas da empresa, conforme detalhado na LangGraphdocumentação. A estrutura permite que as organizações:

- Defina gráficos de fluxo de trabalho sofisticados que representem os processos de negócios.
- Crie padrões de raciocínio em várias etapas com pontos de decisão e lógica condicional.
- Integre recursos de processamento multimodal para lidar com diversos tipos de dados.
- Implemente o controle de qualidade por meio de mecanismos integrados de revisão e validação.

Essa abordagem baseada em gráficos permite que as equipes de negócios modelem processos de decisão complexos como fluxos de trabalho autônomos. As equipes têm uma visibilidade clara de cada etapa do processo de raciocínio e a capacidade de auditar os caminhos de decisão.

Real-world exemplo de LangChain e LangGraph

Vodafoneimplementou agentes autônomos usando LangChain (eLangGraph) para aprimorar seus fluxos de trabalho de engenharia de dados e operações, conforme detalhado em seu estudo de caso LangChain corporativo. Eles criaram assistentes internos de IA que monitoram de forma autônoma as métricas de desempenho, recuperam informações dos sistemas de documentação e apresentam insights acionáveis, tudo por meio de interações em linguagem natural.

A Vodafone implementação usa carregadores LangChain modulares de documentos, integração vetorial e suporte para vários LLMs (OpenAI, LLaMA 3 eGemini) para prototipar e comparar rapidamente esses pipelines. Em seguida, LangGraph costumavam estruturar a orquestração de vários agentes implantando subagentes modulares. Esses agentes realizam tarefas de coleta,

processamento, resumo e raciocínio. LangGraphintegraram esses agentes por meio de APIs em seus sistemas em nuvem.

CrewAI

CrewAIé uma estrutura de código aberto focada especificamente na orquestração autônoma de vários agentes, disponível em. GitHub Ele fornece uma abordagem estruturada para criar equipes de agentes autônomos especializados que colaboram para resolver tarefas complexas sem intervenção humana. CrewAIenfatiza a coordenação baseada em funções e a delegação de tarefas.

Principais características do CrewAI

CrewAIfornece os seguintes recursos principais:

- Role-based design de agentes — Agentes autônomos são definidos com funções, metas e histórias de fundo específicas para permitir conhecimentos especializados. Para obter mais informações, consulte Criação de agentes eficazes na CrewAI documentação.
- Delegação de tarefas — Built-in mecanismos para atribuir tarefas de forma autônoma aos agentes apropriados com base em suas capacidades. Para obter mais informações, consulte Tarefas na CrewAI documentação.
- Colaboração de agentes — Estrutura para comunicação autônoma entre agentes e compartilhamento de conhecimento sem mediação humana. Para obter mais informações, consulte Colaboração na CrewAI documentação.
- Gerenciamento de processos — fluxos de trabalho estruturados para execução sequencial e paralela de tarefas autônomas. Para obter mais informações, consulte Processos na CrewAI documentação.
- Seleção de modelos básicos — Suporte para vários modelos básicos, incluindo modelos Anthropic Claude, Amazon Nova (Premier, Pro, Lite e Micro) no Amazon Bedrock e outros para otimizar diferentes tarefas de raciocínio autônomo. Para obter mais informações, consulte LLMs na CrewAI documentação.
- Integração da API LLM — Integração flexível com várias interfaces de serviço LLM, OpenAI incluindo Amazon Bedrock e implantações de modelos locais. Para obter mais informações, consulte Exemplos de configuração do provedor na CrewAI documentação.
- Suporte multimodal — Capacidades emergentes para lidar com texto, imagem e outras modalidades para interações abrangentes de agentes autônomos. Para obter mais informações, consulte Usando agentes multimodais na CrewAI documentação.

Quando utilizar CrewAI

CrewAIé particularmente adequado para cenários de agentes autônomos, incluindo:

- Problemas complexos que se beneficiam da experiência especializada e baseada em funções trabalhando de forma autônoma
- Projetos que exigem colaboração explícita entre vários agentes autônomos
- Casos de uso em que a decomposição de problemas baseada em equipe melhora a resolução autônoma de problemas
- Cenários que exigem uma separação clara de preocupações entre diferentes funções de agentes autônomos

Abordagem de implementação para CrewAI

CrewAIfornece uma implementação baseada em funções da abordagem de equipes de agentes de IA para as partes interessadas da empresa, conforme detalhado em Introdução na CrewAI documentação. A estrutura permite que as organizações:

- Defina agentes autônomos especializados com funções, metas e áreas de especialização específicas.
- Atribua tarefas aos agentes com base em suas capacidades especializadas.
- Estabeleça dependências claras entre as tarefas para criar fluxos de trabalho estruturados.
- Organize a colaboração entre vários agentes para resolver problemas complexos.

Essa abordagem baseada em funções reflete as estruturas de equipes humanas, tornando-a intuitiva para os líderes de negócios entenderem e implementarem. As organizações podem criar equipes autônomas com áreas de especialização especializadas que colaboram para alcançar os objetivos de negócios, da mesma forma que as equipes humanas operam. No entanto, a equipe autônoma pode trabalhar continuamente sem intervenção humana.

Real-world exemplo de CrewAI

AWS implementou sistemas multiagentes autônomos usando o CrewAI integrado ao Amazon Bedrock, conforme detalhado no estudo de caso publicado. CrewAI AWS e CrewAI desenvolveu uma estrutura segura e neutra em relação ao fornecedor. A arquitetura “CrewAIflows-and-crews” de

código aberto se integra perfeitamente aos modelos básicos, sistemas de memória e barreiras de conformidade do Amazon Bedrock.

Os principais elementos da implementação incluem:

- Planos e código aberto — AWS e designs de referência CrewAI lançados que mapeiam CrewAI agentes para modelos e ferramentas de observabilidade do Amazon Bedrock. Eles também lançaram sistemas exemplares, como uma equipe de auditoria de AWS segurança multiagente, fluxos de modernização de código e automação de back-office de bens de consumo embalados (CPG).
- Integração da pilha de observabilidade — A solução incorpora monitoramento com a Amazon e CloudWatch, AgentOps permitindo a rastreabilidade e a LangFuse depuração, desde a prova de conceito até a produção.
- Retorno sobre o investimento (ROI) demonstrado — Os primeiros pilotos mostram grandes melhorias — execução 70% mais rápida para um grande projeto de modernização de código e cerca de 90% de redução no tempo de processamento de um fluxo de back-office de CPG.

AutoGen

AutoGené uma estrutura de código aberto que foi lançada inicialmente pelaMicrosoft. AutoGenconcentra-se em capacitar agentes de IA autônomos conversacionais e colaborativos. Ele fornece uma arquitetura flexível para criar sistemas multiagentes com ênfase em interações assíncronas e orientadas por eventos entre agentes para fluxos de trabalho autônomos complexos.

Principais características do AutoGen

AutoGenfornece os seguintes recursos principais:

- Agentes conversacionais — Construídos em torno de conversas em linguagem natural entre agentes autônomos, permitindo um raciocínio sofisticado por meio do diálogo. Para obter mais informações, consulte Estrutura de Multi-agent conversação na AutoGen documentação.
- Arquitetura assíncrona — Event-driven design para interações de agentes autônomos sem bloqueio, suportando fluxos de trabalho paralelos complexos. Para obter mais informações, consulte Resolvendo várias tarefas em uma sequência de bate-papos assíncronos na documentação. AutoGen

- Human-in-the-loop— Forte suporte à participação humana opcional em fluxos de trabalho de agentes autônomos, quando necessário. Para obter mais informações, consulte Permitir feedback humano em agentes na AutoGen documentação.
- Geração e execução de código — Recursos especializados para agentes autônomos focados em código que podem escrever e executar código. Para obter mais informações, consulte Execução de código na AutoGen documentação.
- Comportamentos personalizáveis — Configuração flexível de agentes autônomos e controle de conversação para diversos casos de uso. Para obter mais informações, consulte agentchat.conversable_agent na documentação. AutoGen
- Seleção de modelos básicos — Suporte para vários modelos básicos, incluindo modelos Anthropic Claude, Amazon Nova (Premier, Pro, Lite e Micro) no Amazon Bedrock e outros para diferentes capacidades de raciocínio autônomo. Para obter mais informações, consulte Configuração do LLM na AutoGen documentação.
- Integração da API LLM — Configuração padronizada para várias interfaces de serviço LLM, incluindo Amazon Bedrock, e. OpenAI Azure OpenAI Para obter mais informações, consulte oai.openai_utils na Referência da API. AutoGen
- Processamento multimodal — Support para processamento de texto e imagem para permitir interações ricas com agentes autônomos multimodais. Para obter mais informações, consulte Envolvendo-se com modelos multimodais: GPT-4V AutoGen na AutoGen documentação.

