Marcos, plataformas, protocolos y herramientas de inteligencia artificial de las agencias en AWS

AWS Orientación prescriptiva

AWS Orientación prescriptiva: Marcos, plataformas, protocolos y herramientas de inteligencia artificial de las agencias en AWS

Table of Contents

Marcos, plataformas, protocolos y herramientas de inteligencia artificial de las agencias en AWS

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

Enero de 2026 (historia del documento)

La IA de agencia es un poderoso paradigma en la intersección de la IA, los sistemas distribuidos y la ingeniería de software. Es una clase de sistemas inteligentes compuestos por agentes de software autónomos y asíncronos que utilizan modelos de IA y se integran con herramientas y recursos. Los agentes demuestran su capacidad de acción, son capaces de percibir el contexto, razonar antes que los objetivos, tomar decisiones y adoptar medidas decididas en nombre de los usuarios o los sistemas. Estos agentes funcionan de forma independiente, a menudo en colaboración, en entornos distribuidos y están diseñados para perseguir los objetivos delegados con inteligencia, memoria e intención integradas.

Además AWS, las organizaciones pueden aprovechar la IA de los agentes para automatizar flujos de trabajo complejos, mejorar los procesos de toma de decisiones y crear sistemas con mayor capacidad de respuesta. Esta guía proporciona información sobre los componentes clave que son necesarios para crear soluciones de inteligencia artificial eficaces para los agentes:

- Frameworks describe los marcos actuales de inteligencia artificial de las agencias, incluidas las revisiones de sus beneficios y casos de uso. Descubra cómo estos marcos reducen el trabajo pesado indiferenciado entre patrones, protocolos y herramientas. Comprenda los criterios de selección clave para elegir el marco adecuado para sus requisitos.
- Platforms ofrece una visión general de las plataformas de IA de los agentes (agente gestionado, orquestación de código abierto e híbridas) y aspectos a tener en cuenta a la hora de seleccionarlas o diseñarlas.
- Protocols explora los protocolos de comunicación estandarizados esenciales para las interacciones entre los agentes. Agent-to-agentestán surgiendo protocolos, como el Model Context Protocol (MCP) y el Agent2Agent (A2A) de código abierto, junto con otras implementaciones patentadas. Descubra cómo los protocolos comunes permiten que diferentes protocolos interactúen sin problemas.
- Las herramientas proporcionan información sobre las herramientas basadas en protocolos (como el MCP), las herramientas nativas del marco y las metaherramientas. Las organizaciones pueden

crear un conjunto de herramientas que se integre con los sistemas clave de sus flujos de trabajo, lo que permite flujos de trabajo de agentes basados en servidores y usuarios finales.

Destinatarios previstos

Esta guía está dirigida a arquitectos, desarrolladores y líderes tecnológicos que buscan aprovechar el poder de los agentes de software impulsados por la IA en aplicaciones modernas nativas de la nube.

Objetivos

Esta guía lo ayuda a hacer lo siguiente:

- Compare diferentes marcos de IA de agencias para seleccionar el más adecuado para su caso de uso.
- Obtenga información sobre las plataformas de IA de los agentes que ofrecen capacidades para convertir a los agentes individuales en sistemas coordinados y adaptables.
- Comprenda las ventajas de los protocolos abiertos para crear arquitecturas de IA de agencias sostenibles.
- Cree una estrategia de integración de herramientas adecuada al crear sistemas de agentes.

Acerca de esta serie de contenido

Esta guía forma parte de una serie sobre la IA de los agentes en. AWS Para obtener más información y ver las demás guías de esta serie, consulte Agentic AI en el sitio web de orientación prescriptiva. AWS

Marcos

Foundations of Agentic AI on AWS examina los patrones y flujos de trabajo principales que permiten un comportamiento autónomo y orientado a objetivos. En el centro de la implementación de estos patrones se encuentra la elección del marco. Un marco es la base de software del código preescrito que proporciona un entorno estructurado y una funcionalidad común para crear y gestionar las herramientas y las capacidades de organización necesarias para crear agentes de IA autónomos listos para la producción.

Los marcos de inteligencia artificial eficaces proporcionan varias capacidades esenciales que transforman las interacciones sin procesar con modelos de lenguaje extensos (LLM) en sistemas coordinados e inteligentes capaces de razonar, colaborar y actuar:

- La orquestación de agentes coordina el flujo de información y la toma de decisiones entre uno o varios agentes para lograr objetivos complejos sin intervención humana.
- La integración de herramientas permite a los agentes interactuar con sistemas, API y fuentes de datos externos para ampliar sus capacidades más allá del procesamiento del lenguaje. Para obtener más información, consulte la descripción general de las herramientas en la Strands Agents documentación.
- La administración de la memoria proporciona un estado persistente o basado en sesiones para mantener el contexto en todas las interacciones, algo esencial para las tareas de larga duración o adaptativas. Los marcos más avanzados incorporan memoria a largo plazo para almacenar los resúmenes y las preferencias de los usuarios, lo que permite experiencias de los agentes personalizadas y sensibles al contexto. Para obtener más información, consulte Cómo pensar en los marcos de agentes en el blog. LangChain
- La definición del flujo de trabajo admite patrones estructurados como las cadenas, el enrutamiento, la paralelización y los bucles de reflexión que permiten un razonamiento autónomo sofisticado.
- La implementación y el monitoreo facilitan la transición del desarrollo a la producción con la observabilidad de los sistemas autónomos. Para obtener más información, consulte el anuncio de disponibilidad AgentCore general de Amazon Bedrock.

Estas capacidades se implementan con diferentes enfoques y énfasis en todo el panorama estructural, y cada una de ellas ofrece ventajas distintas para los diferentes casos de uso de agentes autónomos y contextos organizacionales.

En esta sección, se describen y comparan los principales marcos para crear soluciones de inteligencia artificial basadas en agentes, centrándose en sus puntos fuertes, limitaciones y casos de uso ideales para el funcionamiento autónomo:

- Agentes de Strands
- LangChain y LangGraph
- Crew AI
- AutoGen
- ???
- Comparación de los marcos de IA de las agencias

> **Note**
>
> Esta sección cubre los marcos que respaldan específicamente a la agencia de la IA y no cubre las interfaces frontend o la IA generativa sin agencia.

Strands Agents

Strands Agentses un SDK de código abierto que fue lanzado inicialmente por AWS, tal y como se describe en el blog de código AWS abierto. Strands Agentsestá diseñado para crear agentes de IA autónomos con un enfoque centrado en el modelo. Proporciona un marco flexible y ampliable diseñado para funcionar sin problemas y, al Servicios de AWS mismo tiempo, abierto a la integración con componentes de terceros. Strands Agents es ideal para crear soluciones totalmente autónomas.

Características clave de Strands Agents

Strands Agentsincluye las siguientes características clave:

- Model-first diseño: se basa en el concepto de que el modelo básico es el núcleo de la inteligencia de los agentes, lo que permite un razonamiento autónomo sofisticado. Para obtener más información, consulte Agent Loop en la Strands Agents documentación.
- Multi-agent patrones de colaboración: modelos de Built-in coordinación como los patrones de Swarm, Graph y Workflow que permiten una colaboración y un gobierno escalables en redes de agentes distribuidas. Para obtener más información, consulte Multi-agent los patrones en la documentación de Strands Agents.

- Integración del MCP: soporte nativo para el Protocolo de Contexto Modelo (MCP), lo que permite el suministro de contexto estandarizado a los LLM para un funcionamiento autónomo y uniforme.
- Servicio de AWS integración: conexión perfecta a Amazon Bedrock y otros Servicios de AWS para flujos de trabajo autónomos integrales. AWS Lambda AWS Step Functions Para obtener más información, consulte el resumen AWS semanal (AWS blog).
- Selección de modelos de base: admite varios modelos de base, incluidos Anthropic Claude, Amazon Nova (Premier, Pro, Lite y Micro) en Amazon Bedrock y otros para optimizarlos para diferentes capacidades de razonamiento autónomo. Para obtener más información, consulte Amazon Bedrock en la Strands Agents documentación.
- Integración de la API LLM: integración flexible con diferentes interfaces de servicios LLM, incluidas Amazon Bedrock, OpenAI y otras, para la implementación en producción. Para obtener más información, consulte Uso básico de Amazon Bedrock en la Strands Agents documentación.
- Capacidades multimodales: Support para múltiples modalidades, incluido el procesamiento de texto, voz e imágenes para interacciones integrales entre agentes autónomos. Para obtener más información, consulte Amazon Bedrock Multimodal Support en la Strands Agents documentación.
- Ecosistema de herramientas: amplio conjunto de herramientas para la Servicio de AWS interacción, con capacidad de ampliación para herramientas personalizadas que amplían las capacidades autónomas. Para obtener más información, consulte la descripción general de las herramientas en la Strands Agents documentación.

Cuándo se debe usar Strands Agents

Strands Agentses especialmente adecuado para escenarios de agentes autónomos, que incluyen:

- Organizaciones que se basan en una AWS infraestructura que desean una integración nativa con flujos Servicios de AWS de trabajo autónomos
- Equipos que requieren funciones de seguridad, escalabilidad y cumplimiento de nivel empresarial para los sistemas autónomos de producción
- Proyectos que necesitan flexibilidad a la hora de seleccionar modelos entre distintos proveedores para realizar tareas autónomas especializadas
- Casos de uso que requieren una estrecha integración con los AWS flujos de trabajo y los recursos existentes para lograr procesos autónomos de principio a fin

Enfoque de implementación para Strands Agents

Strands Agentsproporciona un enfoque de implementación sencillo para las partes interesadas de la empresa, tal como se describe en su Guía de inicio rápido Python y su Guía de inicio rápido para. Typescript El marco permite a las organizaciones:

- Seleccione modelos de base como Amazon Nova (Premier, Pro, Lite o Micro) en Amazon Bedrock en función de los requisitos empresariales específicos.
- Defina herramientas personalizadas que se conecten a los sistemas y fuentes de datos empresariales.
- Procese múltiples modalidades, incluyendo texto, imágenes y voz.
- Implemente agentes que puedan responder de forma autónoma a las consultas empresariales y realizar tareas.

Este enfoque de implementación permite a los equipos empresariales desarrollar y desplegar rápidamente agentes autónomos sin necesidad de una amplia experiencia técnica en el desarrollo de modelos de IA.

