Los patrones y flujos de trabajo de la IA de los agentes están en AWS

AWS Guía prescriptiva

AWS Guía prescriptiva: Los patrones y flujos de trabajo de la IA de los agentes están en AWS

Table of Contents

Flujos de trabajo y patrones de IA de agentes en AWS

Aaron Sempf y Andrew Hooker, Amazon Web Services

Julio de 2025 (historial del documento)

Las organizaciones están adoptando grandes modelos de lenguaje (LLMs) y agentes de software para resolver problemas dinámicos y multidominio mediante una nueva disciplina arquitectónica llamada patrones agentic. Los patrones de los agentes son modelos fundamentales y estructuras modulares que se utilizan para diseñar y organizar agentes de IA orientados a objetivos en muchos contextos.

Destinatarios previstos

Esta guía está dirigida a arquitectos, desarrolladores y líderes de productos que desean crear aplicaciones inteligentes que vayan más allá de la lógica estática, la lógica simbólica y la automatización determinista.

Objetivos

Esta guía proporciona un marco de diseño y un enfoque de implementación para los sistemas de agentes de IA que funcionan de forma autónoma sin dejar de ser controlables y alineados con sus objetivos. Conecta los patrones arquitectónicos basados en eventos con diversas alternativas de agentes y demuestra cómo crear sistemas de agentes aptos para producción utilizando arquitecturas nativas de la nube. En esta guía se tratan los siguientes temas:

- Patrones de agentes: los patrones de agentes son plantillas de diseño reutilizables que describen la estructura y el comportamiento de los agentes individuales. Esto incluye agentes de razonamiento, agentes de recuperación aumentada, agentes de codificación, interfaces de voz, orquestadores de flujos de trabajo y sistemas colaborativos de múltiples agentes. Cada patrón ilustra cómo los agentes perciben, razonan, actúan y aprenden, mapeados. Servicios de AWS
- Flujos de trabajo de LLM: los flujos de trabajo se centran en la forma en que los agentes utilizan el LLMs razonamiento. Exploran las estrategias de incitación y los mecanismos de planificación, y describen cómo se LLMs utilizan no solo para generar texto, sino también para impulsar comportamientos estructurados, interpretables y confiables dentro de un circuito de agentes.

- Patrones de flujo de trabajo de los agentes: los patrones de flujo de trabajo describen cómo interactúan varios agentes, herramientas y entornos para formar sistemas autónomos. Esto incluye patrones para la orquestación de tareas, la delegación de subagentes, la coordinación basada en eventos, la observabilidad y el control. Estos aspectos promueven arquitecturas de IA escalables, componibles y auditables.

Acerca de esta serie de contenido

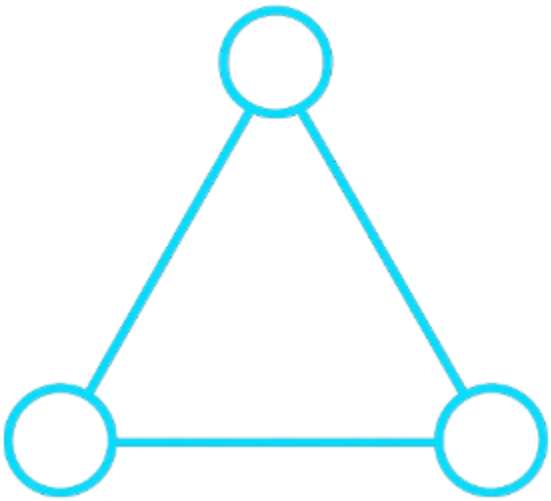
Esta guía forma parte de una serie sobre la IA de los agentes en. AWS Para obtener más información y ver las demás guías de esta serie, consulte Agentic AI en el sitio web de orientación prescriptiva. AWS

Patrones de agentes

Los patrones de agentes son componentes básicos reutilizables y componibles que se pueden adaptar a dominios, casos de uso y niveles de complejidad específicos. Sin embargo, los sistemas de agentes difieren de las aplicaciones tradicionales. En el centro de todos los diseños de agentes de IA se encuentra un modelo conceptual basado en los tres principios fundamentales siguientes:

- Asincrónico: los agentes operan en entornos poco acoplados y repletos de eventos
- Autonomía: los agentes actúan de forma independiente, sin control humano o externo
- Agencia: los agentes actúan con un propósito, en nombre de un usuario o sistema, para alcanzar objetivos específicos

El triángulo del siguiente diagrama representa los componentes básicos de un agente de software: percepción, razón y acción. Esto permite que un sistema de agencia observe, tome decisiones y actúe dentro de su entorno.



Por diseño, los patrones de agente proporcionan un lenguaje de diseño modular para crear sistemas de IA, lo que significa que son accesibles, operativos, ampliables y están listos para la producción. El diseño de estos sistemas requiere prestar especial atención a las siguientes tres dimensiones interrelacionadas, que se analizarán con más detalle más adelante en esta guía.

En esta sección

- Tool-based agentes para servidores
- Computer-use agentes
- Agentes de codificación
- Agentes de voz y voz
- Agentes de orquestación del flujo de trabajo
- Memory-augmented agentes
- Agentes de simulación y de banco de pruebas
- Agentes de observación y monitoreo
- Multi-agent colaboración

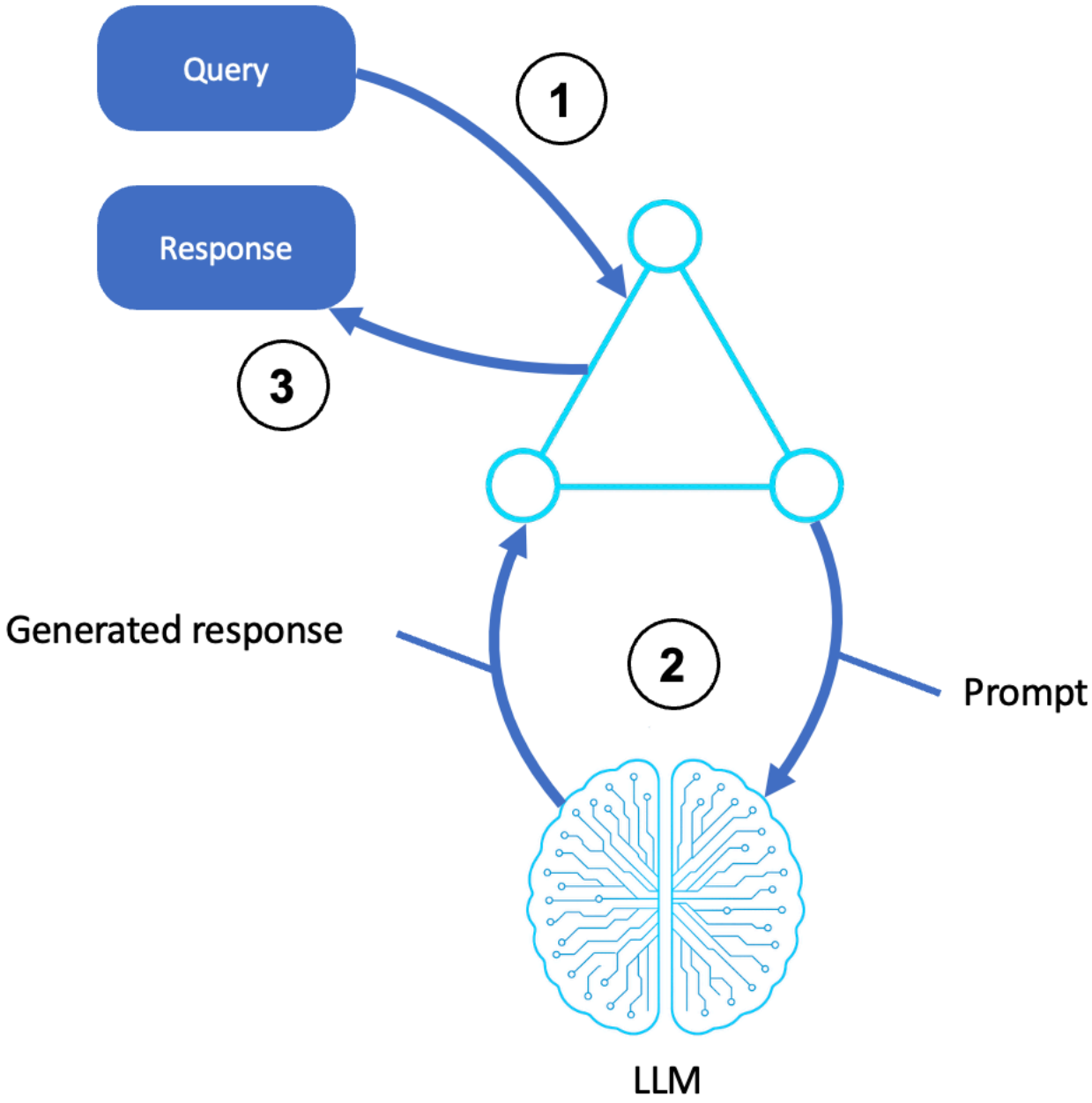
Agentes de razonamiento básicos

Un agente de razonamiento básico es la forma más simple de inteligencia artificial que realiza inferencias lógicas o toma de decisiones en respuesta a una consulta. Acepta la información de un usuario o un sistema y procesa las consultas y genera respuestas mediante indicaciones estructuradas.

Este patrón es útil para las tareas que requieren un razonamiento, una clasificación o un resumen en un solo paso en función de un contexto determinado. No usa memoria, herramientas ni administración de estados, lo que lo hace sin estado, liviano y altamente componible en grandes flujos de trabajo.

Arquitectura

El flujo de un agente de razonamiento básico se muestra en el siguiente diagrama:



Description (Descripción)

1. Recibe una entrada
   - Un usuario, un sistema o un agente principal envía una consulta o instrucción.
   - La entrada se transfiere al shell del agente o a la capa de orquestación.
   - Este paso incluye el preprocesamiento, la creación rápida de plantillas y la identificación de los objetivos.

2. Invoca el LLM
    - El agente transforma la consulta en una solicitud estructurada y la envía a un LLM (por ejemplo, a través de Amazon Bedrock).
    - El LLM genera una respuesta basada en el mensaje utilizando el conocimiento y el contexto previamente entrenados.
    - El resultado generado puede incluir pasos de razonamiento (cadena de pensamiento), respuestas finales u opciones clasificadas.
3. Devuelve una respuesta
    - El resultado generado se retransmite a la interfaz del agente.
    - Esto puede incluir el formateo, el posprocesamiento o una respuesta de la API.

Capacidades

- Admite lenguaje natural o entrada estructurada
- Utiliza una ingeniería rápida para guiar el comportamiento
- Sin estado y escalable
- Se puede integrar en la interfaz de usuario, la CLI, las API y las canalizaciones

Limitaciones

- Sin memoria ni conocimiento histórico
- Sin interacción con herramientas o fuentes de datos externas
- Limitado a lo que el LLM sabe en el momento de la inferencia

Casos de uso comunes

- Preguntas y respuestas conversacionales
- Explicaciones y resúmenes de las políticas
- Guía para la toma de decisiones
- Flujos de chatbots ligeros y automatizados
- Clasificación, etiquetado y puntuación

Guía para la implementación

Puede utilizar las siguientes herramientas y servicios para crear un agente de razonamiento básico:

- Amazon Bedrock para la invocación de LLM (Anthropic, AI21, Meta)
- Amazon API Gateway o AWS Lambda para exponerlo como un microservicio sin estado
- Plantillas de solicitudes almacenadas en el almacén de parámetros o como código AWS Secrets Manager

Resumen

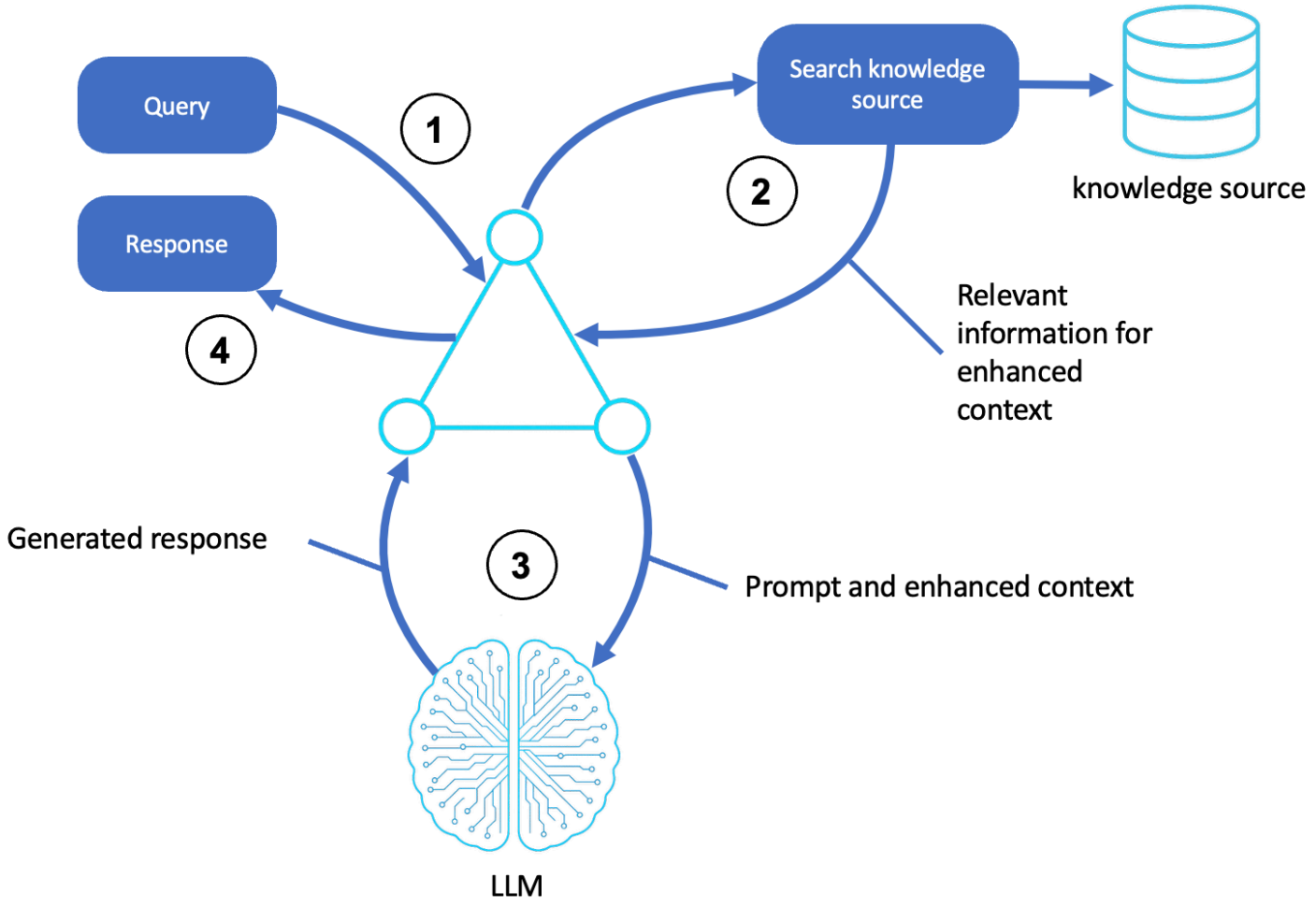
El agente de razonamiento básico es fundamental debido a su estructura simple. Tiene capacidades básicas que convierten los objetivos en vías de razonamiento que conducen a resultados inteligentes. Este patrón suele ser un punto de partida para patrones avanzados, como los agentes basados en herramientas y los agentes que utilizan la generación aumentada por recuperación (RAG). También es un componente fiable y modular de grandes flujos de trabajo.

Agente RAG

Retrieval-augmented La generación (RAG) es una técnica que combina la recuperación de información con la generación de texto para crear respuestas precisas y contextuales. El RAG permite a los agentes recuperar información externa relevante antes de contratar el LLM. Amplía la memoria efectiva y la precisión del razonamiento de un agente al basar sus decisiones en información actualizada, fáctica o específica de un dominio. A diferencia de los LLM apátridas, que se basan únicamente en pesas previamente entrenadas, RAG tiene una capa de búsqueda de conocimientos externa que mejora de forma dinámica las solicitudes en función del contexto.

Arquitectura

La lógica del patrón RAG se ilustra en el siguiente diagrama:



Description (Descripción)

1. Recibe una consulta
   - Un usuario o un sistema anterior envía una consulta o un objetivo al agente.
   - El intérprete de comandos acepta la solicitud y la formatea como una solicitud de razonamiento.
2. Busca una fuente externa
   - El agente identifica los conceptos y la intención a partir de la consulta.
   - Consulta una fuente de conocimiento, como un almacén vectorial, una base de datos o un índice de documentos, mediante la búsqueda semántica o la coincidencia de palabras clave.
   - Los pasajes, documentos o entidades más relevantes se recuperan para usarlos en el siguiente paso.
3. Genera una respuesta contextual

- El agente amplía el mensaje con la información recuperada, lo que constituye una entrada mejorada por el contexto para el LLM.
- El LLM procesa cualquier entrada mediante el razonamiento generativo (por ejemplo, una cadena de pensamiento o una reflexión) para producir una respuesta precisa.