Quando utilizar AutoGen

AutoGené particularmente adequado para cenários de agentes autônomos, incluindo:

- Aplicativos que exigem fluxos conversacionais naturais entre agentes autônomos para raciocínio complexo
- Projetos que precisam de operação totalmente autônoma e recursos opcionais de supervisão humana
- Casos de uso que envolvem geração, execução e depuração autônomas de código sem intervenção humana
- Cenários que exigem padrões de comunicação de agentes autônomos flexíveis e assíncronos

Abordagem de implementação para AutoGen

AutoGenfornece uma abordagem de implementação conversacional para as partes interessadas da empresa, conforme detalhado em Introdução na AutoGen documentação. A estrutura permite que as organizações:

- Crie agentes autônomos que se comunicam por meio de conversas em linguagem natural.
- Implemente interações assíncronas e orientadas por eventos entre vários agentes.
- Combine operação totalmente autônoma com supervisão humana opcional quando necessário.
- Desenvolva agentes especializados para diferentes funções de negócios que colaborem por meio do diálogo.

Essa abordagem conversacional torna o raciocínio do sistema autônomo transparente e acessível aos usuários corporativos. Decision-makers pode observar o diálogo entre os agentes para entender como as conclusões são alcançadas e, opcionalmente, participar da conversa quando o julgamento humano é necessário.

Real-world exemplo de AutoGen

Magentic-Oneé um sistema multiagente generalista e de código aberto projetado para resolver de forma autônoma tarefas complexas de várias etapas em diversos ambientes, conforme descrito no blog AI Frontiers. Microsoft Em sua essência, está o agente Orchestrator, que decompõe metas de alto nível e acompanha o progresso usando livros contábeis estruturados. Esse agente delega subtarefas a agentes especializados (comoWebSurfer, FileSurferCoder, eComputerTerminal) e se adapta dinamicamente replanejando quando necessário.

O sistema é baseado na AutoGen estrutura e é independente do modelo, usando como padrão o GPT-4o. Ele alcança desempenho de última geração em benchmarks como, e —tudo sem ajustes específicos da tarefa. GAIA AssistantBench WebArena Além disso, ele oferece suporte à extensibilidade modular e à avaliação rigorosa por meio de sugestões. AutoGenBench

LlamaIndex

LlamaIndexé uma estrutura de dados projetada especificamente para conectar modelos de linguagem grande (LLMs) a fontes de dados externas para permitir sofisticados aplicativos de geração aumentada de recuperação (RAG) e inteligência artificial. A estrutura fornece abstrações e fluxos de trabalho de desenvolvimento acelerados para sistemas agentes, padrões de orquestração

personalizados e integrações de sistemas que reduzem o tempo de produção de soluções de IA orientadas pelo conhecimento.

Principais características do LlamaIndex

LlamaIndexfornece um conjunto abrangente de recursos que o torna particularmente adequado para aplicativos de IA de agentes corporativos:

- Data-centric arquitetura — se destaca na ingestão, indexação e recuperação de informações de mais de 100 formatos de dados, incluindo PDFs, documentos do Microsoft Word, planilhas e muito mais. A estrutura transforma dados corporativos em bases de conhecimento consultáveis que são otimizadas para agentes de IA. Para obter mais informações, consulte a documentação do LlamaIndex.
- Production-ready implantação — LlamaIndex oferece estruturas de código aberto e serviços gerenciadosLlamaCloud, fornecendo recursos de nível corporativo, incluindo controles de segurança, escalabilidade, integrações de observabilidade e flexibilidade de implantação. Para obter mais informações, consulte a documentação da LlamaIndex estrutura.
- Processamento avançado de documentos — LlamaCloud fornece recursos de análise, extração, indexação e recuperação de documentos que lidam com layouts complexos, tabelas aninhadas, conteúdo multimodal e até anotações manuscritas. Essa análise sofisticada permite que os agentes trabalhem de forma eficaz com documentos corporativos reais que contêm gráficos, diagramas e formatação complexa. Para obter mais informações, consulte a documentação do LlamaCloud.
- Orquestração de fluxos de trabalho — LlamaAgents fornece um mecanismo de orquestração assíncrono e orientado por eventos para criar sistemas agentes de várias etapas. Os fluxos de trabalho oferecem suporte a padrões complexos, incluindo loops, execução paralela, ramificação condicional e retomada com estado, o que os torna ideais para interações sofisticadas com agentes. Para obter mais informações, consulte a documentação dos LlamaIndex fluxos de trabalho.
- Capacidades de recuperação agente — modos de recuperação avançados, incluindo pesquisa híbrida, pesquisa semântica e roteamento automático que determinam de forma inteligente a melhor estratégia de recuperação para cada consulta. A estrutura oferece suporte à recuperação composta em várias bases de conhecimento com reclassificação para maior precisão. Para obter mais informações, consulte a documentação do LlamaIndex RAG.
- Observabilidade e avaliação — LlamaIndex integra-se a uma variedade de ferramentas de observabilidade e avaliação. Esse recurso de integração ajuda você a rastrear e depurar seus

aplicativos, avaliar seu desempenho e monitorar os custos. Para obter mais informações, consulte a documentação de rastreamento, depuração e avaliação. LlamaIndex

Quando utilizar LlamaIndex

LlamaIndexé particularmente adequado para cenários de IA agentes que enfatizam fluxos de trabalho intensivos em dados e gerenciamento de conhecimento:

- Document-heavy aplicativos que exigem que os agentes processem, analisem e extraiam insights de grandes volumes de documentos corporativos, como contratos, relatórios, manuais e registros regulatórios
- Prototipagem rápida para cenários de produção em que as organizações desejam criar e implantar rapidamente agentes centrados em documentos sem sobrecarga de gerenciamento de infraestrutura
- RAG-first arquiteturas que priorizam a precisão da recuperação e a relevância do contexto, especialmente ao trabalhar com documentos complexos e multimodais contendo tabelas, imagens e dados estruturados
- Multi-agent fluxos de trabalho de documentos que exigem agentes especializados para diferentes aspectos do processamento de documentos, como análise, análise, resumo e verificação de conformidade

Abordagem de implementação para LlamaIndex

LlamaIndex fornece blocos de construção de baixo nível e abstrações de alto nível que acomodam diferentes abordagens de implementação:

- Desenvolvimento rápido de aplicativos RAG funcionais em apenas algumas linhas de código usando APIs de LlamaIndex alto nível. Essa abordagem torna LlamaIndex acessível para equipes de negócios e desenvolvedores que são novos na IA agente.
- Integração empresarial LlamaHub por meio de sistemas corporativos populares SharePoint, incluindo Amazon Simple Storage Service (Amazon S3), bancos de dados e APIs. Essa abordagem permite uma integração perfeita com a infraestrutura de dados existente.
- Opções flexíveis de implantação entre implantações auto-hospedadas de código aberto para controle máximo ou serviços LlamaCloud gerenciados para reduzir a sobrecarga operacional e os recursos corporativos.

- Os aplicativos podem começar com mecanismos de consulta simples e adicionar progressivamente recursos agentes, orquestração de vários agentes e fluxos de trabalho complexos à medida que os requisitos evoluem.

Real-world exemplo de LlamaIndex

Este exemplo se concentra em uma subsidiária de uma empresa aeroespacial especializada em soluções de navegação e operações de aviação. Eles precisam enfrentar um desafio crescente que envolve pilotar testes descoordenados de chatbots de IA. Os testes resultaram em trabalho repetido, longos ciclos de desenvolvimento, obstáculos de conformidade e implementações isoladas em toda a organização.

Eles desenvolveram uma estrutura de agente unificada, uma solução reutilizável baseada em modelos criada na estrutura de LlamaIndex código aberto que torna a criação de agentes muito mais eficiente. Eles compararam várias estruturas concorrentes, tanto orientadas por cadeias quanto baseadas em gráficos. Em última análise, eles LlamaIndex escolheram três vantagens essenciais: seu design flexível, componentes modulares e controles de orquestração prontos para produção.

A plataforma reduz o tempo de desenvolvimento e implantação do agente em 87%, de 512 para 64 horas. Essa redução foi alcançada ao permitir que as equipes criassem agentes com aproximadamente 50 linhas de código e um arquivo de configuração JSON. As equipes utilizaram uma estrutura unificada com segurança integrada, conformidade e acesso privilegiado ao sistema. Para obter mais detalhes, consulte os estudos de caso de LlamaIndex clientes.