Real-world ejemplo de Strands Agents

AWS Transform para que .NET Strands Agents potencie sus capacidades de modernización de aplicaciones, tal y como se describe en AWS Transform for.NET, el primer servicio de inteligencia artificial para modernizar las aplicaciones.NET a gran escala (Blog)AWS . Este servicio de producción emplea a varios agentes autónomos especializados. Los agentes trabajan juntos para analizar las aplicaciones de .NET heredadas, planificar estrategias de modernización y ejecutar transformaciones de código en arquitecturas nativas de la nube sin intervención humana. AWS Transform para .NET demuestra la preparación para la producción de los Strands Agents sistemas autónomos empresariales.

LangChain y LangGraph

LangChaines uno de los marcos más consolidados en el ecosistema de la IA de las agencias. LangGraphamplía sus capacidades para admitir flujos de trabajo de agentes complejos y detallados, tal y como se describe en el blog. LangChain En conjunto, proporcionan una solución integral para crear agentes de IA autónomos sofisticados con amplias capacidades de orquestación para un funcionamiento independiente.

Características clave de y LangChain LangGraph

LangChaine LangGraph incluyen las siguientes características clave:

- Ecosistema de componentes: amplia biblioteca de componentes prediseñados para diversas capacidades de agentes autónomos, lo que permite el rápido desarrollo de agentes especializados. Para obtener más información, consulte la sección Inicio rápido en la LangChain documentación.
- Selección de modelos de base: Support para diversos modelos de base, incluidos Anthropic Claude, modelos Amazon Nova (Premier, Pro, Lite y Micro) en Amazon Bedrock y otros para diferentes capacidades de razonamiento. Para obtener más información, consulte Entradas y salidas en la LangChain documentación.
- Integración de la API LLM: interfaces estandarizadas para varios proveedores de servicios de modelos de lenguaje grandes (LLM), incluido Amazon BedrockOpenAI, y otros para una implementación flexible. Para obtener más información, consulte las LLM en la documentación. LangChain
- Procesamiento multimodal: Built-in admite el procesamiento de texto, imágenes y audio para permitir interacciones sofisticadas entre agentes autónomos multimodales. Para obtener más información, consulte Multimodalidad en la documentación. LangChain
- Graph-based flujos de trabajo: LangGraph permite definir comportamientos complejos de agentes autónomos como máquinas de estados, lo que respalda una sofisticada lógica de decisiones. Para obtener más información, consulte el anuncio de LangGraphPlatform GA.
- Abstracciones de memoria: múltiples opciones para la gestión de la memoria a corto y largo plazo, algo esencial para los agentes autónomos que mantienen el contexto a lo largo del tiempo. Para obtener más información, consulte Cómo añadir memoria a los chatbots en la LangChain documentación.
- Integración de herramientas: amplio ecosistema de integraciones de herramientas en varios servicios y API, que amplía las capacidades de los agentes autónomos. Para obtener más información, consulte las herramientas en la LangChain documentación.
- LangGraph plataforma: solución gestionada de implementación y supervisión para entornos de producción, que admite agentes autónomos de larga duración. Para obtener más información, consulte el anuncio de LangGraphPlatform GA.

Cuándo usar LangChain y LangGraph

LangChainy LangGraph son especialmente adecuados para escenarios de agentes autónomos, que incluyen:

- Flujos de trabajo complejos de razonamiento de varios pasos que requieren una orquestación sofisticada para una toma de decisiones autónoma
- Proyectos que necesitan acceso a un gran ecosistema de componentes e integraciones prediseñados para diversas capacidades autónomas
- Equipos con una infraestructura y experiencia Python en aprendizaje automático (ML) existentes que desean crear sistemas autónomos
- Casos de uso que requieren una gestión del estado compleja en sesiones de agentes autónomos de larga duración

Enfoque de implementación para y LangChain LangGraph

LangChainy LangGraph proporcionar un enfoque de implementación estructurado para las partes interesadas de la empresa, tal como se detalla en la LangGraphdocumentación. El marco permite a las organizaciones:

- Defina gráficos de flujo de trabajo sofisticados que representen los procesos empresariales.
- Cree patrones de razonamiento de varios pasos con puntos de decisión y lógica condicional.
- Integre las capacidades de procesamiento multimodal para gestionar diversos tipos de datos.
- Implemente el control de calidad mediante mecanismos integrados de revisión y validación.

Este enfoque basado en gráficos permite a los equipos empresariales modelar procesos de decisión complejos como flujos de trabajo autónomos. Los equipos tienen una visibilidad clara de cada paso del proceso de razonamiento y la capacidad de auditar las vías de toma de decisiones.

Real-world ejemplo de LangChain y LangGraph

Vodafoneha implementado agentes autónomos que utilizan LangChain (yLangGraph) para mejorar sus flujos de trabajo de operaciones e ingeniería de datos, como se detalla en su estudio de caso LangChain empresarial. Crearon asistentes de IA internos que supervisan de forma autónoma las métricas de rendimiento, recuperan información de los sistemas de documentación y presentan información útil, todo ello mediante interacciones en lenguaje natural.

La Vodafone implementación utiliza cargadores de documentos LangChain modulares, integración vectorial y soporte para múltiples LLM (1OpenAI, LLaMA 3 y 2Gemini) para crear prototipos y comparar rápidamente estas canalizaciones. Luego, solían LangGraph estructurar la organización multiagente mediante el despliegue de subagentes modulares. Estos agentes realizan tareas de recopilación, procesamiento, resumen y razonamiento. LangGraphintegraron estos agentes mediante API en sus sistemas en la nube.

CrewAI

CrewAIes un marco de código abierto centrado específicamente en la orquestación autónoma de múltiples agentes, disponible en. GitHub Proporciona un enfoque estructurado para crear equipos de agentes autónomos especializados que colaboran para resolver tareas complejas sin intervención humana. CrewAIhace hincapié en la coordinación basada en funciones y la delegación de tareas.

Características clave de CrewAI

CrewAIproporciona las siguientes características clave:

- Role-based diseño de agentes: los agentes autónomos se definen con funciones, objetivos e historias de fondo específicos para disponer de conocimientos especializados. Para obtener más información, consulte Cómo crear agentes eficaces en la CrewAI documentación.
- Delegación de tareas: Built-in mecanismos para asignar tareas de forma autónoma a los agentes apropiados en función de sus capacidades. Para obtener más información, consulte Tareas en la CrewAI documentación.
- Colaboración entre agentes: marco para la comunicación autónoma entre agentes y el intercambio de conocimientos sin mediación humana. Para obtener más información, consulte la colaboración en la CrewAI documentación.
- Gestión de procesos: flujos de trabajo estructurados para la ejecución secuencial y paralela de tareas autónomas. Para obtener más información, consulte Procesos en la CrewAI documentación.
- Selección de modelos de base: Support para varios modelos de base, incluidos los modelos Anthropic Claude, Amazon Nova (Premier, Pro, Lite y Micro) en Amazon Bedrock y otros para optimizarlos para diferentes tareas de razonamiento autónomo. Para obtener más información, consulte los LLM en la CrewAI documentación.
- Integración de la API LLM: integración flexible con múltiples interfaces de servicios LLM, incluida Amazon BedrockOpenAI, e implementaciones de modelos locales. Para obtener más información, consulte los ejemplos de configuración de proveedores en la documentación. CrewAI

- Soporte multimodal: capacidades emergentes para gestionar texto, imágenes y otras modalidades para interacciones integrales entre agentes autónomos. Para obtener más información, consulte Uso de agentes multimodales en la CrewAI documentación.

Cuándo se debe usar CrewAI

CrewAIes especialmente adecuado para escenarios de agentes autónomos, que incluyen:

- Problemas complejos que se benefician de una experiencia especializada y basada en funciones que funcione de forma autónoma
- Proyectos que requieren la colaboración explícita entre varios agentes autónomos
- Casos de uso en los que la descomposición de problemas en equipo mejora la resolución autónoma de problemas
- Escenarios que requieren una separación clara de las preocupaciones entre las diferentes funciones de los agentes autónomos

Enfoque de implementación para CrewAI

CrewAIproporciona un enfoque de implementación basado en roles de equipos de agentes de IA para las partes interesadas de la empresa, tal como se detalla en Cómo empezar en la CrewAI documentación. El marco permite a las organizaciones:

- Defina agentes autónomos especializados con funciones, objetivos y áreas de experiencia específicos.
- Asigne tareas a los agentes en función de sus capacidades especializadas.
- Establezca dependencias claras entre las tareas para crear flujos de trabajo estructurados.
- Organice la colaboración entre varios agentes para resolver problemas complejos.

Este enfoque basado en roles refleja las estructuras de los equipos humanos, lo que hace que los líderes empresariales lo entiendan e implementen de forma intuitiva. Las organizaciones pueden crear equipos autónomos con áreas de experiencia especializadas que colaboren para alcanzar los objetivos empresariales, de forma similar a como funcionan los equipos humanos. Sin embargo, el equipo autónomo puede trabajar de forma continua sin intervención humana.

Real-world ejemplo de CrewAI

AWS ha implementado sistemas autónomos multiagente mediante CrewAI integrado con Amazon Bedrock, como se detalla en el CrewAI estudio de caso publicado. AWS y CrewAI desarrolló un marco seguro e independiente de los proveedores. La arquitectura CrewAI de código abierto «flujos y personal» se integra perfectamente con los modelos básicos, los sistemas de memoria y las barreras de cumplimiento de Amazon Bedrock.

Los elementos clave de la implementación incluyen:

- Planos y código abierto, y diseños de referencia CrewAI publicados que mapean CrewAI los agentes con los modelos y las herramientas de observabilidad de Amazon Bedrock. AWS También presentaron sistemas ejemplares, como un equipo de auditoría de AWS seguridad compuesto por varios agentes, flujos de modernización del código y automatización administrativa de bienes de consumo envasados (CPG).
- Integración de la pila de observabilidad: la solución incorpora la supervisión con Amazon CloudWatch y permite la trazabilidad y LangFuse la depuración desde la prueba de concepto hasta la producción. AgentOps
- Retorno de la inversión (ROI) demostrado: los primeros proyectos piloto muestran mejoras importantes: una ejecución un 70 por ciento más rápida para un proyecto de modernización de código de gran tamaño y una reducción de aproximadamente un 90 por ciento en el tiempo de procesamiento para un flujo administrativo de CPG.

AutoGen

AutoGenes un marco de código abierto que fue lanzado inicialmente por. Microsoft AutoGense centra en habilitar agentes de IA autónomos conversacionales y colaborativos. Proporciona una arquitectura flexible para crear sistemas multiagente, con énfasis en las interacciones asincrónicas y basadas en eventos entre los agentes para flujos de trabajo autónomos complejos.