4. Devuelve el resultado final
   - El agente prepara el resultado envolviéndolo en cualquier encabezado de comunicación o en el formato requerido y, a continuación, lo devuelve al usuario o al sistema de llamada.
   - (Opcional) Los documentos recuperados y la salida de LLM se pueden registrar, calificar y almacenar en la memoria para futuras consultas.

Capacidades

- Fact-grounded generan resultados incluso en dominios de larga duración o específicos de la empresa
- Ampliación de la memoria sin ajustar el modelo
- Contexto dinámico basado en cada consulta y estado de usuario
- Totalmente compatible con bases de datos vectoriales, índices semánticos y filtrado de metadatos

Casos de uso comunes

- Asistentes de conocimiento empresarial
- Bots de cumplimiento normativo
- Copilotos de atención al cliente
- Search-enhanced chatbots
- Agentes de documentación para desarrolladores

Guía para la implementación

Utilice las siguientes herramientas y servicios para crear un agente que utilice RAG:

- Amazon Bedrock para la invocación de LLM
- Amazon Kendra o Amazon Aurora para documentación o una búsqueda estructurada de datos OpenSearch

- Amazon Simple Storage Service (Amazon S3) para almacenamiento de documentos
- AWS Lambda para organizar la búsqueda, la solicitud y la inferencia de LLM
- Knowledge-based integraciones con agentes (mediante complementos de memoria, recuperadores semánticos o Amazon Bedrock)

Resumen

El agente RAG conecta el razonamiento de modelos estáticos con la inteligencia dinámica del mundo real. Proporciona a los agentes la capacidad de buscar lo que no saben, sintetizar las respuestas a partir del conocimiento recuperado y producir respuestas auditables y de alta confianza.

Los patrones RAG son la base para crear agentes inteligentes que amplíen el acceso al conocimiento sin necesidad de volver a capacitarse. Suele ser un precursor de patrones de orquestación más complejos que implican el uso de herramientas, la planificación y la memoria a largo plazo.

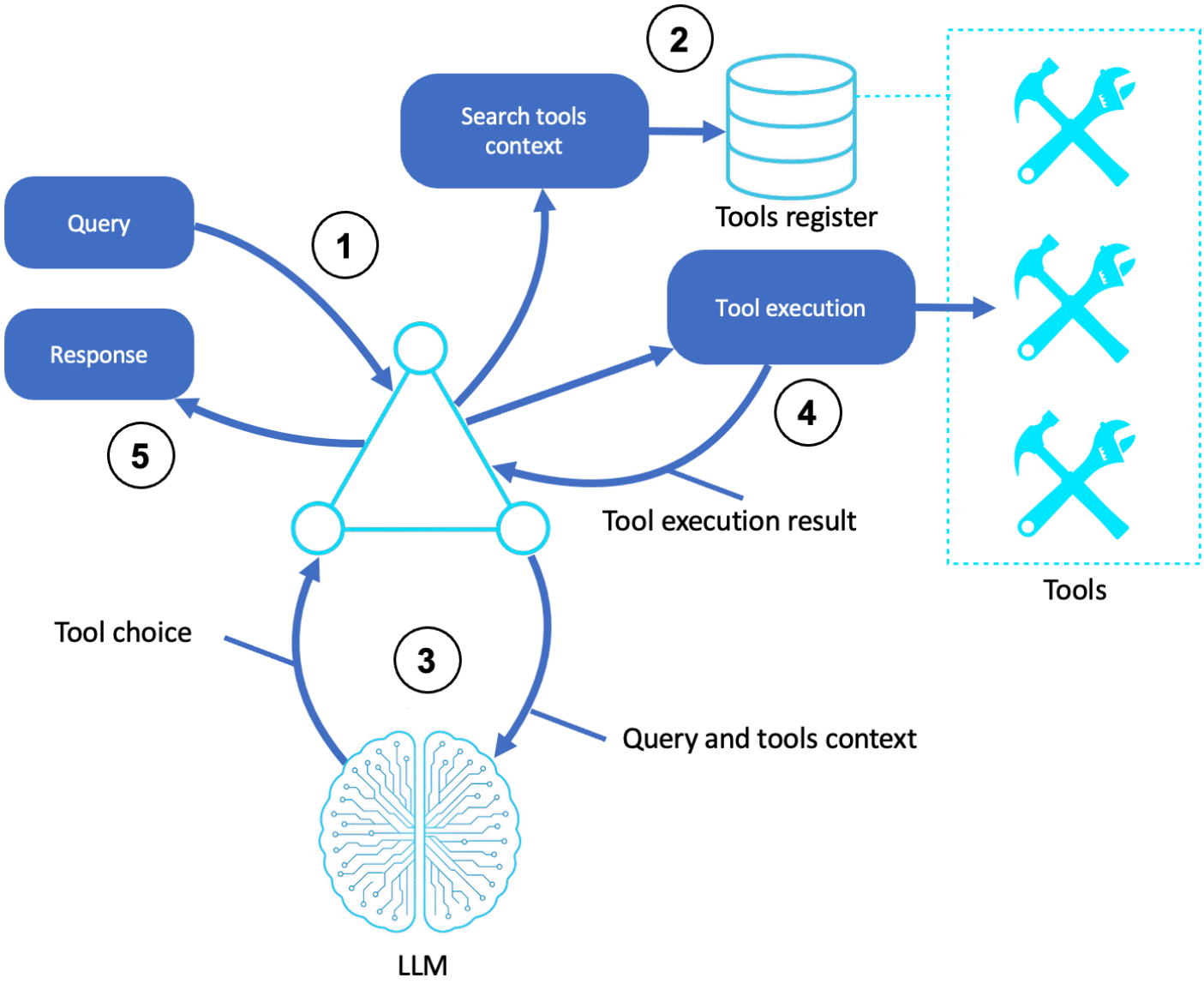
Tool-based agentes para llamar a funciones

Tool-based los agentes amplían las capacidades de los agentes de razonamiento al invocar funciones externas o API para completar tareas que van más allá del razonamiento basado únicamente en el lenguaje. Este patrón utiliza un LLM para decidir qué herramienta utilizar y, a continuación, genera argumentos de llamada e incorpora los resultados de la herramienta en su ciclo de razonamiento.

Este patrón permite a los agentes actuar en lugar de limitarse a dar respuestas. La interfaz de la herramienta representa cualquier capacidad que se pueda invocar, desde cálculos aritméticos y búsquedas en bases de datos hasta API externas y servicios en la nube.

Arquitectura

En el siguiente diagrama se muestra un agente basado en herramientas para llamar a funciones:



Description (Descripción)

1. Recibe una consulta
   - El agente recibe una consulta o tarea en lenguaje natural del usuario o del sistema que realiza la llamada.
2. Busca herramientas
   - El agente utiliza metadatos internos o un registro de herramientas para buscar las herramientas, los esquemas y las capacidades relevantes disponibles.
3. Selecciona e invoca las herramientas

- El LLM recibe los metadatos de la consulta y la herramienta (por ejemplo, los nombres de las funciones, los tipos de entrada y las descripciones) en su solicitud de datos.
- Elige la herramienta más relevante, construye los argumentos de entrada y devuelve una llamada a una función estructurada.

4. Ejecuta la herramienta elegida
   - El shell del agente o el ejecutor de herramientas ejecuta la función seleccionada y devuelve el resultado (por ejemplo, una salida de API, un valor de base de datos o un cálculo).
5. Devuelve una respuesta
   - El LLM pasa los resultados al agente, ya sea directamente o como parte de una solicitud actualizada. A continuación, devuelve un resultado en lenguaje natural.

Capacidades

- Selección dinámica de herramientas basada en el contexto de la tarea
- Schema-based solicitud (OpenAPI, esquema AWS JSON, interfaz de funciones)
- Interpretación de los resultados y encadenamiento de los resultados en el razonamiento
- Operaciones apátridas o con registro de sesiones

Casos de uso comunes

- Asistentes virtuales con acceso a datos externos
- Calculadoras y estimadoras financieras
- API-based trabajadores del conocimiento
- LLM que invocan, SageMaker puntos de conexión de AWS Lambda Amazon y servicios SaaS

Guía para la implementación

Utilice lo siguiente para crear agentes basados en herramientas para llamar a funciones:

- Amazon Bedrock con soporte para llamadas a funciones (Anthropic Claude)
- AWS Lambda como un backend de ejecución de herramientas
- Amazon API Gateway o AWS Step Functions para la organización de herramientas

- Amazon DynamoDB o Amazon Relational Database Service (Amazon RDS) para metadatos de herramientas sensibles al contexto
- Amazon EventBridge Pipelines o AWS Step Functions que mapean estados para enrutar las salidas

Resumen

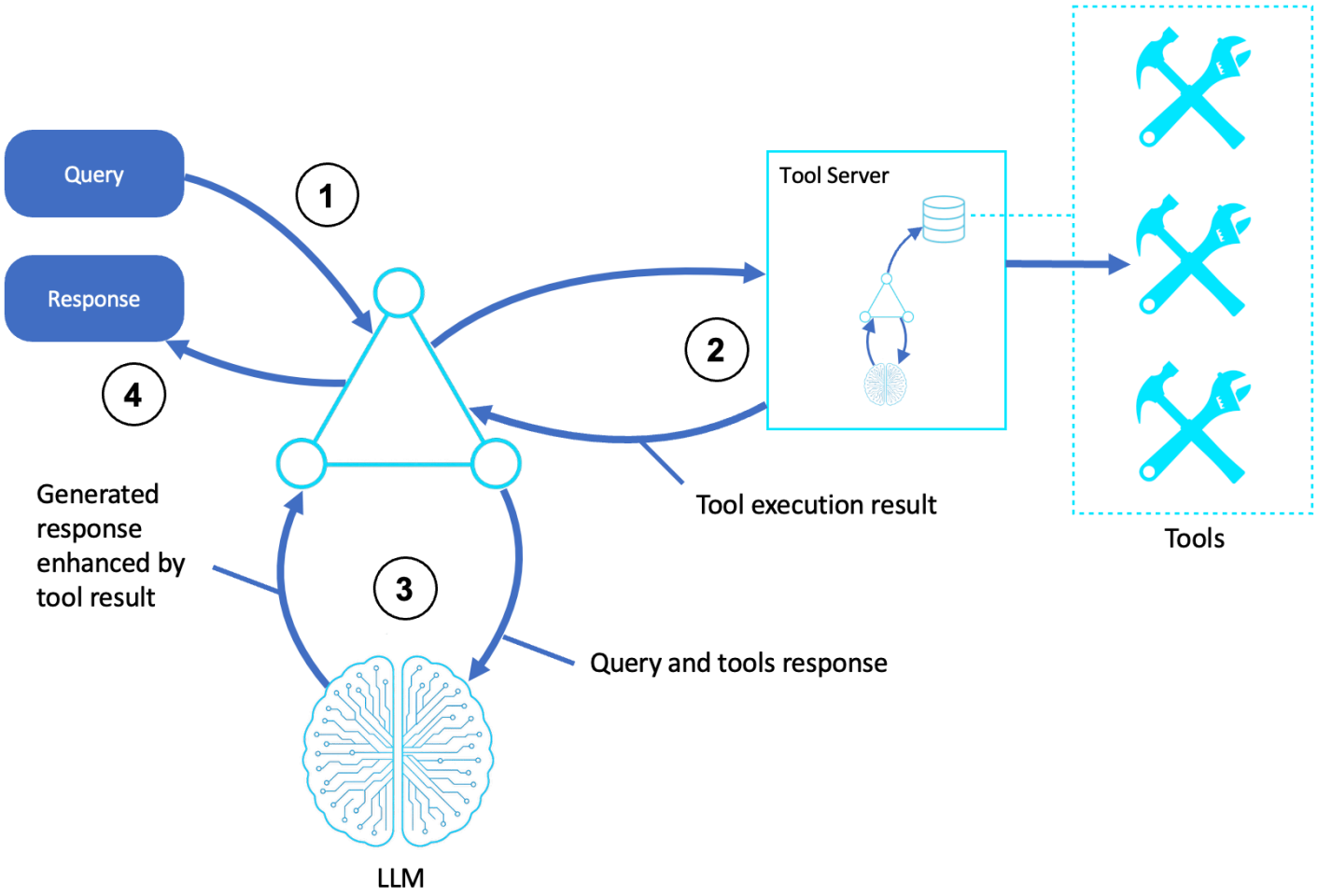
Tool-based los agentes que llaman a las funciones representan un cambio desde la comprensión del lenguaje hasta la ejecución de acciones. Estos agentes recurren a herramientas dinámicas y sensibles al contexto y, al mismo tiempo, mantienen el razonamiento LLM, transformando los asistentes pasivos en sistemas que completan tareas, acceden a los servicios e integran las operaciones comerciales. Este patrón es un componente importante de la IA de los agentes en los entornos empresariales, especialmente cuando se combina con esquemas declarativos, marcos de autorización y sistemas multiagente.

Tool-based agentes para servidores

Tool-based Los agentes para servidores mejoran las funciones de los agentes al delegar la ejecución de las herramientas a un servidor externo que tiene un entorno de ejecución dedicado a las herramientas, los scripts y los agentes compuestos. A diferencia de las llamadas a funciones en línea que el bucle de agentes selecciona e invoca, los agentes basados en servidor subcontratan la lógica y el proceso de ejecución a otros agentes o sistemas. Esto proporciona funciones avanzadas, como el encadenamiento de múltiples herramientas, la ejecución aislada y el razonamiento especializado. Los servidores de herramientas son ideales para acciones complejas, activas o que requieren muchos recursos, en las que las propias herramientas pueden implicar modelos de IA, reglas empresariales o entornos independientes.

Arquitectura

El siguiente es un patrón para los agentes basados en herramientas para servidores:



Description (Descripción)

1. Recibe una consulta
   - Un usuario o un sistema envía una solicitud al shell del agente.
   - El agente interpreta la consulta y se prepara para enviarla a un servidor de herramientas.
2. Ejecuta los procesos del servidor de herramientas
   - El agente envía la tarea, junto con los parámetros estructurados, a un servidor de herramientas.
   - El servidor de herramientas puede entonces:
     - Ejecute scripts o lógica en sistemas de cómputo dedicados (por ejemplo AWS Lambda, contenedores o Amazon SageMaker)
     - Utilice su propio subagente con el razonamiento LLM para seleccionar y ejecutar las herramientas
     - Gestione las dependencias, los reintentos o los flujos de ejecución en varios pasos

    - Envíe los resultados al agente principal cuando se complete la tarea

3. Utiliza el razonamiento LLM con el resultado de la herramienta
    - El agente invoca un LLM y pasa la consulta original y el resultado del servidor de herramientas como parte de la solicitud.
    - El LLM sintetiza una respuesta que incorpora la información recién adquirida.
4. Devuelve una respuesta
    - El agente devuelve una respuesta estructurada o en lenguaje natural al usuario o al sistema que realiza la llamada.
    - (Opcional) Los resultados se pueden almacenar en la memoria o en los registros de auditoría.

Capacidades

- Las herramientas se invocan fuera del ciclo de ejecución del agente principal
- La ejecución de la herramienta puede implicar llamadas LLM, cadenas lógicas o subagentes
- El agente actúa como controlador o despachador, no solo como un envoltorio de herramientas
- Permite la componibilidad, la escalabilidad y el aislamiento de la lógica

Casos de uso comunes

- Organización de cadenas de modelos (por ejemplo, mediante la combinación de LLM, visión y código)
- AI-driven canalizaciones de automatización
- DevOps agentes asistentes con ejecutores de guiones
- Agentes complejos de computación, simulación u optimización financiera
- Herramientas multimodales (por ejemplo, mediante la combinación de audio, documentación y acción)

Guía para la implementación

Puede crear este patrón utilizando lo siguiente: Servicios de AWS

- Amazon Bedrock (inferencia de agente, anfitrión y LLM)

- AWS Lambda, Amazon ECS o Amazon SageMaker Endpoints como entorno de ejecución del servidor de herramientas AWS Fargate
- Amazon API Gateway o AWS App Runner para exponer las API del servidor de herramientas
- Amazon EventBridge para la mensajería disociada del agente a la herramienta
- AWS Step Functions o AWS AppFabric para componer una lógica multiagente en el servidor de herramientas

Resumen

Tool-based los agentes que utilizan servidores son altamente modulares y escalables. Disocian la lógica de decisión de la ejecución, lo que permite que el agente principal sea ligero y, al mismo tiempo, delegue acciones complejas o delicadas a otros sistemas. Esto es importante para la IA de los agentes de nivel empresarial, especialmente en entornos que requieren gobernabilidad, observabilidad, aislamiento, composición dinámica o cualquier combinación de ambos.