Comparando estruturas de IA agênticas

Ao selecionar uma estrutura de IA agente para o desenvolvimento de agentes autônomos, considere como cada opção se alinha aos seus requisitos específicos. Considere não apenas suas capacidades técnicas, mas também sua adequação organizacional, incluindo a experiência da equipe, a infraestrutura existente e os requisitos de manutenção de longo prazo. Muitas organizações podem se beneficiar de uma abordagem híbrida, aproveitando várias estruturas para diferentes componentes de seu ecossistema autônomo de IA.

A tabela a seguir compara os níveis de maturidade (mais forte, forte, adequado ou fraco) de cada estrutura nas principais dimensões técnicas. Para cada estrutura, a tabela também inclui informações sobre as opções de implantação de produção e a complexidade da curva de aprendizado.

Framework	AWS integração	Suporte autônomo a vários agentes	complexidade do fluxo de trabalho autônomo	Capacidades multimodais	Seleção do modelo de fundação	Integração da API LLM	Implantação de produção	Curva de aprendizado
AutoGen	Fraco	Forte	Forte	Adequado	Adequado	Forte	Faça você mesmo (DIY)	Íngreme
CrewAI	Fraco	Forte	Adequado	Fraco	Adequado	Adequado	FAÇA VOCÊ MESMO	Moderada
LangChain / LangGraph	Adequado	Forte	Mais forte	Mais forte	Mais forte	Mais forte	Plataforma ou faça você mesmo	Íngreme
LlamaIndex	Adequado	Adequado	Forte	Adequado	Forte	Forte	Plataforma ou faça você mesmo	Moderada
Strands Agents	Mais forte	Forte	Mais forte	Forte	Forte	Mais forte	FAÇA VOCÊ MESMO	Moderada

Considerações na escolha de uma estrutura de IA agente

Ao desenvolver agentes autônomos, considere os seguintes fatores-chave:

- AWS integração de infraestrutura — As organizações em que investem fortemente se AWS beneficiarão mais das integrações nativas do Strands Agents with Serviços da AWS para fluxos de trabalho autônomos. Para obter mais informações, consulte Resumo AWS semanal (AWS blog).
- Seleção do modelo de fundação — Considere qual estrutura fornece o melhor suporte para seus modelos de fundação preferidos (por exemplo, modelos Amazon Nova no Amazon Bedrock ou Anthropic Claude), com base nos requisitos de raciocínio do seu agente autônomo. Para obter mais informações, consulte Criação de agentes eficazes no Anthropic site.
- Integração da API LLM — Avalie as estruturas com base em sua integração com suas interfaces de serviço preferidas de modelo de linguagem grande (LLM) (por exemplo, Amazon Bedrock ouOpenAI) para implantação em produção. Para obter mais informações, consulte Interfaces de modelo na Strands Agents documentação.
- Requisitos multimodais — Para agentes autônomos que precisam processar texto, imagens e fala, considere os recursos multimodais de cada estrutura. Para obter mais informações, consulte Multimodalidade na LangChain documentação.
- Complexidade do fluxo de trabalho autônomo — Fluxos de trabalho autônomos mais complexos com gerenciamento de estado sofisticado podem favorecer os recursos avançados da máquina de estado. de. LangGraph
- Colaboração autônoma em equipe — Projetos que exigem colaboração autônoma explícita baseada em funções entre agentes especializados podem se beneficiar da arquitetura orientada à equipe do. CrewAI
- Paradigma de desenvolvimento autônomo — equipes que preferem padrões conversacionais e assíncronos para agentes autônomos podem preferir a arquitetura orientada a eventos do. AutoGen
- Abordagem gerenciada ou baseada em código — Organizações que desejam uma experiência totalmente gerenciada com o mínimo de codificação devem considerar os Amazon Bedrock Agents. Organizações que exigem uma personalização mais profunda podem preferir Strands Agents outras estruturas com recursos especializados que se alinhem melhor aos requisitos específicos de agentes autônomos.
- Prontidão de produção para sistemas autônomos — considere as opções de implantação, os recursos de monitoramento e os recursos corporativos para agentes autônomos de produção.

Plataformas

As plataformas de IA da Agentic fornecem as camadas básicas de tempo de execução, orquestração e integração necessárias para implantar, escalar e gerenciar sistemas agentes de nível de produção. As estruturas definem como os agentes são criados e os protocolos governam a forma como eles se comunicam. As plataformas fornecem o ambiente em que esses agentes operam, colaboram e evoluem com segurança em grande escala.

As plataformas Agentic combinam recursos de execução de modelos, gerenciamento de contexto, integração de ferramentas, observabilidade e governança em ambientes unificados. Essas plataformas permitem que as organizações passem da experimentação para a implantação em escala corporativa.

Nesta seção:

Por que as plataformas são importantes

As plataformas de IA da Agentic são essenciais para organizações que buscam operacionalizar sistemas autônomos na produção. Eles oferecem os seguintes recursos:

- Forneça orquestração de tempo de execução para hospedagem, escalabilidade e coordenação de agentes.
- Gerencie o estado, o contexto e a memória em fluxos de trabalho de vários agentes.
- Ofereça controles de segurança, identidade e governança alinhados aos padrões corporativos.
- Integre-se com ecossistemas de ferramentas e sistemas externos por meio de padrões APIs ou protocolos.
- Permita a observabilidade e a auditabilidade nas interações dos agentes e nos fluxos de eventos.
- Support a interoperabilidade entre modelos, permitindo que os agentes usem vários modelos básicos em um único ambiente.

Esses recursos transformam agentes individuais em sistemas coordenados e adaptáveis que podem operar de forma confiável dentro dos limites corporativos e regulatórios.

Tipos de plataformas de IA agênticas

As plataformas de IA da Agentic geralmente se enquadram em uma ou mais das seguintes categorias:

- Agente gerenciado — plataformas totalmente gerenciadas fornecem recursos integrados de infraestrutura, memória e orquestração. Eles reduzem a sobrecarga operacional e aceleram o tempo de produção.
- Orquestração de código aberto — As plataformas agentes de código aberto oferecem flexibilidade e transparência para organizações que preferem ambientes personalizáveis ou implantação local.
- Empresa híbrida — As plataformas híbridas integram componentes gerenciados e auto-hospedados, combinando a escalabilidade dos serviços gerenciados na nuvem com o controle dos sistemas corporativos.

Considerações sobre a seleção da plataforma

Ao selecionar ou projetar uma plataforma de IA agente, as organizações devem considerar o seguinte:

- Profundidade de integração — Avalie o quão bem a plataforma se integra às fontes de dados, ferramentas e protocolos existentes.
- Escalabilidade — Garanta que a plataforma possa ser escalada dinamicamente para suportar cargas de trabalho autônomas e colaboração entre vários agentes.
- Segurança e conformidade — avalie os recursos de privacidade, criptografia e governança de dados em relação aos requisitos organizacionais e regionais.
- Extensibilidade — escolha plataformas com arquiteturas modulares que permitam que novas ferramentas, modelos ou agentes sejam adicionados ao longo do tempo.
- Observabilidade — prefira plataformas que forneçam registros detalhados de telemetria, rastreabilidade e auditoria para interações com agentes.
- Eficiência de custos — considere modelos sem servidor ou baseados em uso para otimizar o custo de cargas de trabalho variáveis.

Amazon Bedrock Agents

O Amazon Bedrock Agents é um serviço totalmente gerenciado que permite criar e configurar agentes autônomos em seus aplicativos. Ele pode orquestrar interações entre modelos básicos, fontes de dados, aplicativos de software e conversas com usuários. Sua abordagem simplificada para criar agentes não exige que você provisione capacidade, gerencie a infraestrutura ou escreva código personalizado.