Características clave de AutoGen

AutoGenproporciona las siguientes características clave:

- Agentes conversacionales: se basan en conversaciones en lenguaje natural entre agentes autónomos, lo que permite un razonamiento sofisticado a través del diálogo. Para obtener más información, consulte el marco de Multi-agent conversación en la AutoGen documentación.

- Arquitectura asíncrona: Event-driven diseño para interacciones de agentes autónomos sin bloqueo, que admite flujos de trabajo paralelos complejos. Para obtener más información, consulte Resolución de varias tareas en una secuencia de chats asíncronos en la documentación. AutoGen
- Human-in-the-loop— Un firme apoyo a la participación humana opcional en los flujos de trabajo de los agentes, que de otro modo serían autónomos, cuando sea necesario. Para obtener más información, consulte Permitir la retroalimentación humana en los agentes en la AutoGen documentación.
- Generación y ejecución de código: capacidades especializadas para agentes autónomos centrados en el código que pueden escribir y ejecutar código. Para obtener más información, consulte la sección Ejecución de código en la AutoGen documentación.
- Comportamientos personalizables: configuración flexible de agentes autónomos y control de conversaciones para diversos casos de uso. Para obtener más información, consulte agentchat.conversable_agent en la documentación. AutoGen
- Selección de modelos de base: Support para varios modelos de base, incluidos los modelos Anthropic Claude, Amazon Nova (Premier, Pro, Lite y Micro) en Amazon Bedrock y otros para diferentes capacidades de razonamiento autónomo. Para obtener más información, consulte Configuración de LLM en la AutoGen documentación.
- Integración de la API LLM: configuración estandarizada para múltiples interfaces de servicios LLM, incluidas Amazon Bedrock, yOpenAI. Azure OpenAI Para obtener más información, consulte oai.openai_utils en la referencia de la API. AutoGen
- Procesamiento multimodal: Support para el procesamiento de texto e imágenes para permitir ricas interacciones multimodales entre agentes autónomos. Para obtener más información, consulte Cómo utilizar modelos multimodales: GPT-4V AutoGen en la documentación. AutoGen

Cuándo se debe usar AutoGen

AutoGenes especialmente adecuado para escenarios de agentes autónomos, que incluyen:

- Aplicaciones que requieren flujos de conversación naturales entre agentes autónomos para un razonamiento complejo
- Proyectos que requieren tanto un funcionamiento totalmente autónomo como capacidades opcionales de supervisión humana
- Casos de uso que implican la generación, ejecución y depuración de código autónomas sin intervención humana

- Escenarios que requieren patrones de comunicación entre agentes autónomos, asíncronos y flexibles

Enfoque de implementación para AutoGen

AutoGenproporciona un enfoque de implementación conversacional para las partes interesadas de la empresa, tal como se detalla en Primeros pasos en la AutoGen documentación. El marco permite a las organizaciones:

- Cree agentes autónomos que se comuniquen a través de conversaciones en lenguaje natural.
- Implemente interacciones asincrónicas y basadas en eventos entre varios agentes.
- Combine un funcionamiento totalmente autónomo con la supervisión humana opcional cuando sea necesario.
- Desarrolle agentes especializados para diferentes funciones empresariales que colaboren a través del diálogo.

Este enfoque conversacional hace que el razonamiento del sistema autónomo sea transparente y accesible para los usuarios empresariales. Decision-makers puede observar el diálogo entre los agentes para comprender cómo se llega a las conclusiones y, opcionalmente, participar en la conversación cuando se requiere el juicio humano.

Real-world ejemplo de AutoGen

Magentic-Onees un sistema multiagente generalista y de código abierto diseñado para resolver de forma autónoma tareas complejas y de varios pasos en diversos entornos, tal y como se describe en el blog AI Frontiers. Microsoft En esencia, está el agente Orchestrator, que descompone los objetivos de alto nivel y realiza un seguimiento del progreso mediante libros de contabilidad estructurados. Este agente delega las subtareas en agentes especializados (comoWebSurfer, yComputerTerminal) y se adapta de forma dinámica FileSurfer Coder replanificándolas cuando es necesario.

El sistema se basa en la AutoGen estructura y es independiente del modelo; por defecto, es GPT-4o. Logra un rendimiento de última generación en puntos de referencia como, y todo ello sin necesidad de ajustes específicos para cada tarea. GAIA AssistantBench WebArena Además, apoya la extensibilidad modular y la evaluación rigurosa mediante sugerencias. AutoGenBench

LlamaIndex

LlamaIndexes un marco de datos diseñado específicamente para conectar grandes modelos de lenguaje (LLM) con fuentes de datos externas a fin de permitir sofisticadas aplicaciones de recuperación, generación aumentada (RAG) e inteligencia artificial agencial. El marco proporciona abstracciones y flujos de trabajo de desarrollo acelerados para sistemas de agencias, patrones de orquestación personalizados e integraciones de sistemas que reducen el tiempo de producción de soluciones de IA basadas en el conocimiento.

Características clave de LlamaIndex

LlamaIndexproporciona un conjunto completo de capacidades que lo hacen especialmente adecuado para aplicaciones de IA de agencias empresariales:

- Data-centric arquitectura: se destaca a la hora de ingerir, indexar y recuperar información de más de 100 formatos de datos, incluidos archivos PDF, documentos de Word, Microsoft hojas de cálculo y más. El marco transforma los datos empresariales en bases de conocimiento consultables y optimizadas para los agentes de IA. Para obtener más información, consulte la Documentación de LlamaIndex.
- Production-ready despliegue: LlamaIndex ofrece tanto marcos de código abierto como servicios gestionados, y proporciona funciones de nivel empresarialLlamaCloud, como controles de seguridad, escalabilidad, integraciones de observabilidad y flexibilidad de despliegue. Para obtener más información, consulte la documentación del marco. LlamaIndex
- Procesamiento avanzado de documentos: LlamaCloud proporciona funciones de análisis, extracción, indexación y recuperación de documentos que permiten gestionar diseños complejos, tablas anidadas, contenido multimodal e incluso notas manuscritas. Este sofisticado análisis permite a los agentes trabajar eficazmente con documentos empresariales reales que contienen gráficos, diagramas y formatos complejos. Para obtener más información, consulte la Documentación de LlamaCloud.
- Orquestación de flujos de trabajo: LlamaAgents proporciona un motor de orquestación asíncrono basado en eventos para crear sistemas de agentes de varios pasos. Los flujos de trabajo admiten patrones complejos que incluyen bucles, ejecución paralela, ramificación condicional y reanudación con estado, lo que los hace ideales para interacciones sofisticadas entre agentes. Para obtener más información, consulte la documentación de los flujos de trabajoLlamaIndex.
- Capacidades de recuperación de agentes: modos de recuperación avanzados que incluyen búsqueda híbrida, búsqueda semántica y enrutamiento automático que determinan de

manera inteligente la mejor estrategia de recuperación para cada consulta. El marco admite la recuperación compuesta en múltiples bases de conocimiento y se reclasifica para mejorar la precisión. Para obtener más información, consulte la documentación del LlamaIndex RAG.

- Observabilidad y evaluación: LlamaIndex se integra con una variedad de herramientas de observabilidad y evaluación. Esta capacidad de integración le ayuda a rastrear y depurar sus aplicaciones, evaluar su rendimiento y monitorear los costos. Para obtener más información, consulte la documentación sobre rastreo, depuración y evaluación. LlamaIndex

Cuándo se debe usar LlamaIndex

LlamaIndexes especialmente adecuado para escenarios de IA de agencias que hacen hincapié en los flujos de trabajo con uso intensivo de datos y la gestión del conocimiento:

- Document-heavy aplicaciones que requieren que los agentes procesen, analicen y extraigan información de grandes volúmenes de documentos empresariales, como contratos, informes, manuales y documentos reglamentarios
- Creación rápida de prototipos para escenarios de producción en los que las organizaciones desean crear e implementar rápidamente agentes centrados en los documentos sin una sobrecarga excesiva de administración de la infraestructura
- RAG-first arquitecturas que priorizan la precisión de la recuperación y la relevancia del contexto, especialmente cuando se trabaja con documentos complejos y multimodales que contienen tablas, imágenes y datos estructurados
- Multi-agent flujos de trabajo de documentos que requieren agentes especializados para diferentes aspectos del procesamiento de documentos, como el análisis, el resumen y la comprobación de la conformidad

Enfoque de implementación para LlamaIndex

LlamaIndex proporciona componentes básicos de bajo nivel y abstracciones de alto nivel que se adaptan a diferentes enfoques de implementación:

- Desarrollo rápido de aplicaciones RAG funcionales en solo unas pocas líneas de código mediante LlamaIndex el uso de API de alto nivel. Este enfoque hace que sea LlamaIndex accesible para los equipos empresariales y los desarrolladores que recién comienzan a usar la IA de los agentes.

- Integración empresarial mediante LlamaHub sistemas empresariales populares SharePoint, como Amazon Simple Storage Service (Amazon S3), bases de datos y API. Este enfoque permite una integración perfecta con la infraestructura de datos existente.
- Opciones de implementación flexibles entre despliegues autohospedados de código abierto para un control máximo o servicios LlamaCloud gestionados para reducir los gastos operativos y funciones empresariales.
- Las aplicaciones pueden empezar con motores de consultas simples y añadir progresivamente capacidades de agente, organización de múltiples agentes y flujos de trabajo complejos a medida que evolucionan los requisitos.

Real-world ejemplo de LlamaIndex

Este ejemplo se centra en una filial de una empresa aeroespacial que se especializa en soluciones de navegación y operaciones de aviación. Deben abordar un desafío cada vez mayor que implica poner a prueba pruebas de chatbots de IA no coordinadas. Las pruebas dieron lugar a la repetición del trabajo, a largos ciclos de desarrollo, a obstáculos en materia de cumplimiento y a implementaciones aisladas en toda la organización.

Desarrollaron un marco de agentes unificado, una solución reutilizable basada en plantillas basada en un marco de LlamaIndex código abierto que hace que la creación de agentes sea mucho más eficiente. Compararon varios marcos de la competencia, tanto orientados a cadenas como basados en gráficos. En última instancia, la eligieron LlamaIndex por tres ventajas fundamentales: su diseño flexible, sus componentes modulares y sus controles de orquestación listos para la producción.