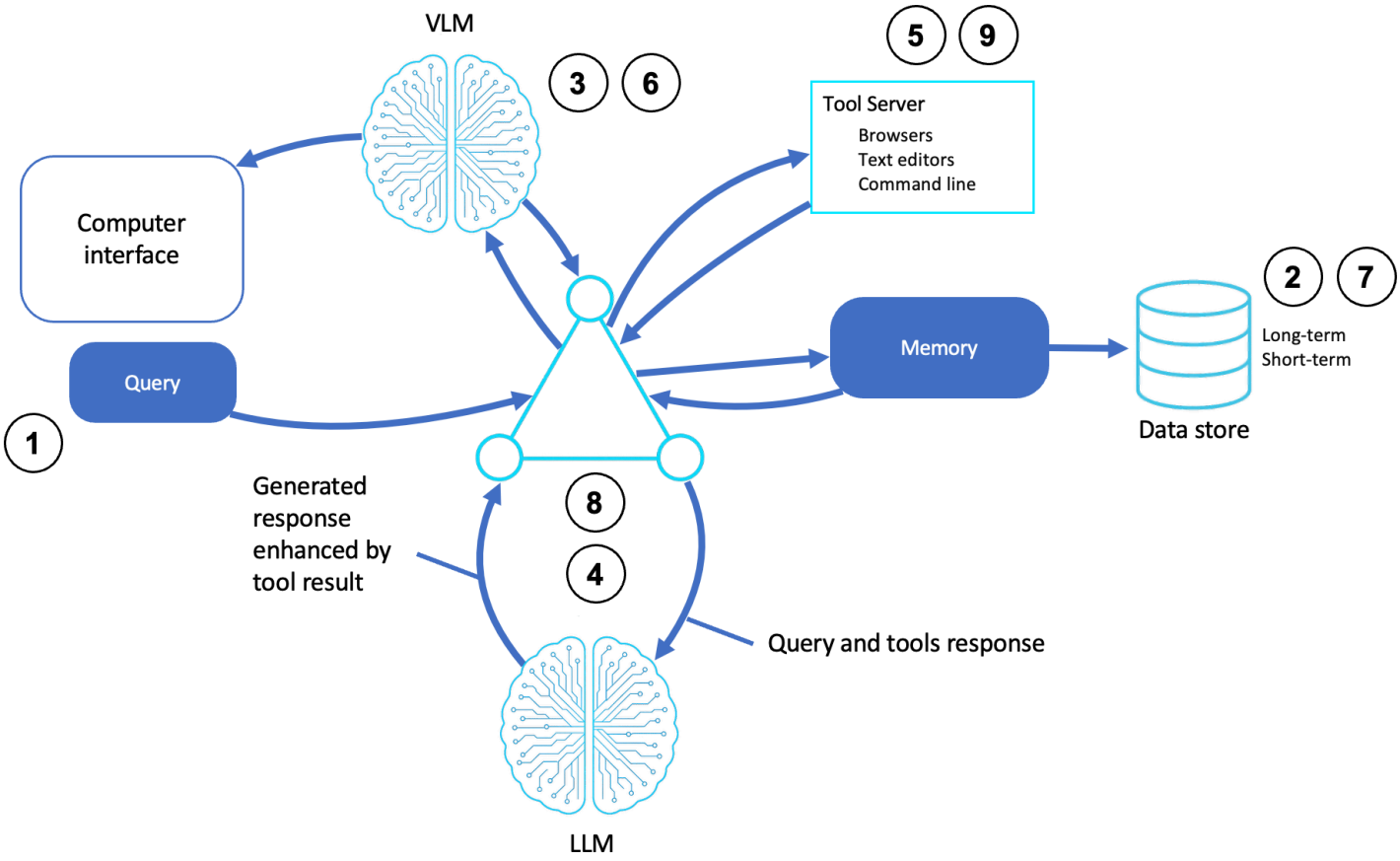
Computer-use agentes

Computer-use los agentes pueden simular o controlar entornos digitales como navegadores, terminales, sistemas de archivos y aplicaciones. Estos agentes interpretan la intención del usuario, interactúan con las interfaces visuales y de texto y realizan acciones orientadas a objetivos mediante la combinación del razonamiento LLM, los modelos de lenguaje visual (VLM) y los servidores de herramientas que ejecutan comandos o simulan eventos de entrada.

Este patrón es importante para las automatizaciones prácticas de IA, en las que el agente funciona no solo como asistente, sino también como un proxy que realiza acciones como lo haría un humano, a menudo utilizando las mismas herramientas y entornos.

Arquitectura

En el siguiente diagrama se muestra un patrón de agentes que utilizan ordenadores:



Description (Descripción)

1. Recibe una consulta
   - Una tarea o solicitud se proporciona a través de una interfaz de usuario, API o lenguaje natural.
2. Accede a la memoria
   - El agente recupera la memoria a corto y largo plazo para recordar órdenes, objetivos y estados del sistema anteriores.
3. Analiza el contexto visual
   - Un VLM observa la pantalla de la computadora, el estado del sistema o los elementos de la interfaz de usuario para comprender un contexto determinado e identificar los elementos procesables.
4. Razones a través de un LLM
   - El LLM combina la consulta, el estado de la memoria, la herramienta y la respuesta del servidor para determinar la siguiente acción.
5. Interactúa con el servidor de herramientas

- El agente invoca herramientas que están alojadas en un servidor, entre las que se incluyen las siguientes:
  - Navegadores (por ejemplo, Headless Chrome) y entornos de shell
  - Editores de texto y código
  - Interfaces de script personalizadas

6. Actualiza las entradas visuales
   - Si la interfaz de usuario del sistema cambia o es necesario realizar más observaciones, el VLM puede volver a analizar el estado de la pantalla o los búferes de texto.
7. Actualiza la memoria
   - Los nuevos conocimientos, los estados del sistema o los comentarios de los usuarios se graban en la memoria a corto y largo plazo.
8. Formula las decisiones y explicaciones finales
   - El LLM sintetiza los resultados o recomienda acciones en función de la consulta y el resultado de la herramienta.
9. Devuelve una respuesta
   - El agente devuelve los resultados a la interfaz (por ejemplo, una tarea completada, una confirmación o un contenido generado).

Capacidades

- Razonamiento multimodal con entradas visuales y textuales
- Control de las aplicaciones mediante entradas simuladas o API-driven
- Gestión de memoria para un estado persistente
- Autonomía en la ejecución secuencial (flujos de varios pasos)

Casos de uso comunes

- Desarrolladores de IA que escriben y ejecutan código en IDE
- Computer-use agentes para flujos de trabajo digitales repetitivos
- Usuarios simulados para pruebas de software y control de calidad
- Agentes de accesibilidad para navegar por las interfaces de usuario mediante instrucciones de voz o de alto nivel

- Automatización robótica inteligente de procesos (RPA) que se mejora con el razonamiento

Guía para la implementación

- Puede crear este patrón con lo siguiente: Servicios de AWS
- Amazon Bedrock para LLM-based planificar y razonar
- Amazon Elastic Compute Cloud (Amazon EC2) AWS Lambda o SageMaker Amazon notebooks para ejecutar servidores de herramientas con entornos de interfaz de usuario simulados
- Amazon Simple Storage Service (Amazon S3) o Amazon DynamoDB para la persistencia de la memoria
- Amazon Rekognition (o modelos personalizados) para el análisis de imágenes de interfaz de usuario en escenarios híbridos
- Amazon CloudWatch Logs o AWS X-Ray para registros de observabilidad y auditoría

Resumen

Computer-use los agentes actúan como operadores digitales autónomos, cerrando la brecha entre las interacciones y las acciones entre humanos y computadoras. AI-driven AI incorporar memoria, orquestación de herramientas y VLM, estos agentes pueden interactuar de forma adaptativa con sistemas diseñados para humanos, ejecutar acciones, actualizar archivos, navegar por los menús y generar respuestas.

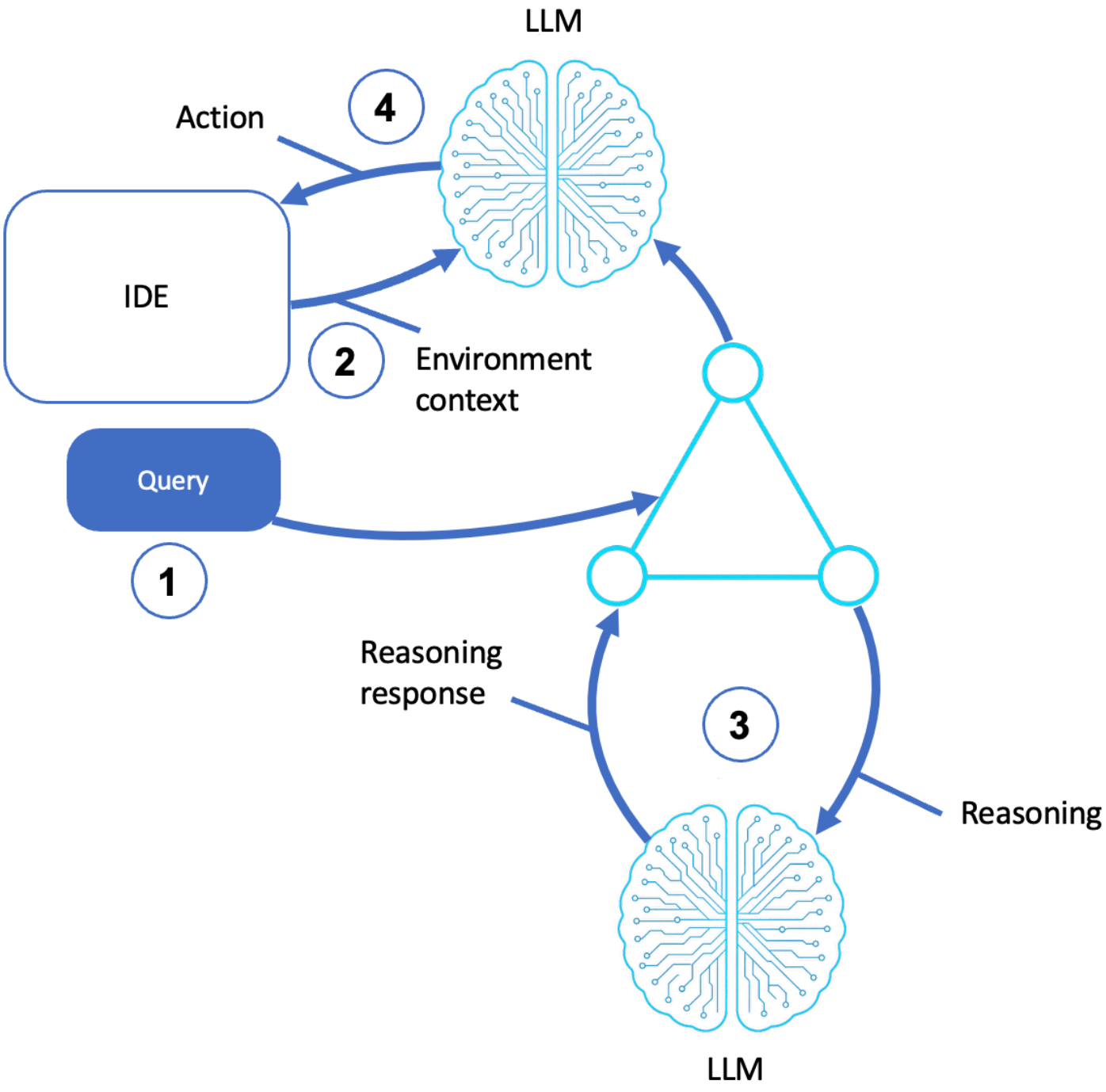
Agentes de codificación

Los agentes de codificación pueden razonar sobre las tareas de programación, generar o modificar código e interactuar con los entornos de los desarrolladores, como los IDE y las CLI. Estos agentes combinan la comprensión del lenguaje natural con el razonamiento estructurado para ayudar, aumentar y automatizar el desarrollo de software, desde la generación de funciones hasta la corrección de errores y la creación de pruebas.

A diferencia de las herramientas de autocompletado, los agentes de codificación interpretan activamente los objetivos de los usuarios, consultan el contexto del entorno de desarrollo (por ejemplo, abre archivos y rastrea los errores), identifican los requisitos y, a continuación, proponen y llevan a cabo acciones.

Arquitectura

En el siguiente diagrama se muestra un patrón de agente de codificación:



Description (Descripción)

1. Recibe la consulta

- El usuario proporciona instrucciones en lenguaje natural a través de una paleta de comandos, una ventana de chat o una CLI (por ejemplo, «Añadir registro a esta función» o «Refactorizar para facilitar la lectura»).

2. Extrae el contexto del entorno
   - El agente recopila el contexto del IDE, incluidos los archivos activos, la posición del cursor, los fragmentos de código y las tablas de símbolos.
   - Genera mensajes de error, resultados de pruebas y resultados de otros agentes.
3. Razonamiento LLM
   - El agente envía un mensaje, que incluye la consulta y el contexto ambiental, a un LLM.
     - El LLM realiza un pase de razonamiento para determinar lo siguiente:
     - ¿Qué debe cambiar
     - ¿Cómo generar una solución
     - ¿Algún paso de refactorización, reescritura o codificación
4. Ejecuta acciones
   - El LLM devuelve la salida al agente y la importa al IDE o al entorno de ejecución.
   - Esto puede incluir insertar o modificar código, generar comentarios o documentación y activar tareas posteriores de compilación, prueba y creación de listas.

Capacidades

- High-contextual conocimiento (por ejemplo, el estado del IDE, el cursor y el árbol de sintaxis)
- Razonamiento iterativo de objetivos y comentarios
- Planificación opcional del código y separación de acciones (por ejemplo, primero razonar y luego actuar)
- Funciona en flujos de trabajo de desarrolladores sincrónicos o asíncronos

Casos de uso comunes

- Generación de código a partir de descripciones de tareas
- Refactorización y optimización del código
- Test-case generación y validación
- Explicaciones de errores y depuración

- Asistentes de documentación
- Copilotos de programación emparejados

Guía para la implementación

- Puede crear este patrón con las siguientes herramientas y Servicios de AWS:
- Amazon Bedrock para la LLM-driven generación y el razonamiento
- Desarrollador de Amazon Q para sugerencias de codificación y finalizaciones
- AWS Lambda o Amazon Elastic Container Service (Amazon ECS) para ejecutar y probar entornos sandbox
- AWS Cloud9, extensiones de VS Code o integraciones IDE personalizadas para alojar y evaluar el contexto
- Amazon Simple Storage Service (Amazon S3) para almacenar mensajes intermedios, respuestas e historial de revisiones

Resumen

Los agentes de codificación son nuevas herramientas de AI-powered desarrollo capaces de interpretar el lenguaje natural, analizar el contexto, generar cambios de código de varios pasos e integrarse en el ciclo de vida del desarrollo del software.

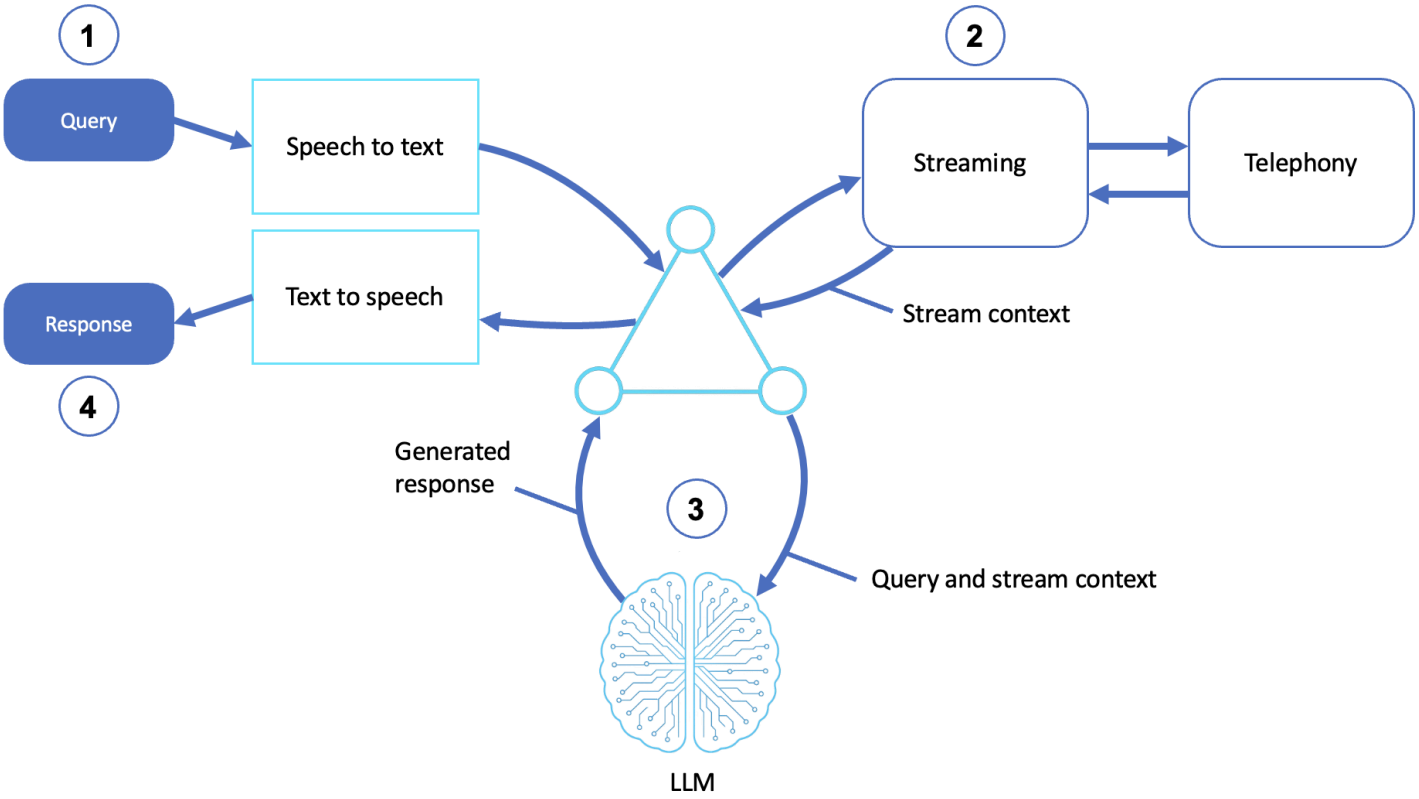
Agentes de voz y voz

Los agentes de voz y voz interactúan con los usuarios a través del diálogo oral. Estos agentes integran el reconocimiento de voz, la comprensión del lenguaje natural y la síntesis de voz para permitir la IA conversacional en plataformas de telefonía, móviles, web e integradas.