Principais características dos Amazon Bedrock Agents

O Amazon Bedrock Agents inclui os seguintes recursos principais:

- Serviço totalmente gerenciado — gerenciamento completo da infraestrutura sem a necessidade de provisionar capacidade ou gerenciar sistemas subjacentes. Para obter mais informações, consulte Automatizar tarefas em seu aplicativo usando agentes de IA na documentação do Amazon Bedrock.
- Desenvolvimento orientado por API — defina e execute agentes por meio de chamadas de API simples, especificando modelos, instruções, ferramentas e parâmetros de configuração. Para obter mais informações, consulte Criar e configurar o agente manualmente na documentação do Amazon Bedrock.
- Grupos de ação — defina ações específicas que seu agente pode realizar criando grupos de ação com esquemas de API. Para obter mais informações, consulte Usar grupos de ação para definir ações para seu agente realizar na documentação do Amazon Bedrock.
- Integração da base de conhecimento — Conecte-se perfeitamente às bases de conhecimento Amazon Bedrock para aumentar as respostas dos agentes com os dados da sua organização. Para obter mais informações, consulte Aumentar a geração de respostas para seu agente com base de conhecimento na documentação do Amazon Bedrock.
- Modelos de solicitação avançados — personalize o comportamento do agente por meio de modelos de solicitação para pré-processamento, orquestração, geração de respostas da base de conhecimento e pós-processamento. Para obter mais informações, consulte Melhorar a precisão do agente usando modelos avançados de solicitação no Amazon Bedrock na documentação do Amazon Bedrock.
- Rastreamento e observabilidade — acompanhe o processo de step-by-step raciocínio do agente usando recursos de rastreamento integrados. Para obter mais informações, consulte Rastrear o processo de step-by-step raciocínio do agente usando trace na documentação do Amazon Bedrock.

- Controle de versão e aliases — Crie várias versões do seu agente e implante-as por meio de aliases para lançamentos controlados. Para obter mais informações, consulte Implantar e usar um agente do Amazon Bedrock em seu aplicativo na documentação do Amazon Bedrock.

Quando usar o Amazon Bedrock Agents

O Amazon Bedrock Agents é particularmente adequado para cenários de agentes autônomos, incluindo:

- Organizações que desejam uma experiência totalmente gerenciada para criar e implantar agentes sem gerenciar a infraestrutura
- Projetos que exigem rápido desenvolvimento e implantação de agentes por meio de configuração em vez de código
- Casos de uso que se beneficiam da forte integração com outros recursos do Amazon Bedrock, como bases de conhecimento e guardrails
- Equipes sem recursos internos para criar agentes do zero, mas precisam de recursos autônomos prontos para produção

Abordagem de implementação para Amazon Bedrock Agents

O Amazon Bedrock Agents fornece uma abordagem de implementação baseada em configuração para as partes interessadas da empresa. O serviço permite que as organizações:

- Defina agentes por meio de chamadas de API Console de gerenciamento da AWS ou de chamadas sem escrever código complexo.
- Crie grupos de ação que especifiquem as operações APIs e que o agente pode realizar.
- Conecte as bases de conhecimento para fornecer informações específicas do domínio ao agente.
- Teste e repita o comportamento do agente por meio de uma interface visual.

Essa abordagem gerenciada permite que as equipes de negócios desenvolvam e implantem rapidamente agentes autônomos sem profundo conhecimento técnico no desenvolvimento de modelos de IA ou no gerenciamento de infraestrutura.

Exemplo real dos Amazon Bedrock Agents

Uma solução de operações financeiras (FinOps) descrita nesta postagem do AWS blog usa a estrutura multiagente Amazon Bedrock para criar um assistente de gerenciamento de custos na nuvem orientado por IA. O modelo econômico da Amazon Nova Foundation capacita a solução em que um agente FinOps supervisor central delega tarefas a agentes especializados. Esses agentes buscam e analisam dados de AWS gastos usando AWS Cost Explorer e geram recomendações de economia de custos usando. AWS Trusted Advisor

O sistema inclui acesso seguro do usuário por meio do Amazon Cognito, um front-end hospedado no AWS Amplify, e grupos de AWS Lambda ação para análise e previsão em tempo real. As equipes financeiras podem fazer perguntas em linguagem natural, como “Quais foram meus custos em fevereiro de 2025?” O sistema responde com detalhamentos, sugestões de otimização e previsões, tudo em uma arquitetura escalável e sem servidor implantada usando. AWS CloudFormation

Amazon Bedrock AgentCore

O Amazon Bedrock AgentCore é uma plataforma agente para criar, implantar e operar agentes altamente capazes com segurança em grande escala usando qualquer estrutura, modelo ou protocolo. Usando AgentCore, você pode fazer o seguinte, tudo sem nenhum gerenciamento de infraestrutura:

- Crie agentes com mais rapidez.
- Permita que os agentes realizem ações em todas as ferramentas e dados.
- Execute agentes de forma segura com baixa latência e tempos de execução estendidos.
- Monitore os agentes na produção.

AgentCore elimina o trabalho pesado indiferenciado de criar uma infraestrutura especializada para agentes, permitindo que você acelere a produção de seus agentes. Seus serviços podem ser usados juntos ou de forma independente e são compatíveis com qualquer estrutura CrewAILangGraph, incluindoLlamaIndex,, Strands Agents e. AgentCoretambém é compatível com qualquer modelo de base disponível dentro ou fora do Amazon Bedrock, oferecendo flexibilidade máxima.

AgentCore é composto por vários serviços principais:

- Amazon Bedrock AgentCore Runtime — Fornece um ambiente seguro, sem servidor e escalável para hospedar e executar seus agentes, sem precisar gerenciar qualquer infraestrutura necessária para implantar e executar agentes ou ferramentas de IA.
- Amazon Bedrock AgentCore Memory — oferece um sistema de memória gerenciada, permitindo que os agentes retenham o contexto das interações para conversas mais personalizadas e coerentes, mantendo o conhecimento imediato e de longo prazo.
- Amazon Bedrock AgentCore Gateway — Simplifica o processo de criação, proteção e localização das ferramentas certas para agentes. Com o AgentCore Gateway, os desenvolvedores podem converter APIs funções Lambda e serviços existentes em ferramentas compatíveis com o Model Context Protocol (MCP) e disponibilizá-las aos agentes.
- Amazon Bedrock AgentCore Identity — fornece um serviço de gerenciamento de identidade e acesso de agentes seguro e escalável que acelera o desenvolvimento de agentes de IA. Com o AgentCore Identity, você pode atribuir identidades exclusivas e verificáveis aos agentes, permitindo um controle de acesso refinado e interações seguras baseadas em agentes com sistemas corporativos.
- Ferramentas AgentCore integradas do Amazon Bedrock — Permite que você use ferramentas integradas para aprimorar seu fluxo de trabalho de desenvolvimento e teste. Use essas ferramentas para interagir com seu aplicativo de forma eficaz, permitindo que agentes de IA escrevam e executem códigos com segurança em ambientes de sandbox. Use a ferramenta do navegador para permitir que agentes de IA interajam com sites em grande escala.
- Amazon Bedrock AgentCore Observability — fornece recursos de registro e monitoramento, oferecendo visibilidade em tempo real do desempenho e do comportamento do seu agente para facilitar a depuração e a otimização.

Principais características do AgentCore

AgentCore inclui os seguintes recursos principais:

- Totalmente gerenciado e extensível — AgentCore é um serviço totalmente gerenciado, o que significa que AWS lida com a infraestrutura e a manutenção subjacentes. Também é extensível, o que permite personalizar e aprimorar a funcionalidade de seus agentes. Para obter mais informações, consulte Introdução ao AgentCore Runtime na AgentCore documentação.
- Memória de longo e curto prazo — Ofereça interações mais personalizadas e relevantes equipando os agentes com um sistema de memória para relembrar o contexto das conversas

atuais e do conhecimento de longo prazo. Para obter mais informações, consulte Introdução à AgentCore memória na AgentCore documentação.

- Desenvolvimento e integração simplificados de ferramentas — Permita que seus agentes descubram e usem ferramentas por meio de um único endpoint seguro. Transforme rapidamente seus recursos corporativos existentes em ferramentas prontas para agentes com apenas algumas linhas de código, liberando os desenvolvedores para se concentrarem na criação de recursos exclusivos. Para obter mais informações, consulte Introdução ao AgentCore Gateway na AgentCore documentação.
- Infraestrutura segura e escalável — AgentCore fornece um ambiente seguro e escalável para a implantação e operação de agentes. Ele inclui recursos para gerenciamento de identidade e acesso, criptografia de dados e segurança de rede. Para obter mais informações, consulte Introdução ao AgentCore Identity na AgentCore documentação.
- Integração com uma ampla variedade de ferramentas — Permite que você integre seus agentes a uma variedade de ferramentas, incluindo um interpretador de código e uma ferramenta de navegador que você pode criar usando as ferramentas AgentCore integradas. Para obter mais informações, consulte Começar com o AgentCore Code Interpreter e Começar com o AgentCore Browser na AgentCore documentação.
- Observabilidade e monitoramento abrangentes — Obtenha visibilidade profunda de seus agentes com ferramentas abrangentes para rastrear, depurar e monitorar seu desempenho na produção. Visualize todo o caminho de execução do agente para auditar seu raciocínio e resolver falhas. Use painéis em tempo real e dados de telemetria padronizados para rastrear as principais métricas operacionais. Para obter mais informações, consulte Adicionar observabilidade aos seus AgentCore recursos do Amazon Bedrock na AgentCore documentação.