La plataforma reduce el tiempo de desarrollo e implementación de los agentes en un 87%, de 512 a 64 horas. Esta reducción se logró al permitir a los equipos crear agentes con aproximadamente 50 líneas de código y un archivo de configuración JSON. Los equipos utilizaron un marco unificado con seguridad, conformidad y acceso privilegiado al sistema integrados. Para obtener más información, consulte los estudios de casos de LlamaIndex clientes.

Comparación de los marcos de IA de las agencias

Al seleccionar un marco de inteligencia artificial para el desarrollo de agentes autónomos, tenga en cuenta cómo se ajusta cada opción a sus requisitos específicos. Tenga en cuenta no solo sus capacidades técnicas, sino también su adecuación organizativa, incluida la experiencia del equipo, la infraestructura existente y los requisitos de mantenimiento a largo plazo. Muchas organizaciones

podrían beneficiarse de un enfoque híbrido, que aproveche múltiples marcos para diferentes componentes de su ecosistema de IA autónomo.

En la siguiente tabla se comparan los niveles de madurez (más sólidos, adecuados o débiles) de cada marco en función de las dimensiones técnicas clave. Para cada marco, la tabla también incluye información sobre las opciones de implementación en producción y la complejidad de la curva de aprendizaje.

Plataforma	AWS integration	Soporte multiagente autónomo	Complejidad del flujo de trabajo	Capacidades multimodales	Selección del modelo básico	Integración de la API LLM	Despliegue de producción	Curva de aprendizaje
AutoGen	Débil	Fuerte	Fuerte	Adecuado	Adecuado	Fuerte	Hágalo usted mismo (DIY)	Empapado
CrewAI	Débil	Fuerte	Adecuado	Débil	Adecuado	Adecuado	BRICOLAJE	Moderado
LangChain / LangGraph	Adecuado	Fuerte	Más fuerte	Más fuerte	Más fuerte	Más fuerte	Plataforma o bricolaje	Acero
LlamaIndex	Adecuado	Adecuado	Fuerte	Adecuado	Fuerte	Fuerte	Plataforma o bricolaje	Moderado
Strands Agents	El más fuerte	Fuerte	Más fuerte	Fuerte	Fuerte	Más fuerte	BRICOLAJE	Moderado

Consideraciones a la hora de elegir un marco de IA para agencias

Al desarrollar agentes autónomos, tenga en cuenta los siguientes factores clave:

- AWS integración de la infraestructura: las organizaciones en las que se invierta mucho AWS se beneficiarán más de las integraciones nativas o de Strands Agents los flujos Servicios de AWS de trabajo autónomos. Para obtener más información, consulte el resumen AWS semanal (AWS blog).
- Selección del modelo de base: considere qué marco proporciona el mejor soporte para sus modelos de base preferidos (por ejemplo, los modelos Amazon Nova en Amazon Bedrock o Anthropic Claude), en función de los requisitos de razonamiento de su agente autónomo. Para obtener más información, consulte Cómo crear agentes eficaces en el sitio Anthropic web.
- Integración de la API LLM: evalúe los marcos en función de su integración con las interfaces de servicio del modelo de lenguaje grande (LLM) preferidas (por ejemplo, Amazon Bedrock oOpenAI) para la implementación en producción. Para obtener más información, consulte las interfaces de modelo en la documentación. Strands Agents
- Requisitos multimodales: para los agentes autónomos que necesitan procesar texto, imágenes y voz, tenga en cuenta las capacidades multimodales de cada marco. Para obtener más información, consulte Multimodalidad en la documentación. LangChain
- Complejidad del flujo de trabajo autónomo: los flujos de trabajo autónomos más complejos con una administración de estado sofisticada podrían favorecer las capacidades avanzadas de las máquinas de estados. LangGraph
- Colaboración autónoma en equipo: los proyectos que requieren una colaboración autónoma explícita y basada en roles entre agentes especializados pueden beneficiarse de la arquitectura orientada al equipo de. CrewAI
- Paradigma de desarrollo autónomo: los equipos que prefieran patrones conversacionales y asíncronos para los agentes autónomos podrían preferir la arquitectura basada en eventos de. AutoGen
- Enfoque gestionado o basado en código: las organizaciones que desean una experiencia totalmente gestionada con un mínimo de codificación deberían considerar Amazon Bedrock Agents. Las organizaciones que requieren una personalización más profunda pueden preferir Strands Agents otros marcos con capacidades especializadas que se ajusten mejor a los requisitos específicos de los agentes autónomos.
- Preparación para la producción de sistemas autónomos: considere las opciones de implementación, las capacidades de monitoreo y las funciones empresariales para los agentes autónomos de producción.

Plataformas

Las plataformas de inteligencia artificial de las agencias proporcionan las capas fundamentales de tiempo de ejecución, orquestación e integración necesarias para implementar, escalar y gestionar sistemas de agentes de producción. Los marcos definen cómo se crean los agentes y los protocolos rigen la forma en que se comunican. Las plataformas proporcionan el entorno en el que estos agentes operan, colaboran y evolucionan de forma segura y a escala.

Las plataformas de los agentes combinan la ejecución de modelos, la gestión del contexto, la integración de herramientas, la observabilidad y las capacidades de gobierno en entornos unificados. Estas plataformas permiten a las organizaciones pasar de la experimentación a la implementación a escala empresarial.

En esta sección:

Por qué son importantes las plataformas

Las plataformas de IA de los agentes son fundamentales para las organizaciones que buscan poner en funcionamiento los sistemas autónomos en la producción. Ofrecen las siguientes capacidades:

- Proporcionan una organización del tiempo de ejecución para alojar, escalar y coordinar los agentes.
- Gestione el estado, el contexto y la memoria en los flujos de trabajo de varios agentes.
- Ofrezca controles de seguridad, identidad y gobierno alineados con los estándares empresariales.
- Intégrelo con ecosistemas de herramientas y sistemas externos mediante estándares APIs o protocolos.
- Habilite la observabilidad y la auditabilidad en todas las interacciones de los agentes y los flujos de eventos.

- Support la interoperabilidad entre modelos, lo que permite a los agentes utilizar varios modelos básicos en un único entorno.

Estas capacidades convierten a los agentes individuales en sistemas coordinados y adaptables que pueden funcionar de manera confiable dentro de los límites empresariales y regulatorios.

Tipos de plataformas de IA para agentes

Las plataformas de IA de los agentes suelen clasificarse en una o más de las siguientes categorías:

- Agente gestionado: las plataformas totalmente gestionadas proporcionan capacidades integradas de infraestructura, memoria y orquestación. Reducen la sobrecarga operativa y aceleran el tiempo de producción.
- Organización de código abierto: las plataformas de agencias de código abierto ofrecen flexibilidad y transparencia a las organizaciones que prefieren entornos personalizables o la implementación local.
- Empresa híbrida: las plataformas híbridas integran componentes gestionados y autohospedados, y combinan la escalabilidad de los servicios gestionados en la nube con el control de los sistemas empresariales.

Consideraciones sobre la selección de plataformas

Al seleccionar o diseñar una plataforma de IA para agencias, las organizaciones deben tener en cuenta lo siguiente:

- Profundidad de integración: evalúe qué tan bien se integra la plataforma con las fuentes de datos, las herramientas y los protocolos existentes.
- Escalabilidad: asegúrese de que la plataforma pueda escalar dinámicamente para soportar cargas de trabajo autónomas y la colaboración entre múltiples agentes.
- Seguridad y conformidad: evalúe las características de privacidad, cifrado y gobierno de los datos en función de los requisitos organizativos y regionales.
- Extensibilidad: elija plataformas con arquitecturas modulares que permitan añadir nuevas herramientas, modelos o agentes a lo largo del tiempo.
- Observabilidad: prefiera plataformas que proporcionen registros detallados de telemetría, trazabilidad y auditoría para las interacciones entre los agentes.

- Rentabilidad: considere modelos sin servidor o basados en el uso para optimizar el costo de las cargas de trabajo variables.

Agentes de Amazon Bedrock

Amazon Bedrock Agents es un servicio totalmente gestionado que le permite crear y configurar agentes autónomos en sus aplicaciones. Puede organizar las interacciones entre los modelos básicos, las fuentes de datos, las aplicaciones de software y las conversaciones de los usuarios. Su enfoque simplificado para crear agentes no requiere aprovisionar capacidad, administrar la infraestructura ni escribir código personalizado.

Características principales de Amazon Bedrock Agents

Amazon Bedrock Agents incluye las siguientes funciones clave:

- Servicio totalmente gestionado: gestión completa de la infraestructura sin necesidad de aprovisionar capacidad ni gestionar los sistemas subyacentes. Para obtener más información, consulte Automatizar tareas en su aplicación mediante agentes de IA en la documentación de Amazon Bedrock.
- Desarrollo basado en API: defina y ejecute agentes mediante sencillas llamadas a la API especificando modelos, instrucciones, herramientas y parámetros de configuración. Para obtener más información, consulte Crear y configurar el agente manualmente en la documentación de Amazon Bedrock.
- Grupos de acciones: defina las acciones específicas que su agente puede realizar mediante la creación de grupos de acciones con esquemas de API. Para obtener más información, consulte Utilizar grupos de acciones para definir las acciones que debe realizar su agente en la documentación de Amazon Bedrock.
- Integración de la base de conocimientos: conéctese sin problemas a las bases de conocimiento de Amazon Bedrock para aumentar las respuestas de los agentes con los datos de su organización. Para obtener más información, consulte Aumente la generación de respuestas para su agente con la base de conocimientos en la documentación de Amazon Bedrock.
- Plantillas de solicitudes avanzadas: personalice el comportamiento de los agentes mediante plantillas de solicitudes para el preprocesamiento, la organización, la generación de respuestas a la base de conocimientos y el posprocesamiento. Para obtener más información, consulte Mejorar la precisión de los agentes mediante plantillas de mensajes avanzadas en Amazon Bedrock en la documentación de Amazon Bedrock.

- Rastreo y observabilidad: realice un seguimiento del proceso de step-by-step razonamiento del agente mediante las funciones de rastreo integradas. Para obtener más información, consulte el proceso de step-by-step razonamiento de un agente mediante el rastreo en la documentación de Amazon Bedrock.
- Control de versiones y alias: cree varias versiones de su agente y despliéguelas mediante alias para una implementación controlada. Para obtener más información, consulte Implementación y uso de un agente de Amazon Bedrock en su aplicación en la documentación de Amazon Bedrock.