Los agentes de voz son particularmente eficaces en entornos de manos libres, en tiempo real o basados en la accesibilidad. Al combinar las interfaces de streaming con el LLM-powered razonamiento, facilitan interacciones ricas y dinámicas que resultan naturales para los usuarios.

Arquitectura

En el siguiente diagrama se muestra un agente de voz y voz:



Description (Descripción)

1. Recibe una consulta de voz
   - El usuario envía una solicitud a un teléfono, micrófono o sistema integrado.
   - Un módulo de voz a texto (STT) convierte el audio en texto.
2. Integra el contexto de streaming y telefonía
   - El agente utiliza una interfaz de streaming para gestionar I/O el audio en tiempo real.
   - Si se implementa en un contexto de contact center o de telecomunicaciones, la integración de telefonía gestiona el enrutamiento de sesiones, la entrada multifrecuencia (DTMF) de doble tono y el transporte multimedia.

Nota: El DTMF hace referencia a los tonos que se generan al pulsar los botones del teclado de un teléfono. En el contexto de la integración del contexto de la transmisión y la telefonía en los agentes de voz, el DTMF se utiliza como mecanismo de entrada de señales durante una llamada telefónica, especialmente en los sistemas de respuesta de voz interactiva (IVR). Las entradas DTMF permiten al agente:

- Reconozca las selecciones del menú (por ejemplo, «Pulse 1» para facturar). Presiona 2 para obtener asistencia»).
- Recopile entradas numéricas (por ejemplo, números de cuenta, PIN y números de confirmación)
- Activa flujos de trabajo o transiciones de estado en los flujos de llamadas
- Vuelva del habla al tono táctil cuando sea necesario

1. Razones a través del contexto de la transmisión de LLM
   - La consulta se envía al agente, que la pasa, junto con los metadatos de la sesión (por ejemplo, el identificador de la persona que llama o el contexto anterior), a un LLM.
   - El LLM genera una respuesta, posiblemente utilizando una estrategia de cadena de pensamiento o una memoria de varios giros si la interacción es continua.
2. Devuelve una respuesta de voz
   - El agente convierte su respuesta en voz mediante la conversión de texto a voz (TTS).
   - Devuelve el audio al usuario a través de un canal de voz.

Capacidades

- Real-time comprensión y generación del habla
- Multilingüe I/O con soporte para STT y TTS
- Integración con las API de telefonía o streaming
- Reconocimiento de la sesión y transferencia de memoria entre turnos

Casos de uso comunes

- Sistemas IVR conversacionales
- Recepcionistas virtuales y planificadores de citas
- Voice-driven agentes del servicio de asistencia
- Asistentes de voz portátiles
- Interfaces de voz para hogares inteligentes y herramientas de accesibilidad

Guía para la implementación

Puede crear este patrón con las siguientes herramientas y Servicios de AWS:

- Amazon Lex V2 o Amazon Transcribe para STT
- Amazon Polly para TTS
- Amazon Chime SDK, Amazon Connect Customer o Amazon Interactive Video Service (Amazon IVS) para streaming y telefonía
- Amazon Bedrock para razonar con Anthropic, AI21 u otros modelos de base
- AWS Lambda para conectar STT, LLM, TTS y el contexto de la sesión

(Opcional) Las mejoras adicionales pueden incluir las siguientes:

- Amazon Kendra o OpenSearch para RAG sensible al contexto
- Amazon DynamoDB para memoria de sesión
- Amazon CloudWatch Logs y AWS X-Ray para la trazabilidad

Resumen

Los agentes de voz y voz son sistemas inteligentes que interactúan a través de conversaciones naturales. Al integrar las interfaces de voz con la infraestructura de razonamiento y transmisión en tiempo real de LLM, los agentes de voz permiten interacciones fluidas, accesibles y escalables.

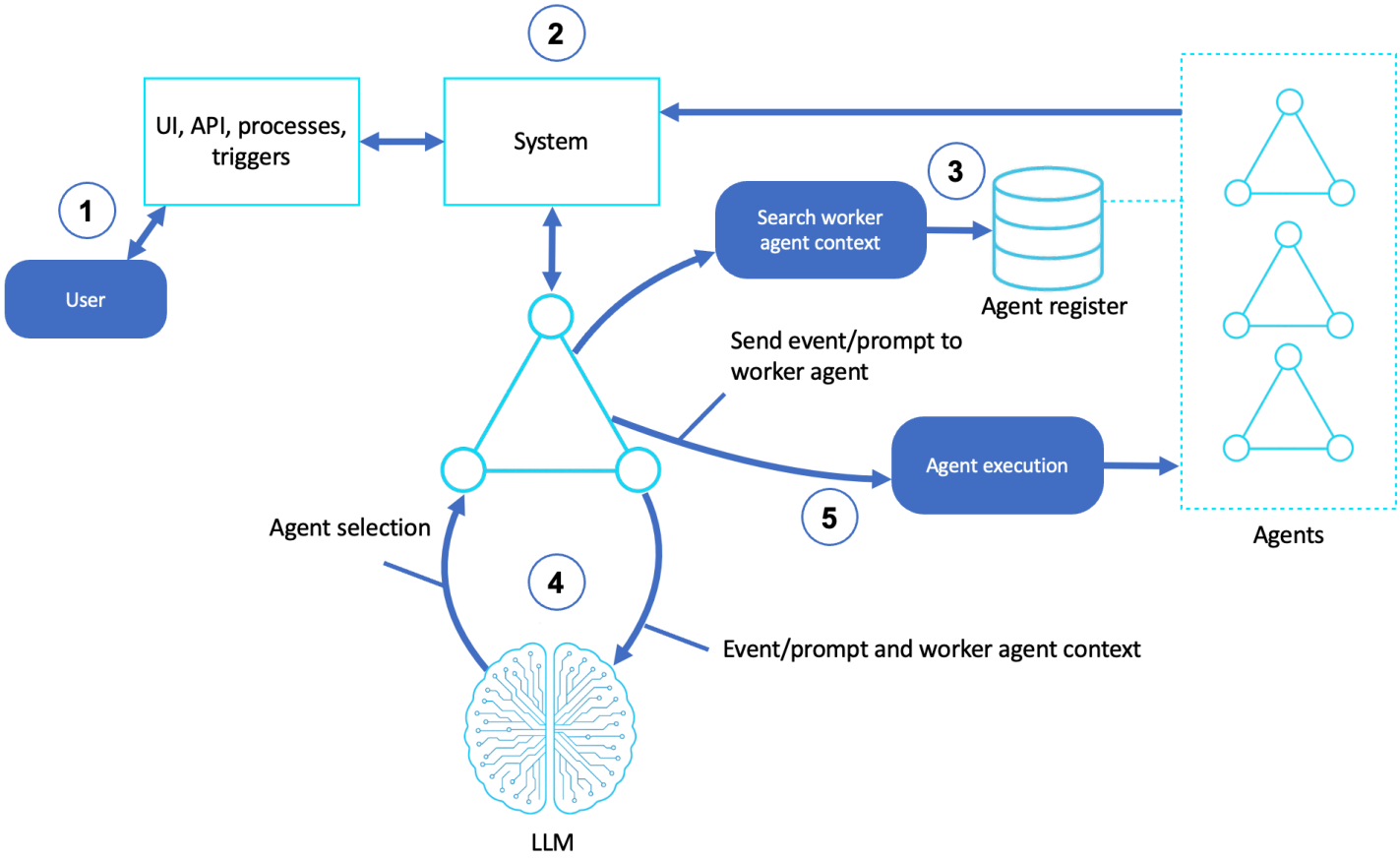
Agentes de orquestación del flujo de trabajo

Los agentes de organización del flujo de trabajo gestionan y coordinan tareas, procesos y servicios de varios pasos en sistemas distribuidos. En lugar de razonar y actuar de forma aislada, estos agentes delegan el trabajo en subagentes u otros sistemas, mantienen el contexto de ejecución y se adaptan en función de los resultados intermedios.

Estos agentes son una parte fundamental de los flujos de automatización. Son particularmente útiles cuando se gestionan tareas de larga duración, composiciones con varios agentes e integraciones entre dominios en las que es necesario utilizar varios agentes y herramientas de forma secuencial o condicional.

Arquitectura

En el siguiente diagrama se muestra un agente de orquestación del flujo de trabajo:



Description (Descripción)

1. Recibe la entrada del usuario
   - Un usuario (o un desencadenador externo) inicia una tarea a través de una interfaz de usuario, una API o un evento del sistema.
2. Maneja los eventos del sistema
   - Un componente del sistema recibe la solicitud y emite un evento o comando que requiere orquestación.
3. Recupera el contexto
   - El agente de flujo de trabajo consulta las bases de conocimiento y los registros de agentes para encontrar el agente de trabajo adecuado para la tarea en función de los metadatos, el dominio y la tasa de éxito anterior.

4. Selecciona un agente de LLM
   - Un LLM ayuda a seleccionar el mejor agente o plan de flujo de trabajo al analizar la descripción de la tarea y las opciones disponibles.
   - También puede formular indicaciones específicas para cada tarea para enviarlas a un agente seleccionado.
5. Delega y ejecuta
   - El agente de trabajo elegido recibe el evento o el mensaje y comienza a ejecutar los comandos.
   - Puede realizar un seguimiento del estado de ejecución, volver a intentarlo en caso de error y pasar los resultados intermedios al siguiente agente de la secuencia.

Capacidades

- Composición de los agentes (por ejemplo, supervisores, agentes colaboradores y herramientas)
- Event-driven o ejecución programada
- Seguimiento de la memoria y el estado a lo largo del tiempo
- Orquestación de tareas jerárquica o paralela (flujos de trabajo síncronos en comparación con los asíncronos)
- Selección y encadenamiento dinámicos de agentes

Casos de uso comunes

- Automatización de varios pasos (por ejemplo, ingesta de datos y generación de informes)
- Enrutamiento y escalamiento del servicio de atención al cliente (por ejemplo, agente como coordinador)
- Los agentes de IA se coordinan con humanos y bots en el mismo ciclo
- Automatiza los procesos empresariales mediante LLM-powered la lógica
- Los sistemas híbridos combinan agentes de IA y herramientas de orquestación tradicionales

Guía para la implementación

Puede crear este patrón con las siguientes herramientas y Servicios de AWS:

- Amazon Bedrock para el razonamiento y la selección de agentes

- AWS Step Functions o Amazon EventBridge para la composición del flujo de trabajo
- AWS Lambda como unidades de ejecución o ejecutores de tareas
- Amazon DynamoDB, Amazon Simple Storage Service (Amazon S3) o Amazon RDS para realizar un seguimiento de los estados y los resultados
- AWS AppFabric o Amazon AppFlow para la coordinación entre sistemas
- (Opcional) Usa Amazon SageMaker run agent para alojar agentes de trabajo de dominios específicos

Resumen

Los agentes de flujo de trabajo coordinan, adaptan y alinean los objetivos en entornos con varios agentes. Esto significa que los agentes de IA pueden colaborar, adaptarse a las condiciones de tiempo de ejecución y ofrecer resultados complejos mediante flujos de trabajo modulares y explicables.

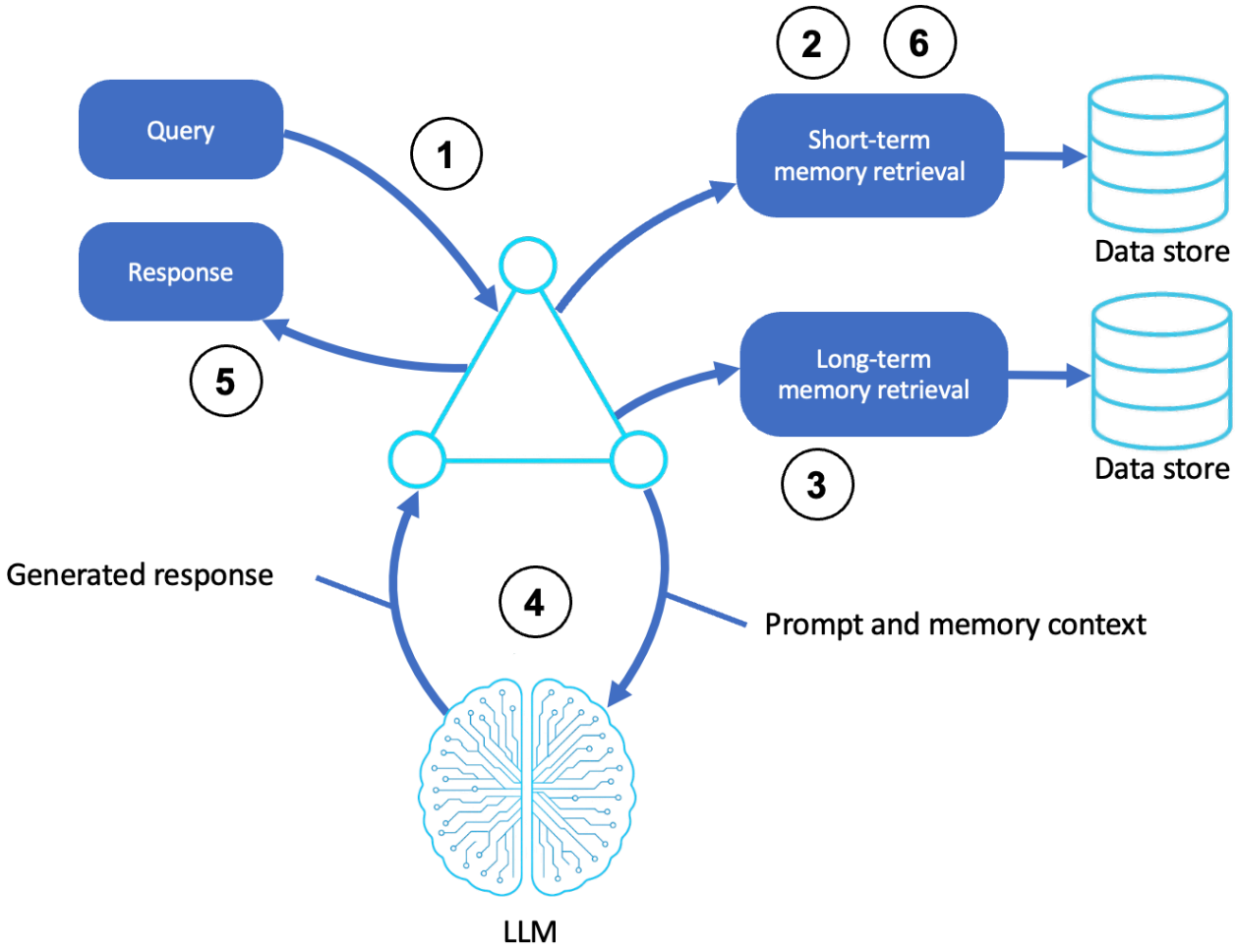
Memory-augmented agentes

Memory-augmented los agentes se han mejorado con la capacidad de almacenar, recuperar y razonar utilizando la memoria a corto y largo plazo. Esto les permite mantener el contexto en múltiples tareas, sesiones e interacciones, lo que produce respuestas más coherentes, personalizadas y estratégicas.

A diferencia de los agentes apátridas, los agentes con memoria aumentada se adaptan haciendo referencia a datos históricos, aprenden de los resultados anteriores y toman decisiones que se ajustan a los objetivos, las preferencias y el entorno del usuario.