Quando usar AgentCore

AgentCore é particularmente adequado para cenários de agentes autônomos, incluindo:

- Organizações que desejam acelerar o desenvolvimento e reduzir a sobrecarga operacional com um serviço totalmente gerenciado que lida com infraestrutura, segurança, ferramentas integradas, observabilidade e escalabilidade
- Projetos que precisam de flexibilidade com serviços modulares que funcionam juntos ou de forma independente e são compatíveis com qualquer estrutura, como CrewAI ouLangGraph, e qualquer modelo básico de qualquer fonte

- Casos de uso que exigem agentes interativos e conversacionais que precisam manter o contexto e aprender com as interações anteriores para fornecer respostas personalizadas e relevantes
- Agentes habilitados para realizar tarefas complexas por meio de integração simples com diversos aplicativos, fontes de dados e APIs

Abordagem de implementação para AgentCore

AgentCore foi projetado para organizações que desejam transferir agentes de IA da prova de conceito, criada usando estruturas de agentes de código aberto ou personalizadas, para a produção. Com AgentCore, as organizações podem fazer o seguinte:

- Implante agentes com segurança em uma infraestrutura sem servidor, oferecendo suporte a qualquer estrutura e modelo, com isolamento de sessão e gerenciamento integrado de identidade e acesso para end-to-end segurança e conformidade. Crie rapidamente agentes AgentCore Runtime para as principais estruturas de agentes usando o kit de ferramentas inicial.
- Melhore os agentes integrando memória persistente para retenção de contexto, simplificando o desenvolvimento e a integração de ferramentas por meio AgentCore do Gateway. Aproveite a ferramenta de navegador e o interpretador de código integrados para fluxos de trabalho avançados.
- Rastreie, depure e monitore agentes de IA em produção usando painéis de observabilidade desenvolvidos pelo Amazon CloudWatch Application Insights e OpenTelemetry rastreando as principais métricas dos AgentCore recursos (tempo de execução, memória, gateway e ferramentas).
- Acelere a implantação e a inovação com serviços modulares totalmente gerenciados, blocos compostos juntos ou de forma independente, com qualquer fornecedor de modelos e estruturas de agentes. Essa flexibilidade ajuda as organizações a passar do protótipo para a produção com mais rapidez.

Essa abordagem gerenciada permite que as organizações criem, implantem e executem com rapidez e segurança agentes de IA e sistemas multiagentes de nível empresarial em qualquer escala.

Exemplo real de AgentCore

AWS observou que um dos maiores bancos da América Latina usa AI/ML há anos uma experiência bancária digital hiperpersonalizada e segura. O banco está expandindo os serviços de inteligência artificial AgentCore para fornecer aos clientes interações intuitivas, segurança aprimorada e maior

automação. De acordo com o CTO, AgentCore espera-se que apoie seus esforços para cumprir os compromissos dos clientes em grande escala. AgentCore fornece aos desenvolvedores as ferramentas e a flexibilidade para criar e gerenciar agentes, ao mesmo tempo em que ajuda a garantir a conformidade com as regulamentações financeiras.

Protocolos

Os agentes de IA exigem protocolos de comunicação padronizados para interagir com outros agentes e serviços. As organizações que implementam arquiteturas de agentes enfrentam desafios significativos em relação à interoperabilidade, independência de fornecedores e à preparação de seus investimentos para o futuro.

Esta seção ajuda você a navegar pelo cenário de agent-to-agent protocolos com foco em padrões abertos que maximizam a flexibilidade e a interoperabilidade. (Para obter informações sobre agent-to-tool protocolos, consulte Estratégia de integração de ferramentas mais adiante neste guia.)

Esta seção destaca o Model Context Protocol (MCP), um padrão aberto originalmente desenvolvido Anthropic em 2024. Hoje, apoia AWS ativamente o MCP por meio de contribuições para o desenvolvimento e implementação do protocolo. AWS está colaborando com as principais estruturas de agentes de código aberto, incluindoLangGraph, e CrewAILlamaIndex, para moldar o futuro da comunicação entre agentes no protocolo. Para obter mais informações, consulte Protocolos abertos para interoperabilidade de agentes, parte 1: Comunicação entre agentes no MCP (blog).AWS

Nesta seção:

- Por que a seleção de protocolos é importante
- Agent-to-agent protocolos
- Seleção de protocolos agentes
- Estratégia de implementação para protocolos agentes
- Começando com o MCP
- ???

Por que a seleção de protocolos é importante

A seleção de protocolos molda fundamentalmente como você pode criar e desenvolver sua arquitetura de agente de IA. Ao escolher protocolos que oferecem suporte à portabilidade entre estruturas de agentes, você ganha a flexibilidade de combinar diferentes sistemas e fluxos de trabalho de agentes para atender às suas necessidades específicas.

Os protocolos abertos permitem que você integre agentes em várias estruturas. Por exemplo, use LangChain para prototipagem rápida e implemente sistemas de produção com comunicação por

meio de um protocolo comumStrands Agents, como MCP ou o protocolo Agent2Agent (A2A). Essa flexibilidade reduz a dependência de provedores específicos de IA, simplifica a integração com os sistemas existentes e permite que você aprimore os recursos dos agentes ao longo do tempo.

Protocolos bem projetados também estabelecem padrões de segurança consistentes para autenticação e autorização em todo o ecossistema de agentes. Mais importante ainda, a portabilidade do protocolo preserva sua liberdade de adotar novas estruturas e recursos de agentes à medida que eles surgem. A escolha de protocolos abertos protege seu investimento no desenvolvimento de agentes e, ao mesmo tempo, mantém a interoperabilidade com sistemas de terceiros.

Vantagens dos protocolos abertos

Ao implementar suas próprias extensões ou criar sistemas de agentes personalizados, os protocolos abertos oferecem vantagens convincentes:

- Documentação e transparência — Normalmente fornecem documentação abrangente e implementações transparentes
- Suporte comunitário — Acesso a comunidades mais amplas de desenvolvedores para solução de problemas e melhores práticas
- Garantias de interoperabilidade — Melhor garantia de que suas extensões funcionarão em diferentes implementações
- Compatibilidade futura — risco reduzido de falhas, alterações ou descontinuação
- Influência no desenvolvimento — Oportunidade de contribuir para a evolução do protocolo

Agent-to-agent protocolos

A tabela a seguir fornece uma visão geral dos protocolos de agentes que permitem que vários agentes colaborem, deleguem tarefas e compartilhem informações.

Protocolo	Ideal para	Considerações
Comunicação interagente MCP	Organizações que buscam padrões flexíveis de colaboração de agentes	• Uma extensão do Model Context Protocol (MCP) proposta por AWS that se baseia em sua base

		existente para comunicação agent-to-agent • Permite a colaboração perfeita dos agentes com segurança OAuth baseada
Protocolo A2A	Ecossistemas de agentes multiplataforma	• Apoiado por Google • Padrão mais novo com adoção mais limitada em comparação com o MCP

Decidindo entre as opções de protocolo

Ao implementar a agent-to-agent comunicação, combine seus requisitos específicos de comunicação com os recursos de protocolo apropriados. Padrões de interação diferentes exigem recursos de protocolo diferentes. A tabela a seguir descreve os padrões comuns de comunicação e recomenda as opções de protocolo mais adequadas para cada cenário.

Padrão	Descrição	Escolha de protocolo ideal
Solicitação e resposta simples	Interações pontuais entre agentes	MCP com fluxos sem estado
Diálogos estatais	Conversas contínuas com contexto	MCP com gerenciamento de sessões
Colaboração com vários agentes	Interações complexas entre vários agentes	Interagente MCP ou AutoGen
Fluxos de trabalho baseados em equipe	Equipes hierárquicas de agentes com funções definidas	Interagente MCP, ou CrewAI AutoGen

Além dos padrões de comunicação, vários fatores técnicos e organizacionais podem influenciar sua seleção de protocolos. A tabela a seguir descreve as principais considerações que podem ajudá-lo a avaliar qual protocolo está mais alinhado com seus requisitos específicos de implementação.