Cuándo usar Amazon Bedrock Agents

Amazon Bedrock Agents es especialmente adecuado para escenarios de agentes autónomos, que incluyen:

- Organizaciones que desean una experiencia totalmente gestionada para crear e implementar agentes sin administrar la infraestructura
- Proyectos que requieren un rápido desarrollo y despliegue de agentes mediante la configuración en lugar de mediante el código
- Casos de uso que se benefician de una estrecha integración con otras capacidades de Amazon Bedrock, como Knowledge Bases y Guardrails
- Los equipos no cuentan con los recursos internos necesarios para crear agentes partiendo de cero, pero necesitan capacidades autónomas listas para la producción

Enfoque de implementación para Amazon Bedrock Agents

Amazon Bedrock Agents ofrece un enfoque de implementación basado en la configuración para las partes interesadas de la empresa. El servicio permite a las organizaciones:

- Defina los agentes mediante llamadas a la API Consola de administración de AWS o a la API sin escribir código complejo.
- Cree grupos de acciones que especifiquen APIs las operaciones que el agente puede realizar.
- Connect bases de conocimiento para proporcionar información específica del dominio al agente.
- Pruebe e itere el comportamiento del agente a través de una interfaz visual.

Este enfoque gestionado permite a los equipos empresariales desarrollar y desplegar rápidamente agentes autónomos sin necesidad de una amplia experiencia técnica en el desarrollo de modelos de IA o en la gestión de infraestructuras.

Ejemplo real de Amazon Bedrock Agents

Una solución de operaciones financieras (FinOps) descrita en esta entrada de AWS blog utiliza el marco multiagente de Amazon Bedrock para crear un asistente de gestión de costes en la nube impulsado por la IA. El rentable modelo básico de Amazon Nova potencia la solución, en la que un agente FinOps supervisor central delega tareas en agentes especializados. Estos agentes recopilan y analizan los datos de AWS gastos mediante el uso AWS Cost Explorer y generan recomendaciones de ahorro de costes mediante el uso. AWS Trusted Advisor

El sistema incluye el acceso seguro de los usuarios a través de Amazon Cognito, una interfaz alojada en AWS Amplify, y grupos de AWS Lambda acción para realizar análisis y pronósticos en tiempo real. Los equipos financieros pueden hacer preguntas en lenguaje natural, como «¿Cuáles fueron mis costes en febrero de 2025?» El sistema responde con desgloses detallados, sugerencias de optimización y previsiones, todo ello dentro de una arquitectura escalable y sin servidores que se implementa mediante el uso. AWS CloudFormation

Amazon Bedrock AgentCore

Amazon Bedrock AgentCore es una plataforma de agencia para crear, implementar y operar agentes de alta capacidad de forma segura y a escala mediante cualquier marco, modelo o protocolo. Con ella AgentCore, puede hacer lo siguiente, sin necesidad de administrar la infraestructura:

- Cree agentes más rápido.
- Permita que los agentes tomen medidas con respecto a las herramientas y los datos.
- Ejecute los agentes de forma segura con tiempos de ejecución prolongados y de baja latencia.
- Supervise los agentes en producción.

AgentCore elimina el trabajo pesado e indiferenciado que supone crear una infraestructura de agentes especializada, lo que le permite acelerar el paso de sus agentes a la producción. Sus servicios se pueden usar juntos o de forma independiente y son compatibles con cualquier marco, incluidosCrewAI, LangGraphLlamaIndex, yStrands Agents. AgentCoretambién es compatible con cualquier modelo de base que esté disponible dentro o fuera de Amazon Bedrock, lo que proporciona la máxima flexibilidad.

AgentCore se compone de varios servicios clave:

- Amazon Bedrock AgentCore Runtime: proporciona un entorno seguro, escalable y sin servidores para alojar y ejecutar sus agentes, sin necesidad de administrar la infraestructura necesaria para implementar y ejecutar agentes o herramientas de IA.
- Amazon Bedrock AgentCore Memory: ofrece un sistema de memoria gestionada que permite a los agentes retener el contexto de las interacciones para mantener conversaciones más personalizadas y coherentes al mantener el conocimiento inmediato y a largo plazo.
- Amazon Bedrock AgentCore Gateway: simplifica el proceso de creación, protección y búsqueda de las herramientas adecuadas para los agentes. Con AgentCore Gateway, los desarrolladores pueden convertir APIs las funciones de Lambda y los servicios existentes en herramientas compatibles con el Model Context Protocol (MCP) y ponerlos a disposición de los agentes.
- Amazon Bedrock AgentCore Identity: proporciona un servicio de administración de acceso e identidad de agentes seguro y escalable que acelera el desarrollo de agentes de IA. Con AgentCore Identity, puede asignar identidades únicas y verificables a los agentes, lo que permite un control de acceso detallado y asegura las interacciones impulsadas por los agentes con los sistemas empresariales.
- Herramientas AgentCore integradas de Amazon Bedrock: le permite utilizar las herramientas integradas para mejorar su flujo de trabajo de desarrollo y pruebas. Utilice estas herramientas para interactuar con su aplicación de forma eficaz, lo que permitirá a los agentes de IA escribir y ejecutar código de forma segura en entornos aislados. Utilice la herramienta del navegador para permitir que los agentes de IA interactúen con los sitios web a gran escala.
- Amazon Bedrock AgentCore Observability: ofrece funciones de registro y supervisión, lo que le proporciona visibilidad en tiempo real del rendimiento y el comportamiento de su agente para facilitar la depuración y la optimización.

Características principales de AgentCore

AgentCore incluye las siguientes características clave:

- Totalmente gestionado y ampliable: AgentCore es un servicio totalmente gestionado, lo que significa que AWS gestiona la infraestructura subyacente y el mantenimiento. También es extensible, lo que le permite personalizar y mejorar la funcionalidad de sus agentes. Para obtener más información, consulte Introducción a AgentCore Runtime en la AgentCore documentación.

- Memoria a corto y largo plazo: ofrezca interacciones más personalizadas y relevantes al equipar a los agentes con un sistema de memoria para recordar el contexto de las conversaciones actuales y los conocimientos a largo plazo. Para obtener más información, consulte Introducción a la AgentCore memoria en la AgentCore documentación.
- Desarrollo e integración simplificados de herramientas: permita a sus agentes descubrir y utilizar las herramientas a través de un único punto final seguro. Convierta rápidamente sus recursos empresariales existentes en herramientas listas para los agentes con solo unas pocas líneas de código, lo que permitirá a los desarrolladores centrarse en desarrollar capacidades únicas. Para obtener más información, consulte Introducción a AgentCore Gateway en la documentación. AgentCore
- Infraestructura segura y escalable: AgentCore proporciona un entorno seguro y escalable para implementar y operar agentes. Incluye funciones para la administración de identidades y accesos, el cifrado de datos y la seguridad de la red. Para obtener más información, consulte Introducción a la AgentCore identidad en la AgentCore documentación.
- Integración con una amplia gama de herramientas: le permite integrar a sus agentes con una variedad de herramientas, como un intérprete de código y una herramienta de navegador que puede crear con las herramientas AgentCore integradas. Para obtener más información, consulte Introducción a AgentCore Code Interpreter y Introducción al AgentCore navegador en la AgentCore documentación.
- Observabilidad y monitoreo integrales: obtenga una visibilidad profunda de sus agentes con herramientas integrales para rastrear, depurar y monitorear su desempeño en producción. Visualice toda la ruta de ejecución del agente para auditar su razonamiento y resolver los errores. Utilice paneles de control en tiempo real y datos de telemetría estandarizados para realizar un seguimiento de las principales métricas operativas. Para obtener más información, consulte Añadir observabilidad a sus AgentCore recursos de Amazon Bedrock en la AgentCore documentación.

¿Cuándo se debe usar AgentCore

AgentCore es especialmente adecuado para escenarios de agentes autónomos, que incluyen:

- Organizaciones que desean acelerar el desarrollo y reducir los gastos operativos con un servicio totalmente gestionado que gestiona la infraestructura, la seguridad, las herramientas integradas, la observabilidad y el escalado

- Proyectos que necesitan flexibilidad con servicios modulares que funcionen juntos o de forma independiente y que sean compatibles con cualquier marco, similar CrewAI oLangGraph, y con cualquier modelo básico de cualquier fuente
- Casos de uso que requieren agentes conversacionales y activos que deben mantener el contexto y aprender de las interacciones pasadas para ofrecer respuestas personalizadas y relevantes
- Los agentes pueden realizar tareas complejas mediante una integración sencilla con diversas aplicaciones, fuentes de datos y APIs

Enfoque de implementación para AgentCore

AgentCore está diseñado para las organizaciones que desean que los agentes de IA pasen de la fase de prueba de concepto, creada con marcos de agentes personalizados o de código abierto, a la producción. Con AgentCore, las organizaciones pueden hacer lo siguiente:

- Implemente agentes de forma segura en una infraestructura sin servidor, compatible con cualquier marco y modelo, con aislamiento de sesiones y administración de identidad y acceso integrada para garantizar la end-to-end seguridad y el cumplimiento. Cree rápidamente agentes en AgentCore tiempo de ejecución para los principales marcos de agentes mediante el kit de herramientas básico.
- Mejore los agentes integrando la memoria persistente para retener el contexto, simplificando el desarrollo y la integración de las herramientas a través AgentCore de Gateway. Aproveche la herramienta de navegador y el intérprete de código integrados para flujos de trabajo avanzados.
- Rastree, depure y supervise los agentes de IA en producción mediante paneles de observabilidad impulsados por Amazon CloudWatch Application Insights y OpenTelemetry realizando un seguimiento de las métricas clave de AgentCore los recursos (tiempo de ejecución, memoria, puerta de enlace y herramientas).
- Acelere la implementación y la innovación con servicios modulares y totalmente gestionados, que pueden componerse en bloques juntos o de forma independiente, con cualquier marco de agentes y proveedor de modelos. Esta flexibilidad ayuda a las organizaciones a pasar del prototipo a la producción con mayor rapidez.

Este enfoque gestionado permite a las organizaciones crear, implementar y ejecutar de forma rápida y segura agentes de IA y sistemas multiagente de nivel empresarial a cualquier escala.

Ejemplo real de AgentCore

AWS ha observado que uno de los bancos más grandes de América Latina lleva AI/ML años ofreciendo una experiencia de banca digital hiperpersonalizada y segura. El banco está ampliando los servicios de inteligencia artificial AgentCore para ofrecer a los clientes interacciones intuitivas, mayor seguridad y mayor automatización. Según el CTO, AgentCore se espera que apoye sus esfuerzos para cumplir los compromisos con los clientes a gran escala. AgentCore proporciona a sus desarrolladores las herramientas y la flexibilidad necesarias para crear y gestionar agentes, al tiempo que ayuda a garantizar el cumplimiento de las normas financieras.