Arquitectura

En el siguiente diagrama se muestra un agente con memoria aumentada:



Description (Descripción)

1. Recibe una entrada o un evento
   - El agente recibe una consulta de usuario o un evento del sistema. Puede ser un texto, un desencadenante de la API o un cambio ambiental.
2. Recupera la memoria a corto plazo
   - El agente recupera el historial de conversaciones recientes, el contexto de la tarea o el estado del sistema que sean relevantes para la sesión o el flujo de trabajo.
3. Recupera la memoria a largo plazo
   - El agente consulta la memoria a largo plazo (por ejemplo, bases de datos vectoriales y almacenes de valores clave) para obtener información histórica, como la siguiente:
     - Preferencias de usuario
     - Decisiones y resultados pasados

    - Conceptos, resúmenes o experiencias aprendidas
4. Razones a través del LLM
    - El contexto de la memoria está integrado en el indicador LLM, lo que permite al agente razonar basándose tanto en las entradas actuales como en los conocimientos previos.
5. Genera salidas
    - El agente produce una respuesta, un plan o una acción sensibles al contexto que se personaliza de acuerdo con el historial de tareas y las aportaciones del usuario.
6. Actualiza la memoria
    - La información nueva, como los objetivos actualizados, las señales de éxito y fracaso y las respuestas estructuradas, se almacena para futuras tareas.

Capacidades

- Continuidad de la sesión en todas las conversaciones o eventos
- Persistencia de objetivos a lo largo del tiempo
- Conciencia contextual basada en un estado en evolución
- Adaptabilidad basada en éxitos y fracasos anteriores
- Personalización alineada con las preferencias y el historial del usuario

Casos de uso comunes

- Copilotos conversacionales que recuerdan las preferencias del usuario
- Agentes de codificación que rastrean los cambios en la base de código
- Agentes de flujo de trabajo que se adaptan según el historial de tareas
- Gemelos digitales que evolucionan a partir del conocimiento del sistema
- Investigue los agentes que eviten las recuperaciones redundantes

Implementación de agentes con memoria aumentada

Utilice las siguientes herramientas y Servicios de AWS para los agentes con memoria aumentada:

Capa de memoria	Servicio de AWS	Finalidad

Short-term	Contexto de Amazon DynamoDB, Redis y Amazon Bedrock	Recuperación rápida de los estados de interacción recientes
Long-term (estructurado)	Amazon Aurora, Amazon DynamoDB, Amazon Neptune	Datos, relaciones y registros
Long-term (semántico)	OpenSearch, PostgreSQL, Pinecone	Embedding-based recuperación (es decir, RAG)
Almacenamiento	Amazon S3	Almacenar transcripciones, memorias estructuradas y archivos
Orquestación	AWS Lambda o bien AWS Step Functions	Gestión del ciclo de vida de las inyecciones y actualizaciones de memoria
Razonamiento	Amazon Bedrock	Claude o Mistral antrópicos con indicaciones de memoria

Implementación de mensajes con inyección de memoria

Para integrar la memoria en el razonamiento de los agentes, utilice una combinación de estado estructurado e inyección de contexto con recuperación aumentada:

- Incluya el estado más reciente del agente y el historial de diálogos recientes como entrada estructurada al crear el mensaje para el modelo lingüístico, de modo que pueda razonar con todo el contexto.
- Utilice la generación de recuperación aumentada (RAG) para extraer documentos o hechos relevantes de la memoria a largo plazo.
- Resuma los planes, el contexto y las interacciones anteriores para hacerlos más comprensivos y relevantes.
- Inserte módulos de memoria externos, como almacenes vectoriales o registros estructurados, durante la inferencia para guiar la toma de decisiones.

Resumen

Memory-augmented los agentes mantienen la continuidad del pensamiento al aprender de la experiencia y recordar el contexto del usuario. Estos agentes superan la inteligencia reactiva al utilizar la colaboración a largo plazo, la personalización y el razonamiento estratégico. En términos de inteligencia artificial, la memoria permite a los agentes comportarse más como homólogos digitales adaptables y menos como herramientas apátridas.

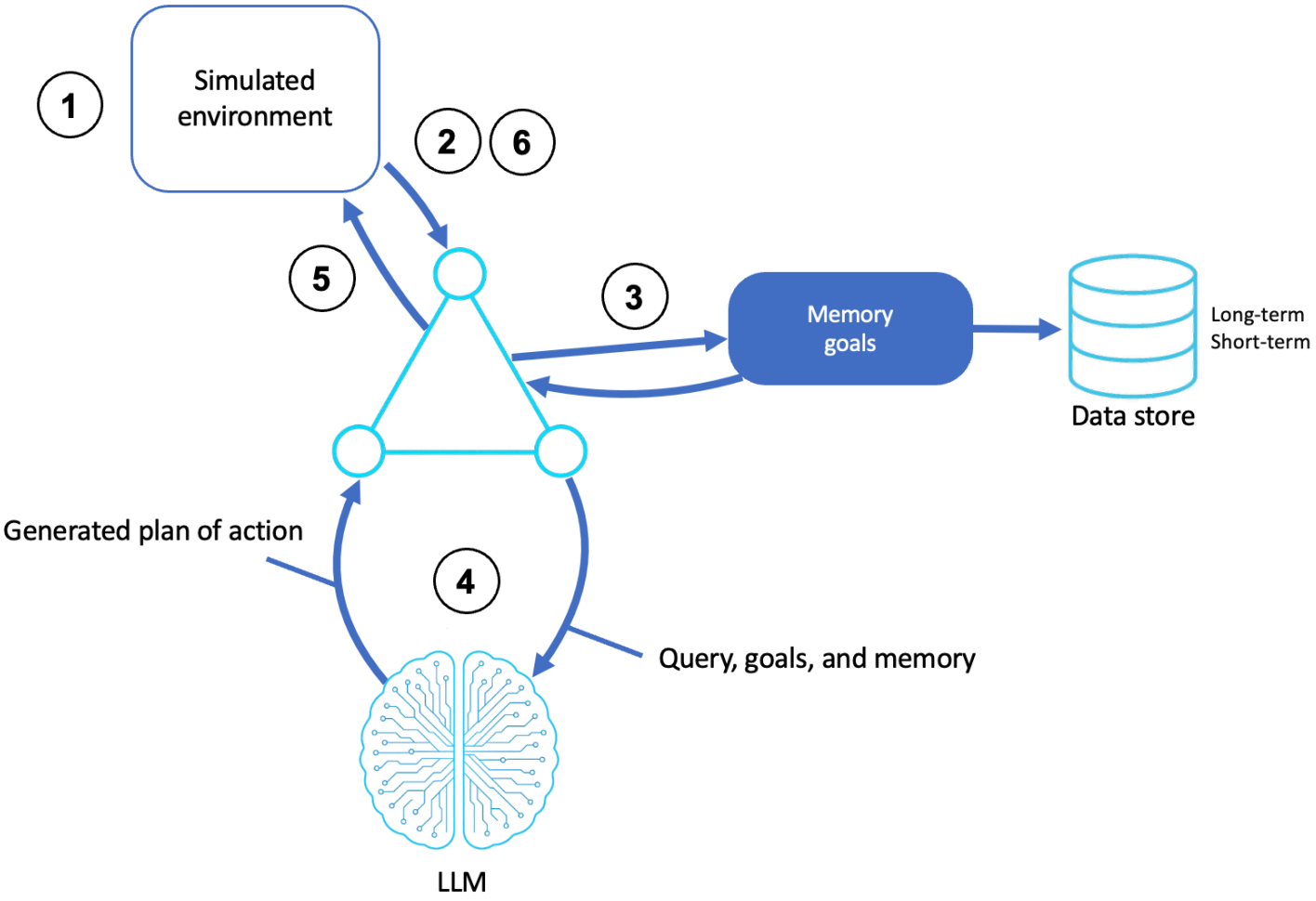
Agentes de simulación y de banco de pruebas

Los agentes de simulación y de banco de pruebas operan en entornos virtualizados o controlados donde razonan, actúan y aprenden. Estos agentes simulan el comportamiento, modelan los resultados y entrenan las estrategias en entornos repetibles antes de aplicarlos a entornos del mundo real.

Este patrón es útil para el desarrollo iterativo, el aprendizaje por refuerzo (RL), la evaluación autónoma de la toma de decisiones y las pruebas de comportamiento emergentes. Los agentes de simulación suelen operar en circuitos cerrados, reciben información de su entorno y ajustan su comportamiento en consecuencia, lo que los convierte en elementos fundamentales para las tareas que implican el razonamiento espacial, el control en tiempo real o la dinámica de sistemas complejos.

Arquitectura

El siguiente diagrama muestra un agente de simulación o de banco de pruebas:



Description (Descripción)

1. Inicia un entorno
   - El agente inicia un entorno simulado (por ejemplo, un mundo 3D, un motor de física, un entorno limitado CLI o un flujo de datos sintéticos).
   - El agente se carga en el entorno con una tarea, un objetivo o una política iniciales.
2. Percibe al agente
   - El agente percibe el estado actual mediante telemetría de simulación (por ejemplo, emulación de sensores, cámara virtual y registros estructurados).
3. Recupera el objetivo y la memoria
   - El agente recupera el objetivo asignado, las instrucciones del escenario o la meta contextual.
   - También puede recuperar la memoria anterior, incluida la siguiente:
     - Long-term estrategias o políticas

    - Mapas ambientales o restricciones conocidas
    - Éxitos o fracasos pasados de simulaciones similares
4. Motivos y planes
    - Un LLM interpreta el estado simulado, los objetivos de la tarea y el conocimiento aprendido.
    - Genera un plan de acción o un comando de control.
5. Ejecuta acciones simuladas
    - El agente ejecuta el plan, modifica el estado, navega por el espacio o interactúa con entidades virtuales.
6. Aprende
    - El agente evalúa los resultados de las acciones
    - Según la configuración del agente, puede hacer lo siguiente:
        - Realice RL
        - Registre los resultados para realizar ajustes en el futuro
        - Adapte las estrategias en tiempo real

Capacidades

- Funciona en entornos sintéticos o virtuales
- Soporta el aprendizaje mediante prueba y error, el perfeccionamiento de políticas y el modelado de sistemas
- Low-risk pruebas de comportamiento, gestión de fallos y casos extremos
- Permite analizar el comportamiento de los agentes emergentes en configuraciones con varios agentes
- Soporta tanto el control en circuito cerrado como la exploración humana en bucle

Casos de uso comunes

- Aprendizaje reforzado para robótica, drones y juegos
- Entrenamiento de vehículos autónomos en carreteras virtuales
- Interfaces de usuario o CLI simuladas para escenarios de banco DevOps de pruebas
- Experimentos de comportamiento emergente en simulaciones sociales

- Validación de seguridad de la lógica de decisión antes de la producción

Guía para la implementación

Puede crear un agente de simulación y de banco de pruebas utilizando las siguientes herramientas y: Servicios de AWS

Componente	Servicio de AWS	Finalidad
Entorno	Amazon ECS, Amazon EC2 o un simulador personalizado en Amazon Studio Lab SageMaker	Ejecute mundos virtuales (Gazebo, Unity, Unreal) o CLI de entorno aislado
Lógica de agentes	Amazon Bedrock SageMaker, Amazon o AWS Lambda	LLM-based planificadores o agentes de RL
Bucle de retroalimentación	Amazon SageMaker Reforcement Learning CloudWatch, Amazon o registros personalizados	Seguimiento de recompensas, puntuación de resultados y registro del comportamiento
Memoria y reproducción	Amazon S3, Amazon DynamoDB o Amazon RDS	Datos persistentes de estado, historial de episodios o escenarios
Visualización	Cuadernos de Amazon o CloudWatch tableros de Amazon SageMaker	Observe los cambios en las políticas, los resultados y las métricas de capacitación

Las siguientes son aplicaciones adicionales:

- AWS SimSpace Weaverpara simulaciones espaciales a gran escala
- AWS IoT Corepara probar dispositivos de sombra
- Amazon SageMaker Experiments para la evaluación de agentes y la evaluación comparativa

Resumen

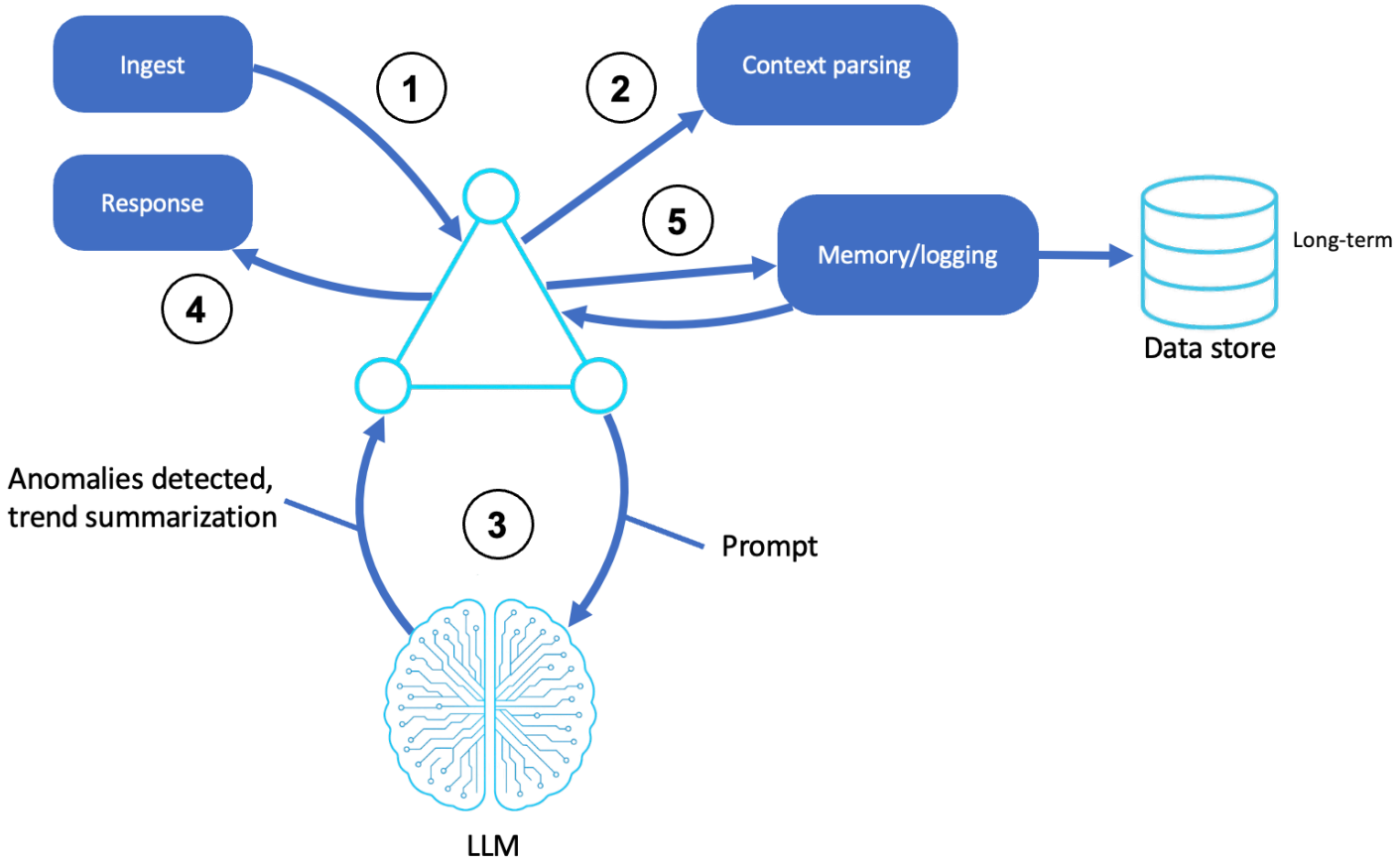
Los agentes de simulación y de banco de pruebas sirven para una exploración estructurada antes de desplegarlos en los sistemas de producción. Utilice estos agentes para entrenar políticas de navegación autónoma, probar procesos empresariales en entornos sintéticos y evaluar los patrones de coordinación de los enjambres.

Agentes de observación y monitoreo

Los agentes observadores y de monitoreo observan de forma pasiva los sistemas, los entornos y las interacciones para detectar patrones, generar información y activar acciones. Como observadores inteligentes, mejoran las alertas, los diagnósticos y las auditorías sin iniciar directamente el comportamiento.

Estos agentes se destacan cuando el monitoreo tradicional carece de adaptabilidad o razonamiento, especialmente en lo que respecta al AI-in-the-loop monitoreo, la detección de anomalías, la supervisión del cumplimiento y la inteligencia de seguridad. Los agentes observadores son detectores de eventos que monitorean continuamente la telemetría del sistema y las interacciones de los usuarios. El agente depende de la percepción, la interpretación y la intensificación condicional o la presentación de informes.