Consideração	Descrição	Exemplo
Modelo de segurança	Requisitos de autenticação e autorização	OAuth 2.0 em MCP
Ambiente de implantação	Onde os agentes correrão e se comunicarão	Máquina distribuída ou única
Compatibilidade com ecossistemas	Integração com estruturas de agentes existentes	LangChain ou Strands Agents
Necessidades de escalabilidade	Crescimento esperado nas interações entre agentes	Capacidades de streaming do MCP

Seleção de protocolos agentes

Para a maioria das organizações que criam sistemas de agentes de produção, o Model Context Protocol (MCP) oferece a base mais abrangente e bem apoiada para agent-to-agent comunicação. O MCP se beneficia das contribuições ativas de desenvolvimento AWS e da comunidade de código aberto.

Selecionar os protocolos agentes corretos é importante para organizações que desejam implementar a IA agente de forma eficaz. As considerações diferem com base no contexto organizacional.

Considerações sobre a seleção de protocolos agentes

As organizações devem considerar as seguintes melhores práticas ao selecionar protocolos para sistemas de IA agentes:

- Priorize padrões abertos — As organizações devem adotar protocolos abertos, como o MCP, para ajudar a garantir a interoperabilidade e a extensibilidade a longo prazo e reduzir o risco de dependência de fornecedores.
- Equilibre velocidade e flexibilidade — As startups e os primeiros usuários podem começar com protocolos proprietários bem suportados para um rápido desenvolvimento, mas devem definir um caminho de migração para padrões abertos à medida que os sistemas amadurecem.

- Implemente camadas de abstração — As empresas devem implementar a abstração de protocolos para simplificar a migração, permitir a adoção híbrida e estratégias de integração preparadas para o futuro.
- Enfatize a segurança e a conformidade — Organizações em setores regulamentados devem selecionar protocolos com recursos robustos de autenticação, criptografia e auditoria para atender aos requisitos de governança e conformidade.
- Avalie a maturidade do ecossistema — Todas as organizações devem avaliar a saúde, a adoção e o apoio comunitário de cada protocolo para garantir a sustentabilidade e minimizar a dívida técnica.
- Envolva-se no desenvolvimento de padrões — As organizações devem participar de órgãos de padrões ou comunidades de código aberto para ajudar a moldar a evolução do protocolo e influenciar as melhores práticas.
- Leve em conta a soberania dos dados — O governo e os setores regulamentados devem garantir que as opções de protocolo estejam alinhadas aos requisitos de residência e soberania dos dados em todas as regiões de implantação.
- Aproveite os serviços gerenciados — sempre que possível, use implementações gerenciadas ou sem servidor de protocolos agentes para reduzir a complexidade operacional e acelerar a implantação.

Estratégia de implementação para protocolos agentes

Para implementar protocolos agentes de forma eficaz em sua organização, considere as seguintes etapas estratégicas:

1. Comece com o alinhamento de padrões — adote protocolos abertos estabelecidos sempre que possível.
2. Crie camadas de abstração — Implemente adaptadores entre seus sistemas e protocolos específicos.
3. Contribua com padrões abertos — participe de comunidades de desenvolvimento de protocolos.
4. Monitore a evolução do protocolo — mantenha-se informado sobre os padrões e atualizações emergentes.
5. Teste a interoperabilidade regularmente — verifique se suas implementações permanecem compatíveis.

Começando com o MCP

AWS apoia ativamente o Model Context Protocol (MCP) por meio de contribuições para o desenvolvimento e implementação do protocolo. AWS está colaborando com as principais estruturas de agentes de código aberto, incluindoLangGraph, e CrewAILlamaIndex, para moldar o futuro da comunicação entre agentes no protocolo.

Para implementar o MCP em sua arquitetura de agente, execute as seguintes ações:

1. Explore as implementações do MCP em estruturas como o SDK. Strands Agents
2. Revise a documentação técnica do Model Context Protocol.
3. Leia Protocolos abertos para interoperabilidade de agentes, parte 1: comunicação entre agentes no MCP (AWS blog) para saber mais sobre a interoperabilidade de agentes.
4. Junte-se à comunidade MCP para influenciar a evolução do protocolo.

O MCP fornece uma camada de comunicação que permite que os agentes interajam com dados e serviços externos e também pode ser usada para permitir que os agentes interajam com outros agentes. A implementação de transporte HTTP Streamable do protocolo oferece aos desenvolvedores um conjunto abrangente de padrões de interação sem precisar reinventar a roda. Esses padrões oferecem suporte a request/response fluxos sem estado e ao gerenciamento de sessões com monitoramento de estado com persistência. IDs

Ao adotar protocolos abertos como o MCP, você posiciona sua organização para criar sistemas de agentes que permaneçam flexíveis, interoperáveis e adaptáveis à medida que a tecnologia de IA evolui. Para obter informações sobre a implementação do agent-to-tool protocolo, consulte Estratégia de integração de ferramentas mais adiante neste guia.

Começando com o A2A

O protocolo Agent2Agent (A2A) permite a colaboração descentralizada entre agentes por meio de uma camada semântica compartilhada. Em vez de rotear todo o trabalho por meio de um orquestrador central, o A2A permite que os agentes se descubram, anunciem suas capacidades, negociem tarefas e compartilhem contexto usando um protocolo leve baseado em JSON. Cada agente publica um manifesto de capacidade.

O exemplo a seguir mostra um manifesto simplificado da capacidade A2A que anuncia as ações suportadas, as entradas necessárias e os metadados operacionais de um agente para permitir a descoberta e a negociação de tarefas:

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

Esse modelo permite a correspondência dinâmica de capacidades, a delegação no meio da tarefa e a colaboração entre organizações. Os agentes podem se auto-organizar em torno de tarefas, formar grupos de trabalho temporários e se adaptar à medida que novos recursos entram ou saem do sistema.

O A2A suporta interações que vão desde simples solicitações sem estado até sessões de negociação em várias etapas, incluindo:

- peer-to-peerMensagens diretas para colaboração de baixa latência
- Negociação semântica de tarefas, em que os agentes selecionam o par mais adequado
- Descoberta baseada em capacidades, permitindo a divisão emergente do trabalho
- Ancoragem de sessão para interações com estado em várias etapas

Ao adotar protocolos abertos e nativos de agentes, como o A2A, as organizações criam sistemas de IA que são modulares, interoperáveis e capazes de colaboração transfronteiriça. O A2A garante que os ecossistemas de agentes permaneçam flexíveis e possam evoluir à medida que novos agentes, equipes ou sistemas externos são introduzidos, sem exigir camadas rígidas de orquestração ou acoplamento prévio.

Para implementar o protocolo A2A em sua arquitetura de agente, execute as seguintes ações:

1. Analise a especificação do protocolo A2A — Leia a versão mais recente da especificação do protocolo Agent2Agent (A2A) para saber como os manifestos de capacidade, os fluxos de negociação e o handshake do agente operam.
2. Explore tempos de execução compatíveis com A2A — Avalie estruturas como o SDK Strands Agents ou camadas de tempo de execução personalizadas que oferecem suporte a manifestos e negociações de recursos no estilo A2A. peer-to-peer

3. Implemente um manifesto de capacidades para seus agentes — defina os `meta` campos e os campos de `can` cada agente para permitir a descoberta, a combinação e a colaboração em nível de intenção. `needs`
4. Experimente os padrões de negociação A2A — use o ciclo de solicitação-oferta—aceitação, consultas de recursos estruturados ou descobertas baseadas em fofocas para entender como os agentes raciocinam sobre quem deve lidar com uma tarefa.
5. Teste o A2A em um ambiente de infraestrutura mista — Combine a negociação entre pares A2A com o roteamento de eventos nativo da AWS Amazon para avaliar padrões de coordenação híbrida. EventBridge
6. Junte-se à comunidade A2A — envolva-se com o grupo de trabalho aberto para se manter atualizado com extensões, recomendações de segurança e melhorias na interoperabilidade entre fornecedores e contribua para o desenvolvimento do protocolo.

Ferramentas

Os agentes de IA agregam valor ao interagir com ferramentas externas e fontes de dados para realizar tarefas úteis. APIs A estratégia certa de integração de ferramentas afeta diretamente as capacidades, a postura de segurança e a flexibilidade de longo prazo do seu agente.

Esta seção ajuda você a navegar pelo cenário de integração de ferramentas com foco em padrões abertos que maximizam sua liberdade e flexibilidade. A seção destaca o Model Context Protocol (MCP) para integração de ferramentas e analisa ferramentas específicas da estrutura e meta-ferramentas especializadas que aprimoram os fluxos de trabalho dos agentes.

Nesta seção:

Categorias de ferramentas

Os sistemas de agentes de construção envolvem três categorias principais de ferramentas.