Protocolos

Los agentes de IA requieren protocolos de comunicación estandarizados para interactuar con otros agentes y servicios. Las organizaciones que implementan arquitecturas de agentes se enfrentan a importantes desafíos en torno a la interoperabilidad, la independencia de los proveedores y la preparación de sus inversiones para el futuro.

Esta sección le ayuda a navegar por el panorama de los agent-to-agent protocolos centrándose en los estándares abiertos que maximizan la flexibilidad y la interoperabilidad. (Para obtener información sobre agent-to-tool los protocolos, consulte Estrategia de integración de herramientas más adelante en esta guía).

En esta sección se destaca el Model Context Protocol (MCP), un estándar abierto desarrollado originalmente Anthropic en 2024. En la actualidad, apoya AWS activamente al MCP mediante contribuciones al desarrollo e implementación del protocolo. AWS colabora con los principales marcos de agentes de código abierto, incluidos LangGraph CrewAILlamaIndex, y para dar forma al futuro de la comunicación entre agentes en el protocolo. Para obtener más información, consulte Protocolos abiertos para la interoperabilidad de los agentes, parte 1: Comunicación entre agentes en el MCP (blog).AWS

En esta sección:

- Por qué es importante la selección de protocolos
- Agent-to-agent protocolos
- Selección de protocolos de agencia
- Estrategia de implementación de los protocolos de los agentes
- Cómo empezar con MCP
- ???

¿Por qué es importante la selección de protocolos

La selección de protocolos determina de manera fundamental la forma en que puede crear y evolucionar su arquitectura de agentes de IA. Al elegir protocolos que admitan la portabilidad entre marcos de agentes, obtiene la flexibilidad necesaria para combinar diferentes sistemas de agentes y flujos de trabajo para satisfacer sus necesidades específicas.

Los protocolos abiertos le permiten integrar agentes en varios marcos. Por ejemplo, utilícelos LangChain para la creación rápida de prototipos e implemente sistemas de producción que se comuniquen a través de un protocolo común, como el MCP o el protocolo Agent2Agent (A2A). Strands Agents Esta flexibilidad reduce la dependencia de proveedores de IA específicos, simplifica la integración con los sistemas existentes y permite mejorar las capacidades de los agentes a lo largo del tiempo.

Los protocolos bien diseñados también establecen patrones de seguridad consistentes para la autenticación y la autorización en todo el ecosistema de agentes. Y lo que es más importante, la portabilidad de los protocolos preserva la libertad de adoptar nuevos marcos y capacidades de agentes a medida que vayan surgiendo. La elección de protocolos abiertos protege su inversión en el desarrollo de agentes y, al mismo tiempo, mantiene la interoperabilidad con sistemas de terceros.

Ventajas de los protocolos abiertos

Al implementar sus propias extensiones o crear sistemas de agentes personalizados, los protocolos abiertos ofrecen ventajas convincentes:

- Documentación y transparencia: normalmente proporcionan documentación completa e implementaciones transparentes
- Soporte comunitario: acceso a comunidades de desarrolladores más amplias para la solución de problemas y las mejores prácticas
- Garantías de interoperabilidad: mayor garantía de que sus extensiones funcionarán en diferentes implementaciones
- Compatibilidad futura: se reduce el riesgo de que se produzcan cambios importantes o de que queden obsoletos
- Influencia en el desarrollo: oportunidad de contribuir a la evolución del protocolo

Agent-to-agent protocolos

La siguiente tabla proporciona una descripción general de los protocolos de los agentes que permiten a varios agentes colaborar, delegar tareas y compartir información.

Protocolo	Ideal para	Consideraciones

Comunicación entre agentes MCP	Organizaciones que buscan patrones flexibles de colaboración entre agentes	• Una extensión del Protocolo de Contexto Modelo (MCP) propuesta por él AWS que se basa en su base de agent-to-agent comunicac ión existente • Permite una colaboración fluida entre los agentes con una seguridad OAuth basada en la seguridad
Protocolo A2A	Ecosistemas de agentes multiplataforma	• Respaldado por Google • Estándar más nuevo con una adopción más limitada en comparación con el MCP

Decidir entre las opciones de protocolo

Al implementar la agent-to-agent comunicación, haga coincidir sus requisitos de comunicación específicos con las capacidades de protocolo adecuadas. Los diferentes patrones de interacción requieren diferentes características de protocolo. En la siguiente tabla se describen los patrones de comunicación más comunes y se recomiendan las opciones de protocolo más adecuadas para cada escenario.

Patrón	Descripción	La elección de protocolo ideal
Solicitud y respuesta sencillas	Interacciones puntuales entre agentes	MCP con flujos sin estado
Diálogos sensatos	Conversaciones continuas con el contexto	MCP con gestión de sesiones
Colaboración entre múltiples agentes	Interacciones complejas entre varios agentes	Interagente MCP o AutoGen

Flujos de trabajo basados en equipos	Equipos de agentes jerárquicos con funciones definidas	Interagente MCP, o CrewAI AutoGen

Más allá de los patrones de comunicación, varios factores técnicos y organizativos pueden influir en la selección del protocolo. En la siguiente tabla se describen las consideraciones clave que pueden ayudarle a evaluar qué protocolo se ajusta mejor a sus requisitos de implementación específicos.

Consideración	Descripción	Ejemplo
Modelo seguridad	Requisitos de autenticación y autorización	OAuth 2.0 en MCP
Entorno de despliegue	Donde los agentes correrán y se comunicarán	Máquina distribuida o única
Compatibilidad con los ecosistemas	Integración con los marcos de agentes existentes	LangChain o Strands Agents
Necesidades de escalabilidad	Crecimiento esperado de las interacciones entre los agentes	Capacidades de transmisión de MCP

Selección de los protocolos de los agentes

Para la mayoría de las organizaciones que crean sistemas de agentes de producción, el Model Context Protocol (MCP) ofrece la base de comunicación más completa y mejor respaldada. agent-to-agent MCP se beneficia de las contribuciones activas al desarrollo AWS y de la comunidad de código abierto.

Seleccionar los protocolos de agencia correctos es importante para las organizaciones que buscan implementar la IA de manera efectiva. Las consideraciones difieren según el contexto organizacional.

Consideraciones sobre la selección del protocolo de la agencia

Las organizaciones deben tener en cuenta las siguientes mejores prácticas al seleccionar los protocolos para los sistemas de IA de las agencias:

- Priorice los estándares abiertos: las organizaciones deben adoptar protocolos abiertos como el MCP para garantizar la interoperabilidad y la extensibilidad a largo plazo y reducir el riesgo de dependencia de un proveedor.
- Equilibre la velocidad y la flexibilidad: las empresas emergentes y las primeras en adoptarlos pueden empezar con protocolos patentados bien compatibles para lograr un desarrollo rápido, pero deberían definir una ruta de migración hacia estándares abiertos a medida que los sistemas maduren.
- Implemente capas de abstracción: las empresas deben implementar la abstracción de protocolos para simplificar la migración, permitir la adopción híbrida y preparar estrategias de integración preparadas para el futuro.
- Haga hincapié en la seguridad y el cumplimiento: las organizaciones de los sectores regulados deben seleccionar protocolos con sólidas capacidades de autenticación, cifrado y auditoría para cumplir con los requisitos de gobierno y cumplimiento.
- Evalúe la madurez del ecosistema: todas las organizaciones deben evaluar el estado, la adopción y el apoyo de la comunidad a cada protocolo para garantizar la sostenibilidad y minimizar la deuda técnica.
- Participar en el desarrollo de estándares: las organizaciones deberían participar en organismos de normalización o comunidades de código abierto para ayudar a dar forma a la evolución de los protocolos e influir en las mejores prácticas.
- Tenga en cuenta la soberanía de los datos: el gobierno y los sectores regulados deben garantizar que las opciones de protocolo se ajusten a los requisitos de residencia y soberanía de los datos en todas las regiones de despliegue.
- Aproveche los servicios gestionados: siempre que sea posible, utilice implementaciones gestionadas o sin servidor de protocolos de agencia para reducir la complejidad operativa y acelerar el despliegue.

Estrategia de implementación de protocolos de agencia

Para implementar de manera efectiva los protocolos de los agentes en toda su organización, considere los siguientes pasos estratégicos:

1. Comience con la alineación de los estándares: adopte protocolos abiertos establecidos siempre que sea posible.
2. Cree capas de abstracción: implemente adaptadores entre sus sistemas y protocolos específicos.

3. Contribuya a los estándares abiertos: participe en las comunidades de desarrollo de protocolos.
4. Supervise la evolución de los protocolos: manténgase informado sobre las nuevas normas y actualizaciones.
5. Pruebe la interoperabilidad con regularidad: compruebe que sus implementaciones siguen siendo compatibles.

Cómo empezar con MCP

AWS apoya activamente el Protocolo de contexto modelo (MCP) mediante contribuciones al desarrollo e implementación del protocolo. AWS colabora con los principales marcos de agentes de código abierto, incluidos LangGraph CrewAILlamaIndex, y para dar forma al futuro de la comunicación entre agentes en el protocolo.

Para implementar el MCP en la arquitectura de sus agentes, lleve a cabo las siguientes acciones:

1. Explore las implementaciones del MCP en marcos como el SDK. Strands Agents
2. Revise la documentación técnica del Model Context Protocol.
3. Lea los protocolos abiertos para la interoperabilidad de los agentes, parte 1: Comunicación entre agentes en el MCP (AWS blog), para obtener más información sobre la interoperabilidad de los agentes.
4. Únase a la comunidad de MCP para influir en la evolución del protocolo.

El MCP proporciona una capa de comunicación que permite a los agentes interactuar con datos y servicios externos y también se puede utilizar para permitir que los agentes interactúen con otros agentes. La implementación de transporte HTTP Streamable del protocolo ofrece a los desarrolladores un conjunto completo de patrones de interacción sin tener que reinventar la rueda. Estos patrones admiten tanto los request/response flujos sin estado como la gestión de sesiones con estado de forma persistente. IDs

Al adoptar protocolos abiertos como el MCP, usted posiciona a su organización para crear sistemas de agentes que sigan siendo flexibles, interoperables y adaptables a medida que la tecnología de IA evoluciona. Para obtener información sobre la implementación de agent-to-tool protocolos, consulte la estrategia de integración de herramientas más adelante en esta guía.