Arquitectura



Description (Descripción)

1. Telemetría de ingestión
   - El agente recibe información de una o más fuentes del sistema, como las siguientes:
     - Registros (aplicación, infraestructura, seguridad)
     - Métricas (rendimiento, latencia, uso)
     - Eventos (llamadas a la API, acciones de los usuarios, datos de sensores)
2. Analice el contexto
   - La entrada sin procesar se analiza, estructura y enriquece con metadatos, como la marca de tiempo, la identidad del actor, el estado del sistema y el identificador de seguimiento.
3. ¿Por qué usar LLM

- El agente utiliza un LLM o módulo lógico para interpretar las entradas analizadas mediante la identificación de anomalías, el resumen de las tendencias y la correlación entre trazas distribuidas o ventanas temporales.

4. Clasifica o alerta
   - El agente determina si el comportamiento observado justifica lo siguiente:
     - ¿Una alerta o escalada
     - Una actualización de un informe o un panel
     - Un desencadenante de respuesta (por ejemplo, una corrección automática y la aplicación de políticas)
5. Registre la memoria o los bucles de retroalimentación
   - El agente almacena eventos y decisiones para el aprendizaje a largo plazo, las auditorías o como referencia futura para otros agentes.

Capacidades

- Pasivo y no invasivo (el agente no actúa directamente)
- Altamente escalable y asíncrono
- AI-driven correlación entre señales ruidosas o distribuidas
- Soporta la auditoría, el cumplimiento y la información en tiempo real
- Puede alimentar los flujos de trabajo humanos o de los agentes intermedios

Casos de uso comunes

- AI-augmented observabilidad de microservicios y API
- Supervisión de la desviación del modelo, la infracción de las políticas o el comportamiento fuera de banda
- Análisis de la actividad del cliente o resúmenes de interacciones
- Agentes de revisión de código que supervisan las confirmaciones o las implementaciones
- Supervisión de los registros de seguridad o cumplimiento mediante el razonamiento LLM

Guía para la implementación

Puede crear un observador y un agente de monitoreo utilizando las siguientes herramientas y Servicios de AWS:

Componente	Servicio de AWS	Finalidad
Ingestión de eventos	Amazon EventBridge, Amazon CloudWatch Logs, Amazon Kinesis, Amazon S3	Ingiera telemetría estructurada y no estructurada
Procesamiento previo	AWS Lambda, AWS Glue, AWS Step Functions	Transforme los datos sin procesar en indicaciones estructuradas
Motor de razonamiento	Amazon Bedrock, Amazon SageMaker, AWS Lambda	Analice eventos, clasifique el comportamiento, genere información
Almacenamiento y memoria	Amazon S3, Amazon DynamoDB, OpenSearch	Observaciones, resúmenes y resultados persistentes
Alertas y escalamiento	Amazon SNS, AWS AppFabric Amazon EventBridge	Activa sistemas o agentes descendentes

Las siguientes son aplicaciones adicionales:

- AWS Security Hub CSPMpara la supervisión de los registros de seguridad
- Amazon Quick para visualizar las salidas de los agentes

Resumen

Los agentes observadores y de monitoreo rastrean los sistemas y los comportamientos en tiempo real. Detectan anomalías, auditan la seguridad y recopilan información sobre las operaciones mediante la identificación de patrones que las personas o las reglas podrían pasar por alto. Esta capacidad ayuda a crear sistemas que pueden adaptarse a las condiciones cambiantes y tomar decisiones basadas en un análisis exhaustivo de los datos.

Multi-agent colaboración

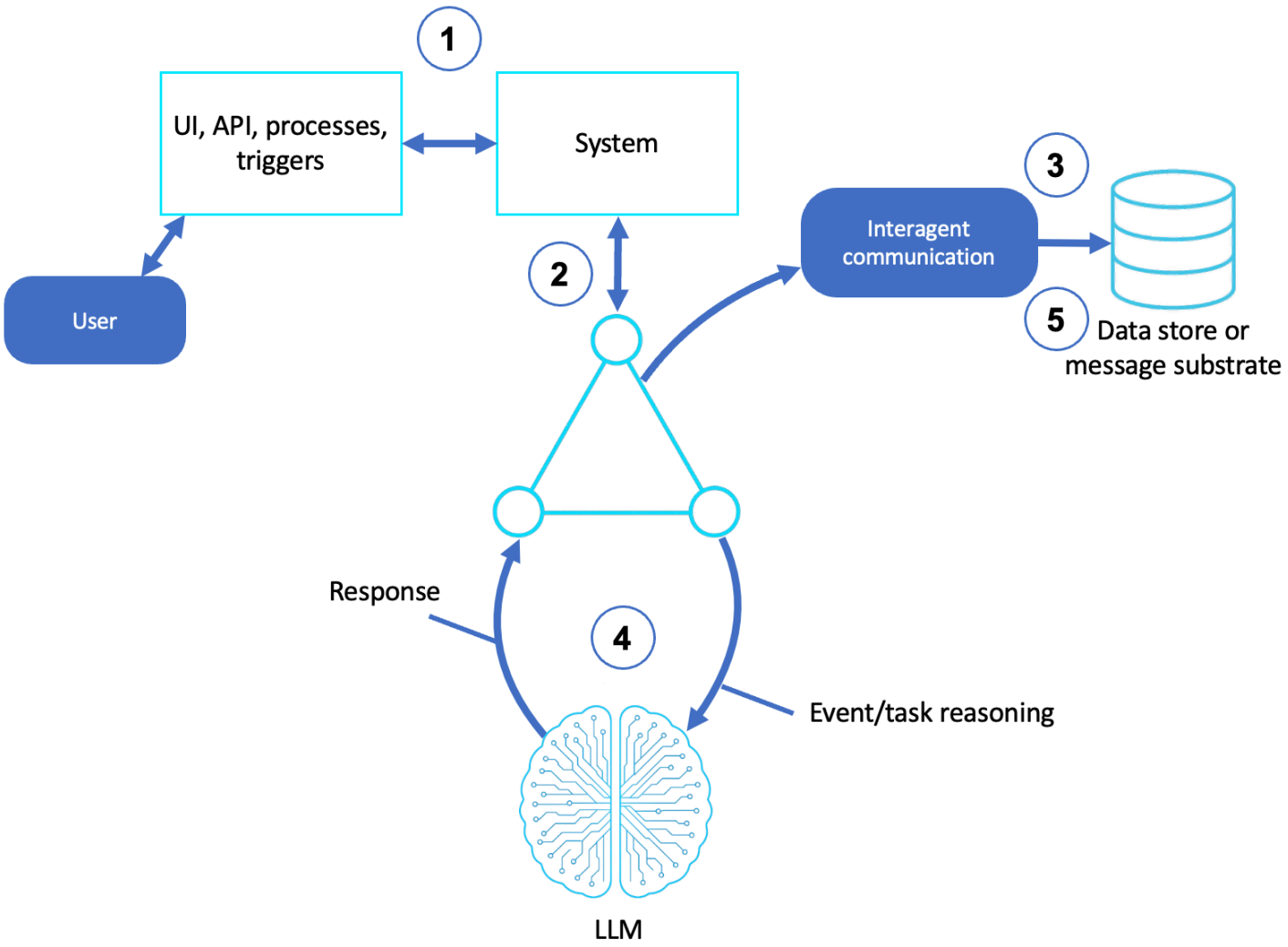
Multi-agent la colaboración se refiere a un patrón en el que varios agentes autónomos, cada uno con una función, especialización u objetivo distintos, negocian para resolver tareas complejas. Estos agentes pueden operar de forma independiente o con otros agentes al compartir información, dividir las responsabilidades y razonar colectivamente para alcanzar un objetivo.

Este patrón difiere del de los agentes de flujo de trabajo, que coordinan y delegan tareas de forma centralizada a los agentes subordinados en un flujo estructurado. Por el contrario, la colaboración entre múltiples agentes hace hincapié en la coordinación entre pares o emergente al permitir la adaptabilidad, el paralelismo y la división de la cognición. En la siguiente tabla se compara la colaboración entre varios agentes y los agentes de flujo de trabajo:

Característica	Agentes de flujo de trabajo	Finalidad
Controlar	Coordinador centralizado	Compañeros descentralizados, distribuidos o basados en roles
Interacción	Un agente delega y realiza un seguimiento de la ejecución	Varios agentes negocian, comparten y se adaptan
Diseño	Secuencia de tareas predefinida	Distribución de tareas flexible y emergente
Coordinación	Orquestación procedimental	Interacciones cooperativas o competitivas
Casos de uso	Automatización de procesos empresariales	Razonamiento complejo, exploración y estrategias emergentes

Arquitectura

El siguiente diagrama muestra la colaboración entre varios agentes:



Description (Descripción)

1. Inicia una tarea
   - Un usuario o un sistema emite un objetivo o un problema de alto nivel.
   - Un «administrador», agente o contexto iniciador define el objetivo.
2. Asigna o descubre roles
   - Los agentes se autoasignan (razonamiento o lógica simbólica) o se les delega (agentes de eventos) otras funciones, como las de planificador, investigador, ejecutor, crítico o explicador.
3. Se comunica con otros agentes
   - Los agentes se comunican a través de la memoria compartida, las colas de mensajes o el encadenamiento de mensajes.

- Pueden debatir, realizar consultas o proponerse subtareas entre sí.

4. Utiliza un razonamiento especializado
   - Cada agente usa su propio modelo o lógica de dominio para resolver su parte del problema.
   - Los agentes pueden usar LLM con instrucciones y memoria específicas para cada función.
5. Coordina los resultados o los objetivos
   - Los agentes sintetizan las contribuciones en una respuesta, un plan o una acción finales.
   - (Opcional) Un agente supervisor puede validar o resumir el resultado sintetizado.

Capacidades

- Peer-level agentes con funciones o habilidades especializadas
- Comportamiento emergente a través de la comunicación o la negociación
- Procesamiento paralelo de problemas complejos o multifacéticos
- Apoya la deliberación, la autocorrección y la iteración reflexiva
- Modele la dinámica social, la colaboración científica o las funciones del equipo empresarial

Casos de uso comunes

- Equipos de investigación autónomos (agente de búsqueda, resumidor y validador)
- Desarrollo de software (planificador, codificador y probador)
- Modelado de escenarios empresariales (finanzas, políticas y cumplimiento)
- Negociación, licitación o razonamiento multipartidista
- Tareas multimodales (imagen, texto y lógica)

Guía para la implementación

Puede crear un sistema multiagente mediante las siguientes herramientas y: Servicios de AWS

Componente	Servicio de AWS	Finalidad
Alojamiento de agentes	Amazon Bedrock, Amazon SageMaker, AWS Lambda	Aloje a agentes individuales LLM-driven

Capa de comunicación	Amazon SQS, Amazon, EventBridge AWS AppFabric	Mensajería y coordinación entre agentes
Memoria compartida	Amazon DynamoDB, Amazon S3 o OpenSearch	Multi-agent memoria o sistema de pizarra
Capa de orquestación	AWS Step Functions, AWS Lambda canalizaciones	Lógica de inicio, tiempo de espera, retroceso y reintento
Identificación del agente	Agentes de Amazon Bedrock (función definida) y API conversada de AWS AppConfig Amazon Bedrock (agentes ajenos a Amazon Bedrock)	Role-based invocación de herramientas o agentes y aplicación de límites
Interacción emergente	Amazon EventBridge Pipelines o registros de agentes	Habilite el enrutamiento o la escalación dinámicos de tareas

Resumen

Multi-agent la colaboración distribuye las tareas de resolución de problemas entre agentes modulares y basados en funciones. A diferencia de la organización del flujo de trabajo, los patrones de colaboración utilizan inteligencia, resiliencia y escalabilidad emergentes que reflejan la forma en que las personas resuelven los problemas. Es especialmente valioso para dominios abiertos, tareas creativas, razonamiento multimodal y entornos que se benefician de diversas perspectivas.

Conclusión

Los patrones analizados anteriormente ilustran los enfoques fundamentales de las implementaciones de la IA de los agentes en el mundo real. Desde el razonamiento básico hasta la inteligencia con memoria aumentada, cada patrón está configurado de forma única para la percepción, la cognición y la acción, y se basa en la autonomía, la asincronía y la agencia.

Estos patrones comparten vocabularios y modelos técnicos para crear sistemas inteligentes y orientados a objetivos. Ya sea que un patrón esté integrado en una interfaz de usuario, se organice

mediante servicios en la nube o se coordine entre equipos de agentes, cada patrón es adaptable y modular.

Conclusiones

- Los patrones de los agentes son componibles: la mayoría de los agentes del mundo real combinan dos o más patrones (por ejemplo, un agente de voz con memoria y razonamiento basados en herramientas).
- El diseño del agente es contextual: elija patrones en función de la superficie de interacción, la complejidad de la tarea, la tolerancia a la latencia y las restricciones específicas del dominio.
- AWSLa implementación nativa se puede lograr: con Amazon Bedrock SageMaker, Amazon y las arquitecturas basadas en eventos AWS Lambda AWS Step Functions, todos los patrones de agentes se pueden ofrecer a escala.

Flujos de trabajo LLM

En cuanto a los patrones de los agentes, exploramos los patrones más comunes de los agentes de la IA, cada uno creado en torno a un conjunto de capacidades modulares: percepción, acción, aprendizaje y cognición. En el centro del módulo cognitivo de muchos patrones de agentes se encuentra un modelo de lenguaje amplio (LLM) que es capaz de razonar, planificar y tomar decisiones. Sin embargo, invocar un LLM por sí solo no es suficiente para producir un comportamiento inteligente y orientado a objetivos.

Para realizar tareas complejas de forma fiable, los agentes deben integrar el LLM en un flujo de trabajo estructurado, en el que las capacidades del modelo se amplíen con herramientas, memoria, ciclos de planificación y lógica de coordinación. Estos flujos de trabajo de LLM permiten a un agente desglosar los objetivos, enrutar las subtareas, llamar a servicios externos, reflexionar sobre los resultados y coordinarse con otros agentes.

Este capítulo presenta los patrones de diseño básicos para crear módulos cognitivos robustos, ampliables e inteligentes basados en la LLM, organizados en torno a flujos de trabajo reutilizables.

En esta sección

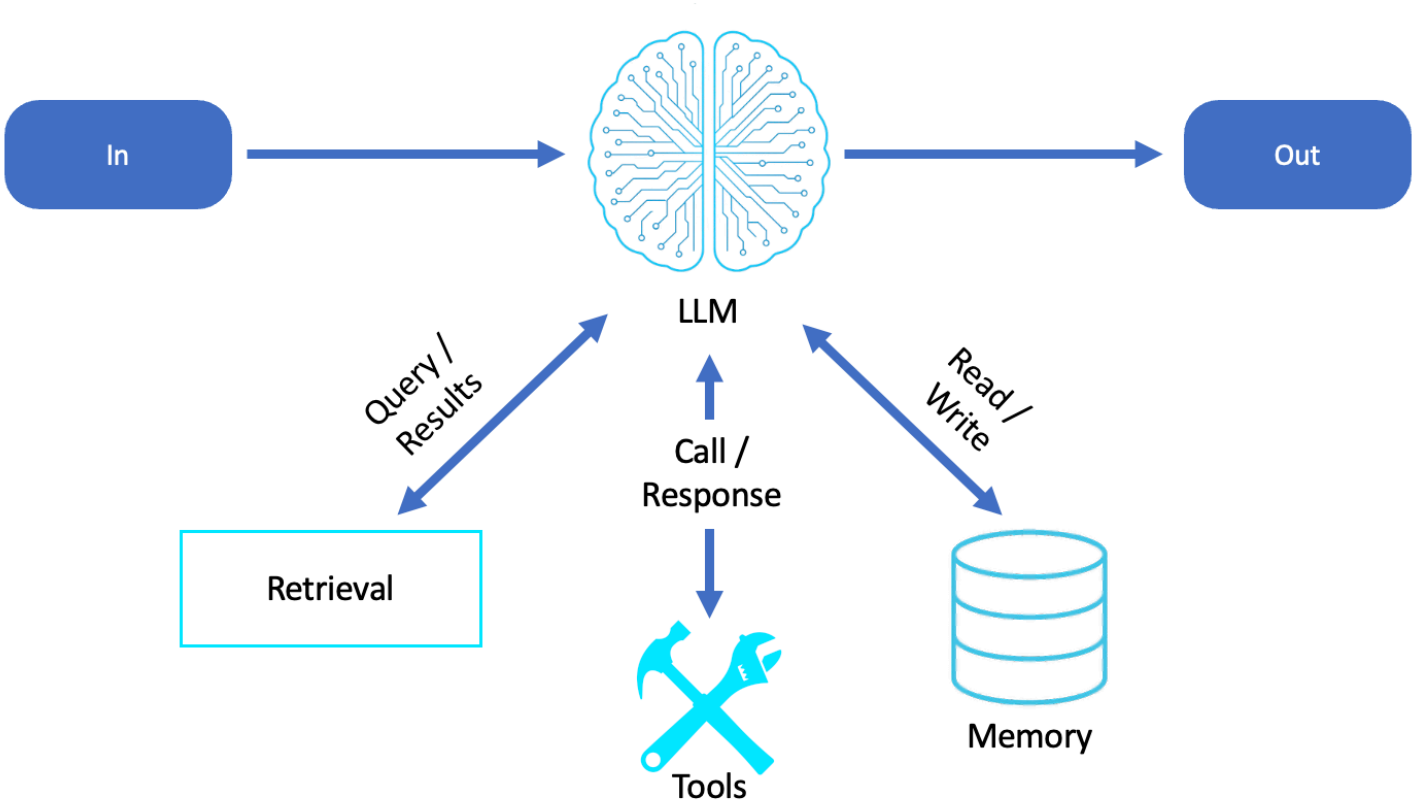
Descripción general de la cognición aumentada con LLM