Ferramentas baseadas em protocolos

As ferramentas baseadas em protocolos usam protocolos padronizados para agent-to-tool comunicação:

- Ferramentas MCP — ferramentas padrão abertas que funcionam em várias estruturas com opções de execução local e remota.
- OpenAIchamada de função — ferramentas proprietárias que são específicas para OpenAI modelos.
- Anthropictools — Uma variação da chamada de OpenAI função para ferramentas proprietárias que são específicas dos modelos Anthropic Claude.

Ferramentas nativas da estrutura

As ferramentas nativas da estrutura são incorporadas diretamente em estruturas específicas de agentes:

- Strands Agents ferramentas — quick-to-implement Ferramentas leves, específicas para a Strands Agents estrutura.
- LangChainferramentas — ferramentas Python baseadas em ferramentas que estão totalmente integradas ao LangChain ecossistema.
- LlamaIndexferramentas — Ferramentas que são otimizadas para recuperação e processamento de dados internosLlamaIndex.

Meta-ferramentas

As meta-ferramentas aprimoram os fluxos de trabalho dos agentes sem realizar ações externas diretamente:

- Ferramentas de fluxo de trabalho — gerencie o fluxo de execução do agente, a lógica de ramificação e o gerenciamento do estado.
- Ferramentas de gráfico de agentes — coordene vários agentes em fluxos de trabalho complexos.
- Ferramentas de memória — fornecem armazenamento e recuperação persistentes de informações em todas as sessões do agente.
- Ferramentas de reflexão — Permita que os agentes analisem e melhorem seu próprio desempenho.

Ferramentas baseadas em protocolos

Ao considerar ferramentas baseadas em protocolos, o Model Context Protocol (MCP) fornece a base mais abrangente e flexível para a integração de ferramentas. Conforme declarado na postagem do blog AWS Open Source sobre interoperabilidade de agentes, AWS adotou o MCP como um protocolo estratégico, contribuindo ativamente para seu desenvolvimento.

A tabela a seguir descreve as opções para a implantação da ferramenta MCP.

Modelo de implantação	Descrição	Ideal para	Implementação
Baseado em estúdio local	As ferramentas são executadas no mesmo processo que o agente	Desenvolvimento, teste e ferramentas simples	Rápido de implementar sem sobrecarga de rede
Baseado em eventos enviados pelo servidor local (SSE)	As ferramentas são executadas localmente, mas se comunicam por HTTP	Ferramentas locais mais complexas com separação de interesses	Melhor isolamento, mas ainda baixa latência
HTTP remoto que pode ser transmitido	Ferramentas executadas em servidores remotos	Ambientes de produção e ferramentas compartilhadas	Escalável e gerenciado centralmente

Os MCP oficiais SDKs estão disponíveis para criar ferramentas MCP:

- PythonSDK — Implementação abrangente com suporte total ao protocolo
- TypeScriptSDK —JavaScript/TypeScriptimplementação para aplicativos web
- JavaSDK — implementação Java para aplicativos corporativos

Eles SDKs fornecem os alicerces para a criação de ferramentas compatíveis com MCP em sua linguagem preferida, com implementações consistentes da especificação do protocolo.

Além disso, AWS implementou o MCP no Strands AgentsSDK. O Strands Agents SDK fornece uma maneira simples de criar e usar ferramentas compatíveis com MCP. Uma documentação abrangente está disponível no Strands Agents GitHub repositório. Para casos de uso mais simples ou ao trabalhar fora da Strands Agents estrutura, o MCP oficial SDKs oferece implementações diretas do protocolo em vários idiomas.

Recursos de segurança das ferramentas MCP

Os recursos de segurança das ferramentas MCP incluem o seguinte:

- OAuth Autenticação 2.0/2.1 — Autenticação padrão do setor

- Escopo de permissões — controle de acesso refinado para ferramentas
- Descoberta da capacidade da ferramenta — descoberta dinâmica das ferramentas disponíveis
- Tratamento estruturado de erros — padrões de erro consistentes

Introdução às ferramentas MCP

Para implementar o MCP para integração de ferramentas, execute as seguintes ações:

1. Explore o Strands AgentsSDK para uma implementação de MCP pronta para produção.
2. Revise a documentação técnica do MCP para entender os principais conceitos.
3. Use os exemplos práticos descritos nesta postagem do blog de código AWS aberto.
4. Comece com ferramentas locais simples antes de passar para ferramentas remotas.
5. Junte-se à comunidade MCP para influenciar a evolução do protocolo.

Conheça o AgentCore Gateway

O Amazon Bedrock AgentCore Gateway fornece uma maneira fácil e segura para os desenvolvedores criarem, implantarem, descobrirem e se conectarem às ferramentas de MCP e a outros endpoints de destino em grande escala. Com o AgentCore Gateway, os desenvolvedores podem converter APIs AWS Lambda funções e serviços existentes em ferramentas compatíveis com MCP. Então, com apenas algumas linhas de código, eles podem disponibilizar essas ferramentas aos agentes por meio de endpoints do AgentCore Gateway. AgentCore O Gateway suporta OpenAPISmithy, e Lambda como tipos de entrada e é a única solução que fornece autenticação abrangente de entrada e autenticação de saída em um serviço totalmente gerenciado.

Ferramentas nativas da estrutura

Embora o Model Context Protocol (MCP) forneça a base mais flexível, as ferramentas nativas da estrutura oferecem vantagens para casos de uso específicos.

O Strands AgentsSDK oferece ferramentas Python baseadas caracterizadas por seu design leve que requer sobrecarga mínima para operações simples. Eles permitem uma implementação rápida e permitem que os desenvolvedores criem ferramentas com apenas algumas linhas de código. Além disso, eles são totalmente integrados para funcionar perfeitamente dentro da Strands Agents estrutura.