Cómo empezar con A2A

El protocolo Agent2Agent (A2A) permite la colaboración descentralizada entre agentes a través de una capa semántica compartida. En lugar de dirigir todo el trabajo a través de un orquestador central, el A2A permite que los agentes se descubran entre sí, anuncien sus capacidades, negocien tareas y compartan el contexto mediante un protocolo ligero basado en JSON. Cada agente publica un manifiesto de capacidades.

El siguiente ejemplo muestra un manifiesto de capacidades A2A simplificado que anuncia las acciones respaldadas por un agente, las entradas requeridas y los metadatos operativos para permitir el descubrimiento y la negociación de tareas:

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

Este modelo permite la combinación dinámica de capacidades, la delegación a mitad de las tareas y la colaboración entre organizaciones. Los agentes pueden autoorganizarse en torno a las tareas, formar grupos de trabajo temporales y adaptarse a medida que nuevas capacidades entran o salen del sistema.

El A2A admite interacciones que van desde simples solicitudes sin estado hasta sesiones de negociación de varios pasos, que incluyen:

- peer-to-peerMensajería directa para una colaboración de baja latencia
- Negociación semántica de tareas, en la que los agentes seleccionan al compañero más adecuado
- Descubrimiento basado en capacidades, que posibilita una división emergente del trabajo
- Fijación de sesiones para interacciones estables de varios pasos

Al adoptar protocolos abiertos y nativos de los agentes, como el A2A, las organizaciones crean sistemas de IA modulares, interoperables y capaces de colaborar de forma transfronteriza. El A2A garantiza que los ecosistemas de agentes sigan siendo flexibles y puedan evolucionar a medida que se introduzcan nuevos agentes, equipos o sistemas externos, sin necesidad de capas de orquestación rígidas ni de un acoplamiento previo.

Para implementar el protocolo A2A en la arquitectura de sus agentes, lleve a cabo las siguientes acciones:

1. Revise la especificación del protocolo A2A: lea la última versión de la especificación del protocolo Agent2Agent (A2A) para saber cómo se manifiestan las capacidades, los flujos de negociación y el protocolo de contacto entre los agentes.
2. Explore los tiempos de ejecución compatibles con A2A: evalúe marcos como el SDK de Strands Agents o capas de tiempo de ejecución personalizadas que admitan las manifestaciones de capacidad y la negociación al estilo A2A. peer-to-peer
3. Implemente un manifiesto de capacidades para sus agentes: defina los `meta` campos y los de cada agente para facilitar la detección `canneeds`, el emparejamiento y la colaboración a nivel de intención.
4. Experimente con patrones de negociación A2A: utilice el ciclo de solicitud, oferta y aceptación, consultas de capacidad estructuradas o descubrimiento basado en chismes para comprender cómo los agentes razonan sobre quién debe encargarse de una tarea.
5. Pruebe el A2A en un entorno de infraestructura mixta: combine la negociación entre pares del A2A con el enrutamiento de eventos nativo a través de AWS Amazon EventBridge para evaluar los patrones de coordinación híbridos.
6. Únase a la comunidad de A2A: participe en el grupo de trabajo abierto para mantenerse al día con las ampliaciones, las recomendaciones de seguridad y las mejoras de interoperabilidad entre proveedores, y contribuya al desarrollo del protocolo.

Tools (Herramientas)

Los agentes de IA aportan valor al interactuar con herramientas y fuentes de datos externas para realizar tareas útiles. APIs La estrategia de integración de herramientas adecuada afecta directamente a las capacidades, la postura de seguridad y la flexibilidad a largo plazo del agente.

Esta sección le ayuda a navegar por el panorama de la integración de herramientas centrándose en los estándares abiertos que maximizan su libertad y flexibilidad. La sección destaca el Protocolo de contexto modelo (MCP) para la integración de herramientas y analiza las herramientas específicas del marco y las metaherramientas especializadas que mejoran los flujos de trabajo de los agentes.

En esta sección:

Categorías de herramientas

La creación de sistemas de agentes implica tres categorías principales de herramientas.

Herramientas basadas en protocolos

Las herramientas basadas en protocolos utilizan protocolos estandarizados para la comunicación: agent-to-tool

- Herramientas MCP: herramientas estándar abiertas que funcionan en varios marcos con opciones de ejecución local y remota.
- OpenAIllamada a funciones: herramientas patentadas que son específicas de los OpenAI modelos.
- Anthropicherramientas: una variante de la OpenAI función que requiere herramientas patentadas que son específicas de los modelos de Anthropic Claude.

Herramientas nativas de Framework

Las herramientas nativas de Framework se integran directamente en marcos de agentes específicos:

- Strands Agents herramientas: herramientas ligeras quick-to-implement y específicas del marco. Strands Agents
- LangChainherramientas: herramientas Python basadas en herramientas que están estrechamente integradas con el LangChain ecosistema.
- LlamaIndexherramientas: herramientas que están optimizadas para la recuperación y el procesamiento internos LlamaIndex de datos.

Metaherramientas

Las metaherramientas mejoran los flujos de trabajo de los agentes sin tomar acciones externas directas:

- Herramientas de flujo de trabajo: administre el flujo de ejecución de los agentes, la lógica de ramificación y la administración del estado.
- Herramientas gráficas de agentes: coordine varios agentes en flujos de trabajo complejos.
- Herramientas de memoria: proporcionan almacenamiento y recuperación persistentes de la información en todas las sesiones de los agentes.
- Herramientas de reflexión: permiten a los agentes analizar y mejorar su propio rendimiento.

Herramientas basadas en protocolos

Al considerar las herramientas basadas en protocolos, el Model Context Protocol (MCP) proporciona la base más completa y flexible para la integración de herramientas. Como se indica en la entrada del blog de código AWS abierto sobre la interoperabilidad de los agentes, AWS ha adoptado el MCP como un protocolo estratégico y ha contribuido activamente a su desarrollo.

En la siguiente tabla se describen las opciones para el despliegue de la herramienta MCP.

Modelo de despliegue	Descripción	Ideal para	Implementación

Basado en un estudio local	Las herramientas se ejecutan en el mismo proceso que el agente	Desarrollo, pruebas y herramientas sencillas	Rápida de implement ar sin sobrecarga de red
Basado en eventos enviados por el servidor local (SSE)	Las herramientas se ejecutan localmente pero se comunican a través de HTTP	Herramientas locales más complejas con separación de preocupaciones	Mejor aislamiento pero baja latencia
HTTP remoto transmisible	Las herramientas se ejecutan en servidore s remotos	Entornos de producción y herramientas compartidas	Escalable y gestionad o de forma centraliz ada

Los MCP oficiales SDKs están disponibles para crear herramientas de MCP:

- PythonSDK: implementación integral con soporte completo de protocolos
- TypeScriptSDK:JavaScript/TypeScriptimplementación para aplicaciones web
- JavaSDK: implementación de Java para aplicaciones empresariales

Estos SDKs proporcionan los componentes básicos para crear herramientas compatibles con MCP en su idioma preferido, con implementaciones coherentes de la especificación del protocolo.

Además, AWS ha implementado el MCP en el SDK. Strands Agents El Strands Agents SDK proporciona una forma sencilla de crear y utilizar herramientas compatibles con el MCP. La documentación completa está disponible en el Strands Agents GitHub repositorio. Para casos de uso más sencillos o cuando se trabaja fuera del Strands Agents marco, el MCP oficial SDKs ofrece implementaciones directas del protocolo en varios idiomas.

Características de seguridad de las herramientas MCP

Las características de seguridad de las herramientas MCP incluyen las siguientes:

- OAuth Autenticación 2.0/2.1: autenticación estándar del sector
- Alcance de los permisos: control de acceso detallado para las herramientas

- Descubrimiento de la capacidad de la herramienta: descubrimiento dinámico de las herramientas disponibles
- Gestión estructurada de errores: patrones de error consistentes

Cómo empezar con las herramientas de MCP

Para implementar el MCP para la integración de herramientas, lleve a cabo las siguientes acciones:

1. Explore el Strands AgentsSDK para obtener una implementación de MCP lista para la producción.
2. Revise la documentación técnica del MCP para comprender los conceptos básicos.
3. Utilice los ejemplos prácticos descritos en esta entrada de blog de código AWS abierto.
4. Comience con herramientas locales sencillas antes de pasar a herramientas remotas.
5. Únase a la comunidad de MCP para influir en la evolución del protocolo.

Explore Gateway AgentCore

Amazon Bedrock AgentCore Gateway proporciona a los desarrolladores una forma fácil y segura de crear, implementar, descubrir y conectarse a las herramientas de MCP y otros puntos de enlace de destino a escala. Con AgentCore Gateway, los desarrolladores pueden convertir APIs AWS Lambda las funciones y los servicios existentes en herramientas compatibles con el MCP. Luego, con solo unas pocas líneas de código, pueden poner estas herramientas a disposición de los agentes a través de los puntos finales de AgentCore Gateway. AgentCore Gateway admite OpenAPI Lambda como tipos de entrada y es la única solución que proporciona autenticación integral de entrada y autenticación de salida en un servicio totalmente gestionado. Smithy

Herramientas nativas de Framework

Si bien el Model Context Protocol (MCP) proporciona la base más flexible, las herramientas nativas del framework ofrecen ventajas para casos de uso específicos.

El Strands AgentsSDK ofrece herramientas Python basadas en herramientas que se caracterizan por su diseño liviano que requiere una sobrecarga mínima para operaciones sencillas. Permiten una implementación rápida y permiten a los desarrolladores crear herramientas con solo unas pocas líneas de código. Además, están estrechamente integrados para funcionar sin problemas dentro del Strands Agents marco.