En esencia, el módulo cognitivo de un agente de software puede considerarse un LLM repleto de mejoras. El agente puede utilizar los siguientes componentes básicos para razonar eficazmente en su entorno:

- Solicitud: enmarcar la entrada utilizando el contexto, las instrucciones, los ejemplos y la memoria

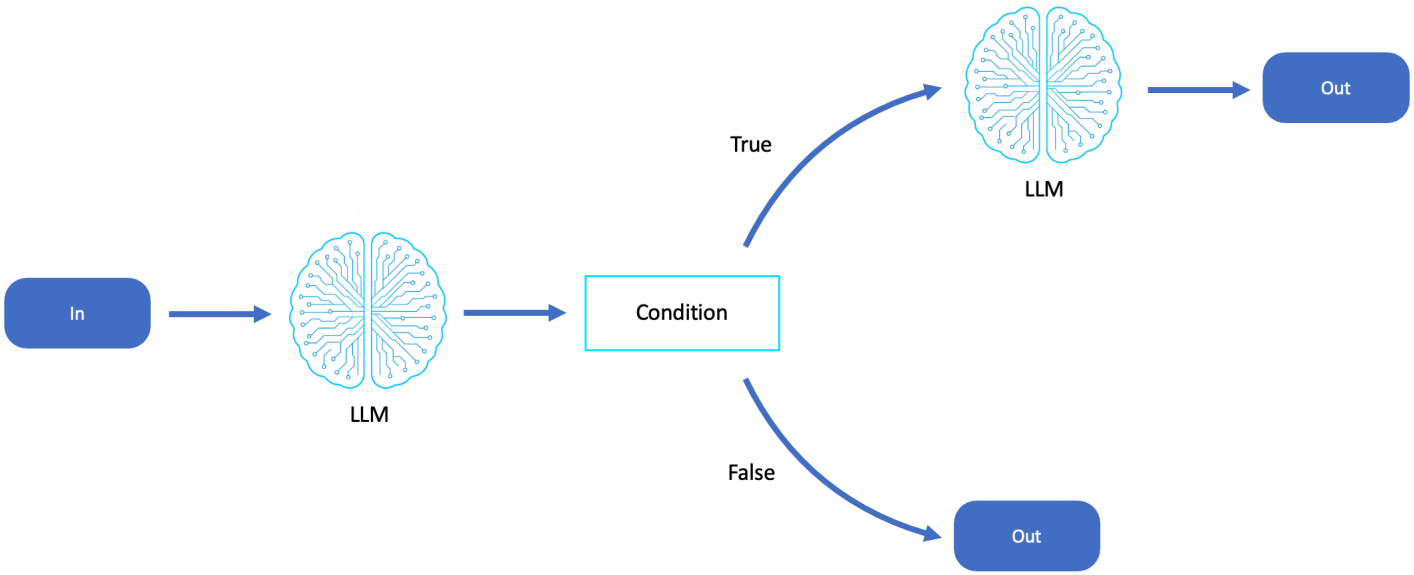
- Recuperación: proporcionar up-to-date conocimientos específicos de un dominio a la solicitud LLM mediante la búsqueda vectorial o la memoria semántica, por ejemplo, mediante la generación de recuperación aumentada (RAG)
- Uso de herramientas: permite al LLM invocar o llamar a funciones para recuperar información o actuar en función de ella APIs
- Memoria: incorporar un estado persistente o basado en una sesión en el ciclo de razonamiento, ya sea mediante bases de datos estructuradas o resúmenes contextuales

Estos aumentos se componen de flujos de trabajo que definen cómo se utiliza el LLM a lo largo del tiempo y en todas las tareas, lo que lo transforma de un motor sin estado en un agente de razonamiento dinámico.



Flujo de trabajo para un encadenamiento rápido

El encadenamiento rápido descompone las tareas complejas en una secuencia de pasos, donde cada paso es una invocación discreta de LLM que procesa o se basa en el resultado del anterior.



El flujo de trabajo de encadenamiento rápido es adecuado para escenarios en los que las tareas se pueden dividir de forma lógica en pasos de razonamiento secuenciales y en los que los resultados intermedios sirven de base para la siguiente etapa. Destaca en los flujos de trabajo que requieren un pensamiento estructurado, una transformación progresiva o un análisis por capas, como la revisión de documentos, la generación de código, la extracción de conocimientos y el perfeccionamiento del contenido.

Description (Descripción)

- La complejidad de la tarea supera la ventana de contexto o la profundidad de razonamiento de una sola llamada de LLM.
- Los resultados de un paso (por ejemplo, el análisis, el resumen o la planificación) se convierten en insumos para una fase de generación o decisión de seguimiento.
- Se necesita transparencia y control en todas las etapas del razonamiento (por ejemplo, resultados intermedios auditables).
- Desea conectar la lógica externa de validación, filtrado o enriquecimiento entre los pasos.
- Es ideal para los agentes que trabajan en ciclos de razonamiento de tipo continuo, como los agentes de investigación, los asistentes de redacción, los sistemas de planificación y los copilotos de varias etapas.

Capacidades

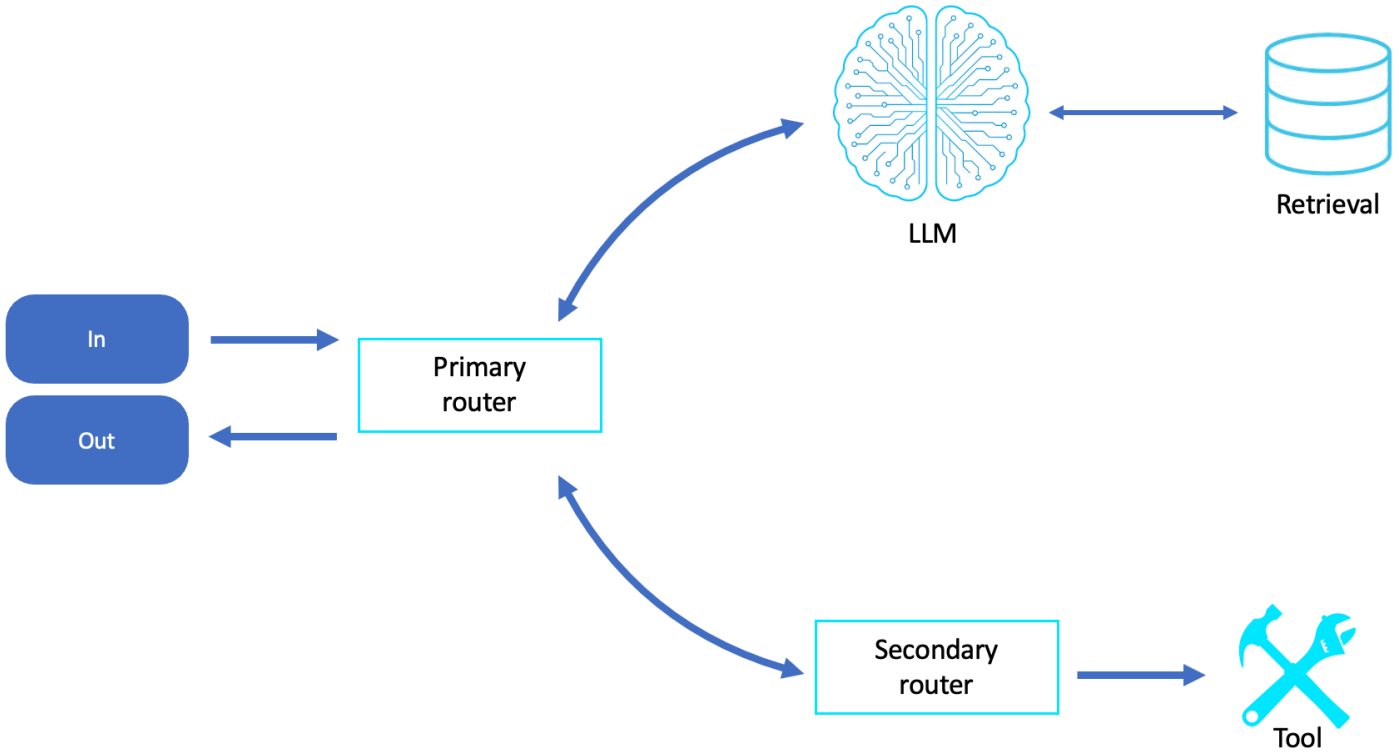
- Cadenas lineales o ramificadas de llamadas LLM
- Los resultados intermedios se transmiten como datos estructurados o se incluyen en las instrucciones de seguimiento
- Se puede orquestar con corredores o con AWS Step Functions un AWS Lambda agente específico

Casos de uso comunes

- Tareas de razonamiento de varios pasos (por ejemplo, «resumir, criticar, reescribir»)
- Los asistentes de investigación sintetizan resultados estratificados (por ejemplo, «buscar, extraer datos, responder a una pregunta»)
- Procesos de generación de código («generar un plan, escribir el código, probar el código, explicar el resultado»)

Flujo de trabajo para enrutamiento

En el patrón de enrutamiento, un clasificador o agente de enrutamiento utiliza un LLM para interpretar la intención o la categoría de una consulta y, a continuación, enruta la entrada a un agente o tarea posterior especializado.



El flujo de trabajo de enrutamiento se utiliza en situaciones en las que un agente debe clasificar rápidamente la intención de entrada, el tipo de tarea o el dominio y, a continuación, delegar la solicitud a un subagente, herramienta o flujo de trabajo especializados. Resulta especialmente útil en el caso de los agentes de capacidades, como los que actúan como asistentes generales, son los encargados de dar acceso a las funciones empresariales o las interfaces de IA orientadas al usuario que abarcan varios dominios.

El enrutamiento es particularmente efectivo cuando:

- Clasifica las solicitudes en función de una variedad de tareas (por ejemplo, búsqueda, resumen, reserva o cálculos).
- Las entradas deben preprocesarse o normalizarse antes de entrar en flujos de trabajo más especializados.
- Los diferentes tipos de entrada (por ejemplo, imágenes frente a texto, consultas estructuradas frente a consultas no estructuradas) requieren un tratamiento personalizado.
- Un agente actúa como una centralita conversacional y delega tareas a agentes especializados o microservicios.

- Este flujo de trabajo es común en los copilotos de dominios específicos, los bots de atención al cliente, los enrutadores de servicios empresariales y los agentes multimodales, donde el despacho inteligente determina tanto la calidad como la eficiencia del comportamiento de los agentes.

Capacidades

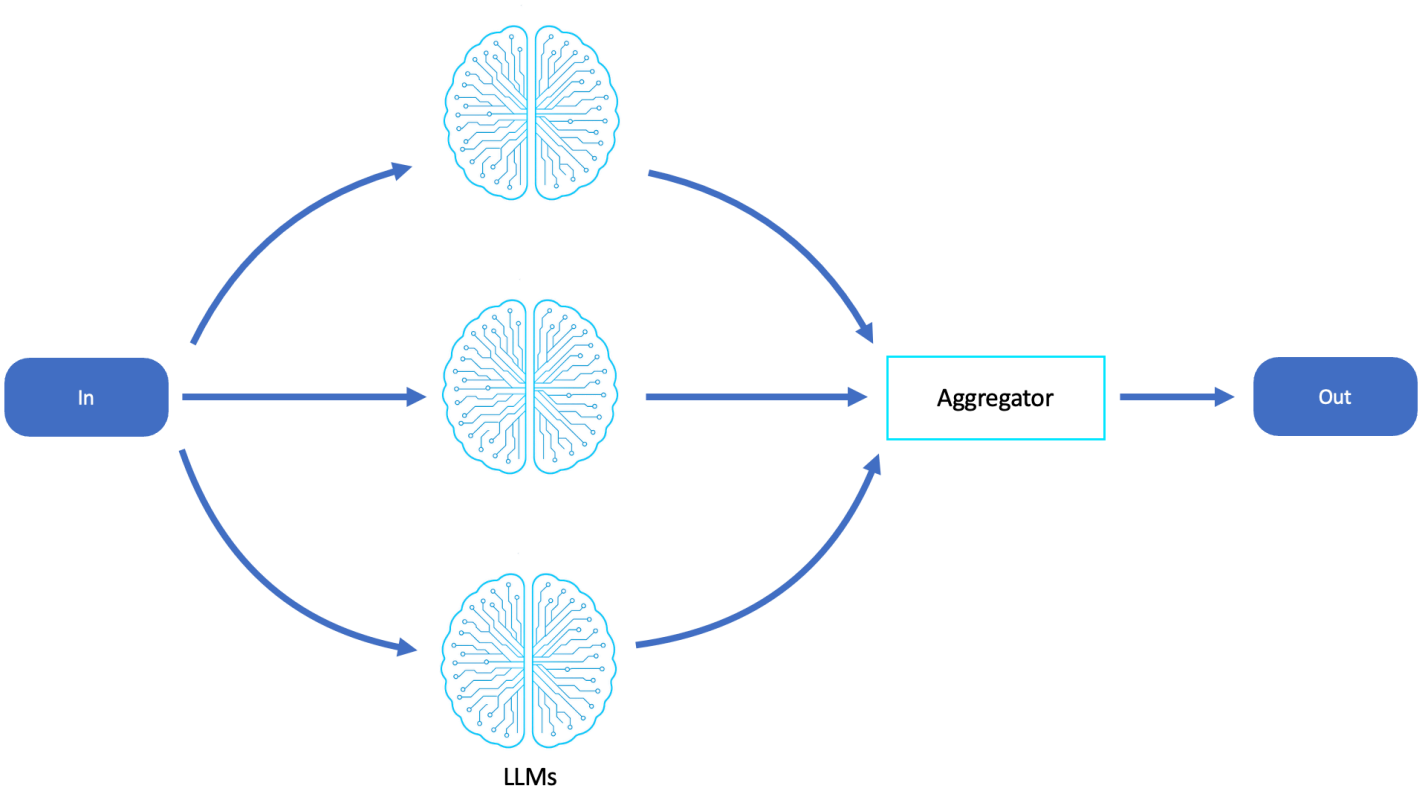
- Un LLM de primer paso actúa como despachador
- Las rutas pueden invocar flujos de trabajo distintos o incluso otros patrones de agentes
- Soporta la expansión modular de las capacidades

Casos de uso comunes

- Asistentes multidominio («¿se trata de una cuestión legal, médica o financiera?»)
- Árboles de decisión mejorados con el razonamiento LLM
- Selección dinámica de herramientas (por ejemplo, búsqueda frente a generación de código)

Flujo de trabajo para la paralelización

Este flujo de trabajo implica dividir una tarea en subtareas independientes que pueden ser gestionadas simultáneamente por varios agentes o llamadas de LLM. Luego, los resultados se agregan mediante programación y se sintetizan en un resultado.



El flujo de trabajo de paralelización se utiliza cuando una tarea se puede dividir en subtareas independientes y no secuenciales que se pueden procesar simultáneamente, lo que mejora significativamente la eficiencia, el rendimiento y la escalabilidad. Es especialmente eficaz en espacios problemáticos con muchos datos, orientados a lotes o con múltiples perspectivas, en los que el agente debe analizar o generar contenido a partir de múltiples entradas.

La paralelización es particularmente eficaz cuando:

- Las subtareas no dependen de los resultados intermedios de las demás, lo que les permite ejecutarse en paralelo sin coordinación.
- Una tarea implica repetir el mismo proceso de razonamiento en varios elementos (por ejemplo, resumir varios documentos o evaluar una lista de opciones).
- Se exploran múltiples hipótesis o perspectivas en paralelo para promover la diversidad, la creatividad o la solidez.
- Debe reducir la latencia de las solicitudes de gran volumen o alta frecuencia mediante la ejecución simultánea de la LLM.
- Este flujo de trabajo se suele utilizar en agentes de procesamiento de documentos, motores de encuestas o comparación, resumidores de lotes, sesiones de intercambio de ideas con varios

agentes y tareas de clasificación o etiquetado escalables, especialmente cuando el razonamiento rápido y paralelo es una ventaja de rendimiento.