O exemplo a seguir demonstra como criar uma ferramenta climática simples usandoStrands Agents. Os desenvolvedores podem transformar rapidamente Python funções em ferramentas acessíveis por agentes com o mínimo de sobrecarga de código e gerar automaticamente a documentação apropriada a partir da docstring da função.

```
#Example of a simple Strands native tool

@tool

def weather(location: str) -> str:

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

Implementation here

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

Para prototipagem rápida ou casos de uso simples, as ferramentas nativas da estrutura podem acelerar o desenvolvimento. No entanto, para sistemas de produção, as ferramentas MCP oferecem melhor interoperabilidade e flexibilidade futura do que as ferramentas nativas da estrutura.

A tabela a seguir fornece uma visão geral de outras ferramentas específicas da estrutura.

Framework	Tipo de ferramenta	Vantagens	Considerações
AutoGen	Definições de funções	Forte suporte multiagente	Microsoftecossistema
LangChain	Pythonaulas	Grande ecossistema de ferramentas pré-construídas	Bloqueio de estrutura
LlamaIndex	Funções do Python	Otimizado para operações de dados	Limitado a LlamaIndex

Meta-ferramentas

As meta-ferramentas não interagem diretamente com sistemas externos. Em vez disso, eles aprimoram as capacidades dos agentes implementando padrões agentes. Esta seção aborda o fluxo de trabalho, o gráfico do agente e as meta-ferramentas de memória.

Meta-ferramentas de fluxo de trabalho

As meta-ferramentas de fluxo de trabalho gerenciam o fluxo de execução do agente:

- Gerenciamento de estado — mantenha o contexto em várias interações de agentes
- Lógica de ramificação — Habilite caminhos de execução condicional
- Mecanismos de repetição — Lide com falhas com estratégias sofisticadas de repetição

Exemplos de estruturas com meta-ferramentas de fluxo de trabalho incluem LangGraphrecursos de fluxo de trabalho. Strands Agents

Meta-ferramentas de gráficos de agentes

As meta-ferramentas de gráficos de agentes coordenam vários agentes trabalhando juntos:

- Delegação de tarefas — atribua subtarefas a agentes especializados
- Agregação de resultados — Combine resultados de vários agentes
- Resolução de conflitos — Resolva divergências entre agentes

Frameworks como AutoGene se CrewAIespecializam em coordenação gráfica de agentes.

Meta-ferramentas de memória

As meta-ferramentas de memória fornecem armazenamento e recuperação persistentes:

- Histórico de conversas — mantenha o contexto em todas as sessões
- Bases de conhecimento — Armazene e recupere informações específicas do domínio
- Armazenamentos vetoriais — Habilite recursos de pesquisa semântica

O sistema de recursos do MCP fornece uma maneira padronizada de implementar meta-ferramentas de memória que funcionam em diferentes estruturas de agentes.

Estratégia de integração de ferramentas

Sua escolha de estratégia de integração de ferramentas afeta diretamente o que seus agentes podem realizar e a facilidade com que seu sistema pode evoluir. Priorize protocolos abertos, como

o Model Context Protocol (MCP), enquanto usa estrategicamente ferramentas e meta-ferramentas nativas da estrutura. Dessa forma, você pode criar um ecossistema de ferramentas que permaneça flexível e poderoso à medida que a tecnologia de IA avança.

A seguinte abordagem estratégica para integração de ferramentas maximiza a flexibilidade e, ao mesmo tempo, atende às necessidades imediatas da sua organização:

1. Adote o MCP como sua base — O MCP fornece uma maneira padronizada de conectar agentes a ferramentas com recursos de segurança robustos. Comece com o MCP como seu protocolo de ferramenta principal para:
   - Ferramentas estratégicas que serão usadas em várias implementações de agentes.
   - Ferramentas sensíveis à segurança que exigem autenticação e autorização robustas.
   - Ferramentas que precisam de execução remota em ambientes de produção.
2. Use ferramentas nativas da estrutura quando apropriado — Considere as ferramentas nativas da estrutura para:
   - Prototipagem rápida durante o desenvolvimento inicial.
   - Ferramentas simples e não críticas com requisitos mínimos de segurança.
   - Funcionalidade específica da estrutura que aproveita recursos exclusivos.
3. Implemente meta-ferramentas para fluxos de trabalho complexos — Adicione meta-ferramentas para aprimorar sua arquitetura de agente:
   - Comece de forma simples com padrões básicos de fluxo de trabalho.
   - Adicione complexidade à medida que seus casos de uso amadurecem.
   - Padronize as interfaces entre agentes e meta-ferramentas.
4. Planeje a evolução — Crie com a flexibilidade futura em mente:
   - Documente as interfaces das ferramentas independentemente das implementações.
   - Crie camadas de abstração entre agentes e ferramentas.
   - Estabeleça caminhos de migração de protocolos proprietários para protocolos abertos.

Melhores práticas de segurança para integração de ferramentas

A integração de ferramentas afeta diretamente sua postura de segurança. Esta seção descreve as melhores práticas a serem consideradas em sua organização.

Autenticação e autorização

Use os seguintes controles de acesso robustos:

- Use OAuth 2.0/2.1 — Implemente a autenticação padrão do setor para ferramentas remotas.
- Implemente o menor privilégio — conceda às ferramentas somente as permissões de que elas precisam.
- Alterne as credenciais — atualize regularmente as chaves de API e os tokens de acesso.

Proteção de dados

Para ajudar a proteger os dados, adote as seguintes medidas:

- Valide entradas e saídas — Implemente a validação do esquema para todas as interações da ferramenta.
- Criptografe dados confidenciais — use o TLS para todas as comunicações remotas com ferramentas.
- Implemente a minimização de dados — passe somente as informações necessárias para as ferramentas.

Monitoramento e auditoria

Mantenha a visibilidade e o controle usando esses mecanismos:

- Registre todas as invocações de ferramentas — mantenha trilhas de auditoria abrangentes.
- Monitore anomalias — Detecte padrões incomuns de uso de ferramentas.
- Implemente a limitação de taxa — Evite abusos por meio de chamadas excessivas de ferramentas.

O modelo de segurança do Model Context Protocol (MCP) aborda essas preocupações de forma abrangente. Para obter mais informações, consulte Considerações de segurança na documentação do MCP.

Conclusão

O cenário da IA agêntica continua evoluindo rapidamente, oferecendo às organizações novas maneiras poderosas de criar sistemas inteligentes e autônomos. Este guia explorou três componentes essenciais para uma implementação bem-sucedida: estruturas que fornecem a base, plataformas que fornecem o ambiente, protocolos que permitem a comunicação e ferramentas que ampliam os recursos.

À medida que as estruturas amadurecem, você pode esperar maior interoperabilidade, padronização em torno de protocolos como o Model Context Protocol (MCP) e recursos de orquestração mais sofisticados para agentes autônomos. As organizações que estabelecem experiência com essas estruturas hoje estarão bem posicionadas para criar agentes cada vez mais autônomos e inteligentes que ofereçam um valor comercial significativo.

As plataformas fornecem o ambiente de execução, governança e ciclo de vida no qual os sistemas agentes operam. Eles lidam com questões como identidade, limites de segurança, observabilidade, gerenciamento de memória, aterramento de sessões e interação segura com ferramentas e dados. Em AWS ambientes, plataformas como tempos de execução de agentes gerenciados e serviços de orquestração permitem que as organizações implantem, monitorem, evoluam e governem agentes autônomos e sistemas agentes em grande escala. As plataformas unem as estruturas fundamentais aos requisitos operacionais do mundo real.

A escolha dos protocolos do agente representa uma decisão estratégica que equilibra as necessidades imediatas de desenvolvimento com flexibilidade e interoperabilidade de longo prazo. Ao priorizar protocolos abertos e criar camadas de abstração apropriadas, as organizações podem criar sistemas de agentes que permaneçam adaptáveis às tecnologias em evolução e, ao mesmo tempo, atendam aos requisitos comerciais atuais.

Para a maioria das organizações, o MCP representa uma base sólida devido ao seu padrão aberto, ecossistema em crescimento, suporte a padrões de agent-to-agent comunicação e recursos de integração de ferramentas. AWS adotou o MCP e o Agent2Agent (A2A) como protocolos estratégicos, contribuindo ativamente para seu desenvolvimento e implementando-os em serviços como o SDK. Strands Agents Ao usar MCP ou A2A junto com ferramentas e meta-ferramentas nativas da estrutura apropriadas, você pode criar sistemas de agentes que ofereçam valor imediato e, ao mesmo tempo, permaneçam adaptáveis às inovações futuras.

Recursos

Use os seguintes AWS e outros recursos relacionados ao desenvolvimento de agentes autônomos.

AWS Blogs

- Amazon Bedrock AgentCore Memory: criando agentes sensíveis ao contexto
- Best practices for building robust generative AI applications with Amazon Bedrock Agents – Part 1
- Best practices for building robust generative AI applications with Amazon Bedrock Agents – Part 2
- Crie pipelines RAG poderosos com o LlamaIndex Amazon Bedrock
- Crie agentes de IA confiáveis com o Amazon Bedrock Observability AgentCore
- Avalie as respostas do RAG com o Amazon Bedrock e o RAGAS LlamaIndex
- Apresentando o Amazon Bedrock AgentCore Code Interpreter
- Apresentando o Amazon Bedrock AgentCore Gateway: transformando o desenvolvimento de ferramentas de agentes de IA corporativos
- Apresentando o Amazon Bedrock AgentCore Identity: protegendo a IA agente em grande escala
- Apresentando Strands Agents um SDK de agentes de IA de código aberto
- Protocolos abertos para interoperabilidade de agentes, parte 1: Comunicação entre agentes no MCP
- Lance e escale com segurança seus agentes e ferramentas no Amazon Bedrock Runtime AgentCore
- AWS Transform para o.NET, o primeiro serviço de IA agente para modernizar aplicativos.NET em grande escala
- AWS Resumo semanal: Strands Agents

AWS Orientação prescritiva

- Operacionalizando a IA agente em AWS
- Fundamentos da IA agêntica em AWS
- Padrões e fluxos de trabalho de IA agentes em AWS
- Construindo arquiteturas sem servidor para IA agente em AWS

- Criação de arquiteturas multilocatárias para IA agente em AWS
- Segurança para IA agente em AWS
- Opções e arquiteturas de geração aumentada de recuperação em AWS

AWS recursos

- Documentação do Amazon Bedrock
- Documentação do Amazon Bedrock AgentCore
- Kit de ferramentas Amazon Bedrock AgentCore Starter (repositório) GitHub
- Documentação do Amazon Nova
- AWS Servidores MCP (GitHubrepositório)

Outros recursos da

- AutoGendocumentação (Microsoft)
- Construindo agentes eficazes (Anthropic)
- CrewAI GitHubrepositório
- Documentação da LangChain
- LangGraphplataforma
- Documentação da LlamaIndex
- Documentação do Model Context Protocol
- Documentação da Strands Agents
- Strands AgentsVisão geral das ferramentas
- Strands AgentsGuia de início rápido

Histórico do documento

A tabela a seguir descreve alterações significativas feitas neste guia. Se desejar receber notificações sobre futuras atualizações, inscreva-se em um feed RSS.

Alteração	Descrição	Data
Nova seção	Seção de plataformas adicionada	16 de janeiro de 2026
Publicação inicial	—	14 de julho de 2025

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.