El siguiente ejemplo demuestra cómo crear una herramienta meteorológica sencilla utilizandoStrands Agents. Los desarrolladores pueden transformar rápidamente Python las funciones en herramientas accesibles a los agentes con una sobrecarga de código mínima y generar automáticamente la documentación adecuada a partir de la cadena de documentos de la función.

```
#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 la creación rápida de prototipos o para casos de uso sencillos, las herramientas nativas del framework pueden acelerar el desarrollo. Sin embargo, para los sistemas de producción, las herramientas MCP ofrecen una mejor interoperabilidad y flexibilidad en el futuro que las herramientas nativas del marco.

La siguiente tabla proporciona una descripción general de otras herramientas específicas del marco.

Plataforma	Tipo de herramienta	Ventajas	Consideraciones
AutoGen	Definiciones de funciones	Sólido soporte multiagente	Microsoftecosistema
LangChain	Pythonclases	Amplio ecosistema de herramientas prediseñadas	Bloqueo de un marco
LlamaIndex	Funciones de Python	Optimizado para operaciones de datos	Limitado a LlamaIndex

Meta-herramientas

Las metaherramientas no interactúan directamente con sistemas externos. En cambio, mejoran las capacidades de los agentes mediante la implementación de patrones de los agentes. En esta sección se analiza el flujo de trabajo, el gráfico de agentes y las metaherramientas de memoria.

Metaherramientas de flujo de trabajo

Las metaherramientas de flujo de trabajo gestionan el flujo de ejecución de los agentes:

- Gestión del estado: mantenga el contexto en las interacciones entre varios agentes
- Lógica de ramificación: habilite las rutas de ejecución condicionales
- Mecanismos de reintento: gestione los errores con estrategias de reintento sofisticadas

Entre los ejemplos de marcos con metaherramientas de flujo de trabajo se incluyen LangGraphlas capacidades de flujo de trabajo. Strands Agents

Metaherramientas Agent Graph

Las metaherramientas Agent Graph coordinan el trabajo conjunto de varios agentes:

- Delegación de tareas: asigne subtareas a agentes especializados
- Agregación de resultados: combine los resultados de varios agentes
- Resolución de conflictos: resuelva los desacuerdos entre los agentes

Los marcos CrewAIse especializan en la coordinación de gráficos de agentes AutoGeny se especializan en ella.

Metaherramientas de memoria

Las metaherramientas de memoria proporcionan almacenamiento y recuperación persistentes:

- Historial de conversaciones: mantenga el contexto en todas las sesiones
- Bases de conocimiento: almacene y recupere información específica del dominio
- Almacenes vectoriales: habilitan las capacidades de búsqueda semántica

El sistema de recursos de MCP proporciona una forma estandarizada de implementar metaherramientas de memoria que funcionan en diferentes marcos de agentes.

Estrategia de integración de herramientas

La estrategia de integración de herramientas que elijas repercute directamente en lo que tus agentes pueden conseguir y en la facilidad con la que puede evolucionar tu sistema. Priorice los protocolos abiertos, como el Model Context Protocol (MCP), y utilice estratégicamente las metaherramientas y las herramientas nativas del marco. De esta forma, podrá crear un ecosistema de herramientas que siga siendo flexible y potente a medida que avance la tecnología de IA.

El siguiente enfoque estratégico para la integración de herramientas maximiza la flexibilidad y, al mismo tiempo, satisface las necesidades inmediatas de su organización:

1. Adopte el MCP como base: el MCP proporciona una forma estandarizada de conectar a los agentes con herramientas con sólidas funciones de seguridad. Comience con el MCP como su protocolo de herramientas principal para:
   - Herramientas estratégicas que se utilizarán en múltiples implementaciones de agentes.
   - Herramientas sensibles a la seguridad que requieren una autenticación y una autorización sólidas.
   - Herramientas que necesitan ejecución remota en entornos de producción.
2. Utilice herramientas nativas del marco cuando sea apropiado: considere usar herramientas nativas del marco para:
   - Creación rápida de prototipos durante el desarrollo inicial.
   - Herramientas sencillas y no esenciales con requisitos de seguridad mínimos.
   - Funcionalidad específica del marco que aprovecha capacidades únicas.
3. Implemente metaherramientas para flujos de trabajo complejos: añada metaherramientas para mejorar la arquitectura de sus agentes:
   - Comience de forma sencilla con patrones de flujo de trabajo básicos.
   - Añada complejidad a medida que vayan madurando sus casos de uso.
   - Estandarice las interfaces entre los agentes y las metaherramientas.
4. Planifique para la evolución: construya pensando en la flexibilidad del futuro:
   - Documente las interfaces de las herramientas independientemente de las implementaciones.
   - Cree capas de abstracción entre los agentes y las herramientas.

- Establezca rutas de migración de protocolos propietarios a protocolos abiertos.

Mejores prácticas de seguridad para la integración de herramientas

La integración de herramientas afecta directamente a su postura de seguridad. En esta sección se describen las prácticas recomendadas que debe tener en cuenta para su organización.

Autenticación y autorización

Utilice los siguientes controles de acceso robustos:

- Utilice la OAuth versión 2.0/2.1: implemente la autenticación estándar del sector para las herramientas remotas.
- Implemente los privilegios mínimos: conceda a las herramientas solo los permisos que necesitan.
- Cambie las credenciales: actualice periódicamente las claves de API y los tokens de acceso.

Protección de datos

Para ayudar a proteger los datos, adopta las siguientes medidas:

- Valide las entradas y salidas: implemente la validación del esquema para todas las interacciones entre herramientas.
- Cifre los datos confidenciales: utilice TLS para todas las comunicaciones remotas con las herramientas.
- Implemente la minimización de datos: transfiera solo la información necesaria a las herramientas.

Monitoreo y auditoría

Mantenga la visibilidad y el control mediante el uso de estos mecanismos:

- Registre todas las invocaciones de herramientas: mantenga registros de auditoría exhaustivos.
- Supervise las anomalías: detecte patrones de uso inusuales de las herramientas.
- Implemente la limitación de velocidad: evite el abuso mediante el uso excesivo de herramientas.

El modelo de seguridad del Model Context Protocol (MCP) aborda estas preocupaciones de manera integral. Para obtener más información, consulte Consideraciones de seguridad en la documentación del MCP.

Conclusión

El panorama de la IA de los agentes sigue evolucionando rápidamente y ofrece a las organizaciones nuevas y poderosas formas de crear sistemas inteligentes y autónomos. Esta guía ha explorado tres componentes esenciales para una implementación exitosa: los marcos que proporcionan la base, las plataformas que proporcionan el entorno, los protocolos que permiten la comunicación y las herramientas que amplían las capacidades.

A medida que los marcos vayan madurando, cabe esperar una mayor interoperabilidad, una estandarización en torno a protocolos como el Model Context Protocol (MCP) y capacidades de orquestación más sofisticadas para los agentes autónomos. Las organizaciones que adquieran experiencia en estos marcos en la actualidad estarán bien posicionadas para crear agentes cada vez más autónomos e inteligentes que ofrezcan un valor empresarial significativo.

Las plataformas proporcionan el entorno de ejecución, gobierno y ciclo de vida en el que funcionan los sistemas de las agencias. Se ocupan de cuestiones como la identidad, los límites de seguridad, la observabilidad, la administración de la memoria, la conexión a las sesiones y la interacción segura con las herramientas y los datos. En AWS los entornos, plataformas como los tiempos de ejecución de los agentes gestionados y los servicios de orquestación permiten a las organizaciones implementar, supervisar, desarrollar y controlar los agentes autónomos y los sistemas de agentes a escala. Las plataformas unen los marcos fundamentales con los requisitos operativos del mundo real.

La elección de los protocolos de los agentes representa una decisión estratégica que equilibra las necesidades de desarrollo inmediatas con la flexibilidad y la interoperabilidad a largo plazo. Al dar prioridad a los protocolos abiertos y crear las capas de abstracción adecuadas, las organizaciones pueden crear sistemas de agentes que se adapten a las tecnologías en evolución y, al mismo tiempo, cumplan con los requisitos empresariales actuales.

Para la mayoría de las organizaciones, el MCP representa una base sólida debido a su estándar abierto, su creciente ecosistema, su compatibilidad con los patrones de agent-to-agent comunicación y sus capacidades de integración de herramientas. AWS ha adoptado el MCP y el Agent2Agent (A2A) como protocolos estratégicos, contribuyendo activamente a su desarrollo e implementándolos en servicios como el SDK. Strands Agents Al utilizar MCP o A2A junto con las herramientas y metaherramientas nativas del marco adecuadas, puede crear sistemas de agentes que ofrezcan un valor inmediato y, al mismo tiempo, se adapten a las innovaciones futuras.

Recursos

Utilice los siguientes AWS y otros recursos relacionados con el desarrollo de agentes autónomos.

AWS Blogs

- Amazon Bedrock AgentCore Memory: creación de agentes sensibles al contexto
- Prácticas recomendadas para crear aplicaciones sólidas de IA generativa con agentes de Amazon Bedrock (Parte 1)
- Prácticas recomendadas para crear aplicaciones sólidas de IA generativa con agentes de Amazon Bedrock (Parte 2)
- Cree potentes canalizaciones de RAG con LlamaIndex Amazon Bedrock
- Cree agentes de IA confiables con Amazon Bedrock Observability AgentCore
- Evalúe las respuestas de RAG con Amazon Bedrock y RAGAS LlamaIndex
- Presentamos el intérprete de AgentCore código Amazon Bedrock
- Presentamos Amazon Bedrock AgentCore Gateway: Transformando el desarrollo de herramientas de agentes de IA empresarial
- Presentamos Amazon Bedrock AgentCore Identity: protección de la IA de los agentes a gran escala
- Presentamos un Strands Agents SDK de código abierto para agentes de IA
- Protocolos abiertos para la interoperabilidad de los agentes, parte 1: Comunicación entre agentes en el MCP
- Inicie y escale de forma segura sus agentes y herramientas en Amazon Bedrock Runtime AgentCore
- AWS Transform para .NET, el primer servicio de inteligencia artificial para modernizar las aplicaciones.NET a gran escala
- AWS Resumen semanal: Strands Agents

AWS Guía prescriptiva

- Operacionalización de la IA de los agentes en AWS
- Fundamentos de la IA agencial en AWS

- Los patrones y flujos de trabajo de la IA de los agentes están activos AWS
- Creación de arquitecturas sin servidor para la IA de los agentes en AWS
- Creación de arquitecturas multiusuario para la IA de los agentes en AWS
- Seguridad para la IA de los agentes en AWS
- Recupere las opciones y arquitecturas de generación aumentada activadas AWS

AWS recursos

- Documentación de Amazon Bedrock
- Documentación de Amazon Bedrock AgentCore
- Amazon Bedrock AgentCore Starter Toolkit (repositorio) GitHub
- Documentación de Amazon Nova
- AWS Servidores MCP (GitHubrepositorio)

Otros recursos de

- AutoGendocumentación () Microsoft
- Creación de agentes eficaces (Anthropic)
- CrewAI GitHubrepositorio
- Documentación de LangChain
- LangGraphplataforma
- Documentación de LlamaIndex
- Documentación sobre el protocolo Model Context
- Documentación de Strands Agents
- Strands AgentsDescripción general de las herramientas
- Strands AgentsGuía de inicio rápido

Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las notificaciones RSS.

Cambio	Descripción	Fecha
Sección nueva	Se agregó la sección de plataformas	16 de enero de 2026
Publicación inicial	—	14 de julio de 2025

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.