Capacidades

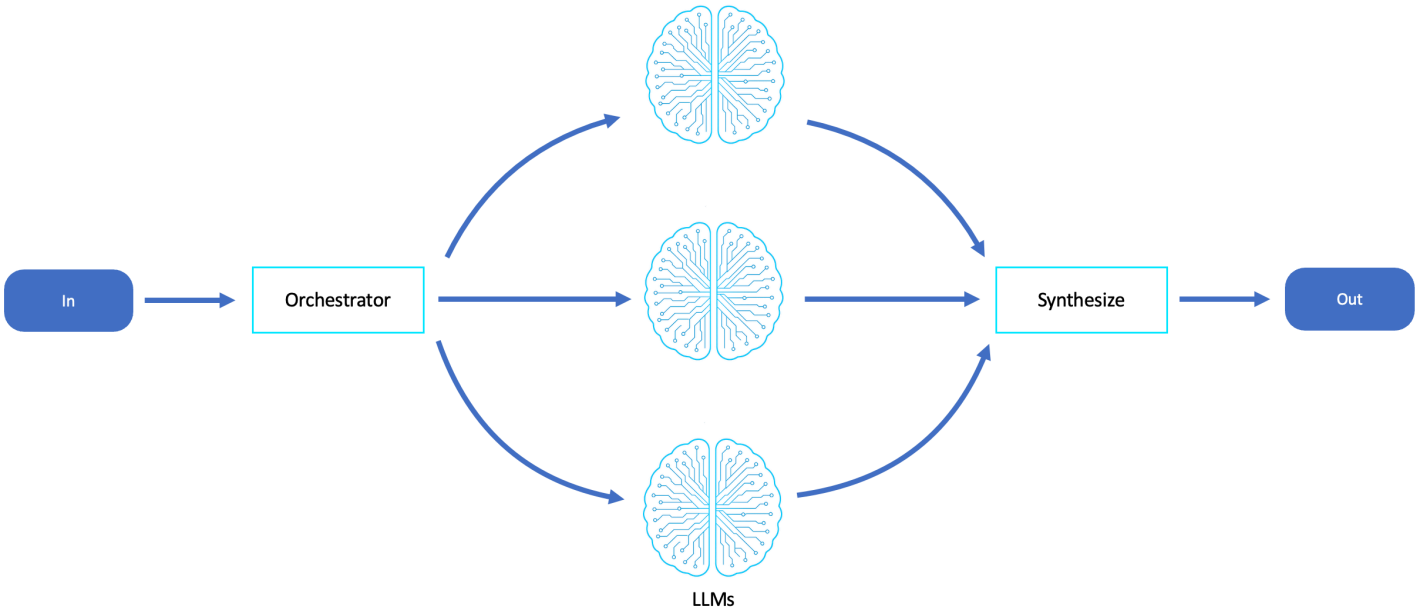
- Ejecución paralela de tareas de LLM (mediante el uso de AWS Lambda, AWS Fargate o un estado del mapa) AWS Step Functions
- Requiere alinear, validar o deduplicar los resultados en la etapa de síntesis
- Muy adecuado para bucles de agentes sin estado

Casos de uso comunes

- Analizar varios documentos o perspectivas en paralelo
- Generar diversos borradores, resúmenes o planes
- Acelerar el rendimiento en los trabajos por lotes

Flujo de trabajo para la orquestación

Un agente orquestador central utiliza un LLM para planificar, descomponer y delegar subtareas en agentes o modelos trabajadores especializados, cada uno con una función específica o experiencia en el campo. Esto refleja las estructuras de los equipos humanos y apoya el comportamiento emergente de varios agentes.



El flujo de trabajo de orquestación es ideal para escenarios complejos, jerárquicos o multidisciplinarios que requieren una descomposición estructurada y una ejecución especializada. Es especialmente adecuado para tareas que requieren división del trabajo, en las que es mejor que agentes con distintas capacidades, conocimientos o conjuntos de herramientas se encarguen mejor de los diferentes subcomponentes de una tarea.

Este flujo de trabajo es particularmente eficaz cuando:

- Las tareas se pueden dividir en subtareas que varían en cuanto a alcance, tipo o razonamiento (por ejemplo, planificar, investigar, implementar y probar).
- Un LLM o un metaagente debe coordinar a otros agentes, supervisar el progreso y sintetizar los resultados.
- Desea modular las responsabilidades de los agentes para permitir la escalabilidad, la reutilización y el ajuste especializado.
- El sistema requiere un comportamiento basado en roles, que imite la forma en que los equipos humanos (por ejemplo, directores de proyectos, desarrolladores y revisores) trabajan en colaboración.

Orchestration es ideal para agentes de planificación con múltiples turnos, copilotos de desarrollo de software, agentes de procesos empresariales y ejecutores de proyectos autónomos. Resulta especialmente útil cuando se implementan sistemas con varios agentes que requieren un desglose

centralizado de las tareas pero una lógica de ejecución distribuida, lo que permite una mayor extensibilidad y un comportamiento más explicable en todos los niveles de agentes.

Capacidades

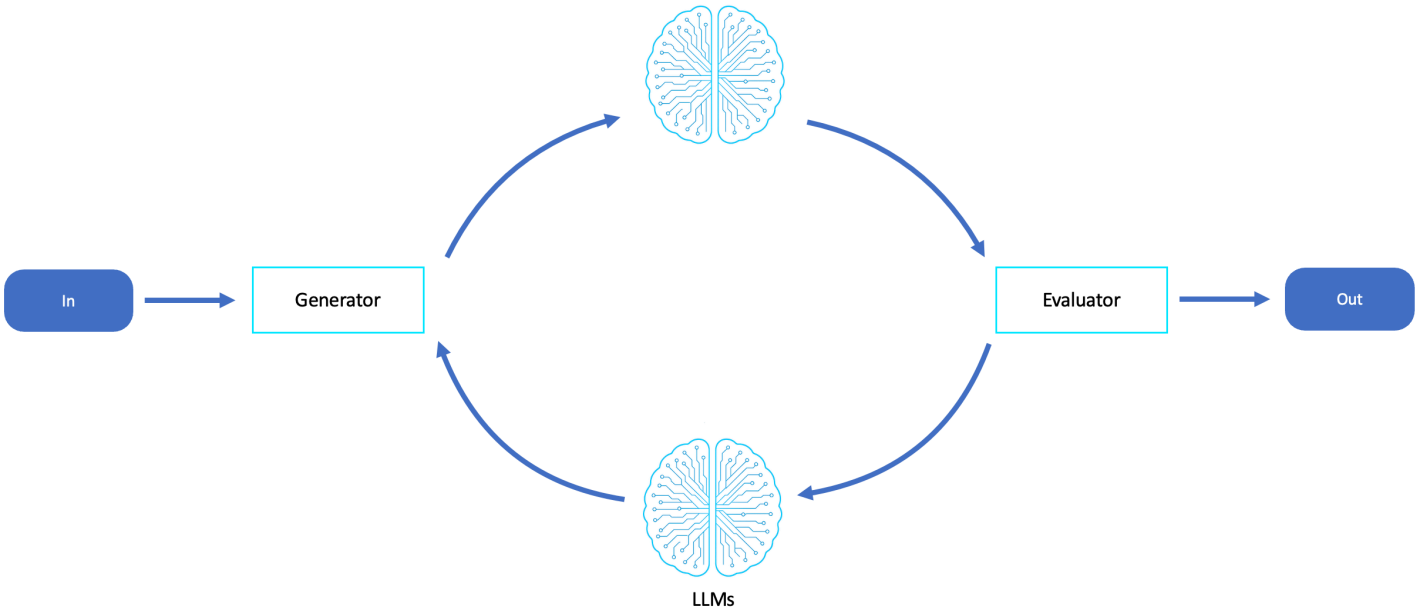
- Orchestrator realiza un metarrazonamiento de objetivos
- Los agentes de trabajo pueden incluir el acceso a las herramientas, la memoria o las solicitudes específicas de un dominio
- Puede ser jerárquico (es decir, delegación de tareas en varios niveles)

Casos de uso comunes

- Directores de proyectos, investigadores coordinadores, redactores y agentes de control de calidad
- Copilotos de codificación que combinan la planificación, la ejecución y las pruebas
- Agentes que supervisan las cadenas de herramientas o los patrones de acceso a las API

Flujo de trabajo para evaluadores y ciclos de reflexión y refinamiento

Este flujo de trabajo proporciona un circuito de retroalimentación en el que un LLM genera un resultado y otro evalúa o critica el resultado. Esto promueve la autorreflexión, la optimización y las mejoras iterativas.



El flujo de trabajo del evaluador es ideal para escenarios en los que la calidad, la precisión y la alineación de los resultados son importantes y en los que la generación de una sola pasada no es fiable o insuficiente. Este flujo de trabajo destaca cuando los agentes deben autocriticarse, repetir y refinar sus resultados, ya sea para cumplir con un estándar más alto de corrección o para explorar alternativas mejoradas en función de los comentarios.

Este flujo de trabajo es particularmente eficaz cuando:

- El resultado incluye métricas de calidad subjetivas (por ejemplo, estilo, tono y legibilidad) o criterios objetivos (por ejemplo, corrección, seguridad y rendimiento).
- El agente debe razonar haciendo concesiones, evaluar las limitaciones u optimizar la búsqueda de una meta.
- Necesita redundancia y garantía de calidad integradas, especialmente en los ámbitos regulados, orientados al cliente o creativos.
- Human-in-the-loop la revisión es cara o no está disponible, y se desea una validación autónoma.

Este flujo de trabajo se utiliza para la generación de contenido, la síntesis y revisión del código, la aplicación de políticas, la comprobación de la alineación, el ajuste de las instrucciones y el posprocesamiento del RAG. También es útil para los agentes que se mejoran a sí mismos, ya que la retroalimentación continua ayuda a dar forma a mejores respuestas a lo largo del tiempo para crear ciclos de decisión autónomos y confiables.

Casos de uso comunes

- Los agentes del equipo rojo se comparan con los del equipo azul
- Agentes que generan, evalúan y revisan códigos o planes
- Control de calidad, detección de alucinaciones y control del estilo

Capacidades

- Soporta la generación y la evaluación disociadas utilizando diferentes modelos (por ejemplo, Claude para la generación y Mistral para la evaluación)
- La retroalimentación está estructurada y se utiliza para generar resultados revisados
- Soporta múltiples iteraciones o umbrales de convergencia

Conclusión

LLMs proporcionan el núcleo cognitivo de los agentes de software modernos, pero la invocación de modelos sin procesar no es suficiente para lograr una inteligencia sólida, controlable y útil. Para pasar de la generación de resultados al razonamiento estructurado y a un comportamiento alineado con los objetivos, LLMs debe integrarse en patrones de flujo de trabajo intencionales que definan la forma en que los modelos procesan las entradas, gestionan los contextos y coordinan las acciones.

Los flujos de trabajo de la LLM introducen las bases para construir el módulo cognitivo de un agente:

- El encadenamiento rápido divide el razonamiento complejo en pasos modulares y auditables.
- El enrutamiento permite una clasificación inteligente de las tareas y una delegación específica.
- La paralelización acelera el rendimiento y promueve un razonamiento diverso.
- La orquestación de agentes estructura la colaboración entre varios agentes mediante la descomposición de tareas y la ejecución basada en roles.
- El evaluador (bucle reflect-refine) permite la superación personal, el control de calidad y la comprobación de la alineación.

Cada flujo de trabajo representa un patrón componible que se puede adaptar a las necesidades del agente, a la complejidad de la tarea y a las expectativas del usuario. Estos flujos de trabajo no se excluyen mutuamente. Son componentes básicos que a menudo se combinan en arquitecturas

híbridas que respaldan el razonamiento dinámico, la coordinación multiagente y la confiabilidad de nivel empresarial.

A medida que pase al siguiente capítulo sobre los patrones de flujo de trabajo de los agentes, estos flujos de trabajo LLM volverán a aparecer como estructuras integradas en sistemas más grandes, lo que favorecerá la delegación de objetivos, la organización de herramientas, los ciclos de decisión y la autonomía del ciclo de vida. Dominar estos flujos de trabajo de LLM es esencial para diseñar agentes de software que no se limiten a predecir el texto, sino que también razonen, se adapten y actúen con determinación.

Patrones de flujo de trabajo de los agentes

Los patrones de flujo de trabajo de Agentic integran agentes de software modulares con flujos de trabajo estructurados de modelos de lenguaje grande (LLM), lo que permite razonar y actuar de forma autónoma. Si bien se inspiran en las arquitecturas tradicionales sin servidor y basadas en eventos, estos patrones cambian la lógica central del código estático a los agentes potenciados por la LLM, lo que mejora la adaptabilidad y la toma de decisiones contextuales. Esta evolución transforma las arquitecturas de nube convencionales, pasando de ser sistemas deterministas a sistemas capaces de realizar interpretaciones dinámicas y ampliaciones inteligentes, manteniendo al mismo tiempo los principios fundamentales de escalabilidad y capacidad de respuesta.

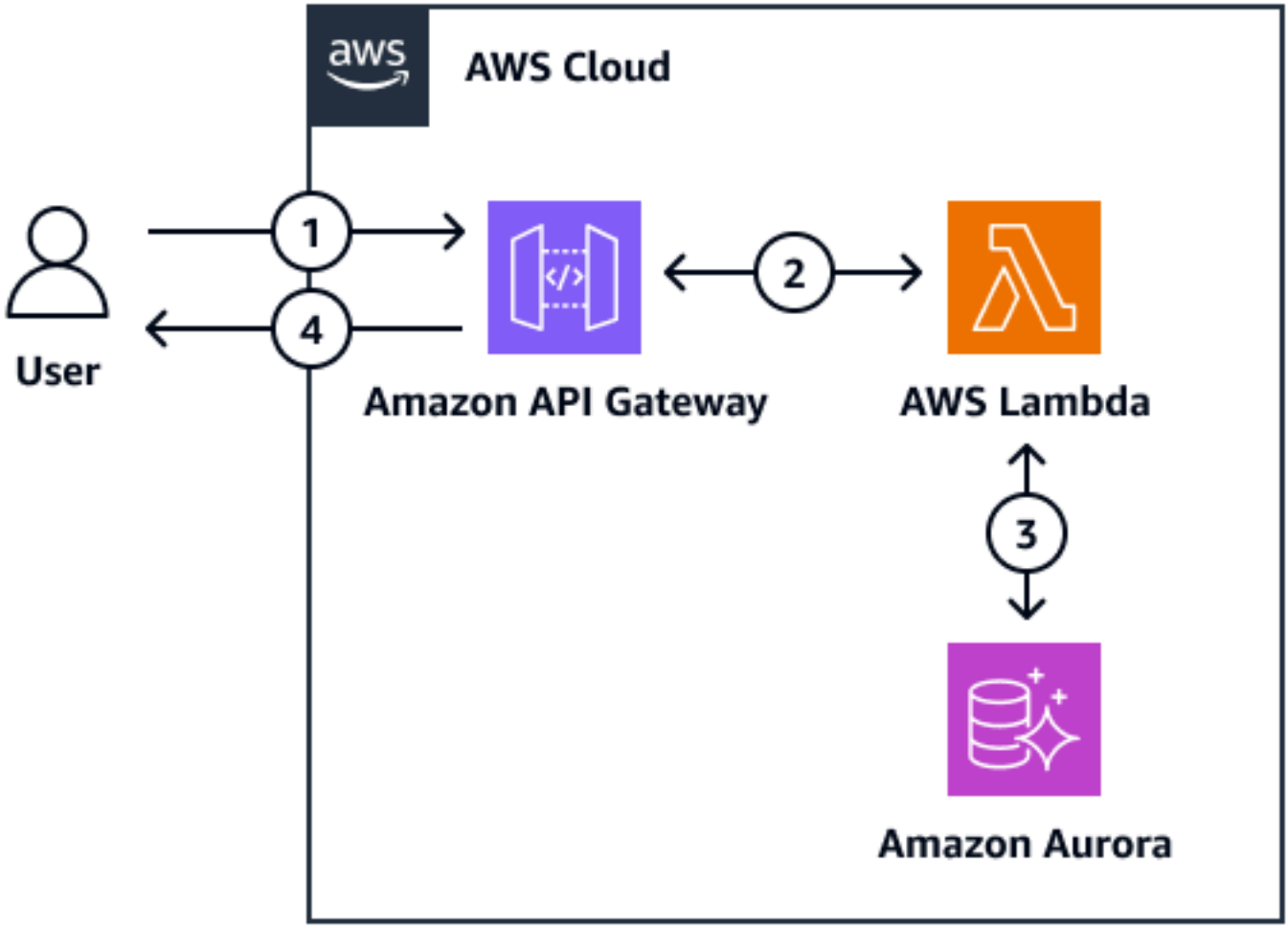
En esta sección

Desde sistemas basados en eventos hasta sistemas con cognición aumentada

Las arquitecturas de nube modernas, especialmente las que se basan en los principios de la ausencia de servidores y las basadas en eventos, se han basado tradicionalmente en patrones como el enrutamiento, la distribución y el enriquecimiento para crear sistemas escalables y con capacidad de respuesta. Los sistemas de inteligencia artificial de las agencias se basan en estos fundamentos y, al mismo tiempo, los reformulan en torno al razonamiento y la flexibilidad cognitiva aumentados por la LLM. Este enfoque permite capacidades de automatización y resolución de problemas más sofisticadas, lo que podría revolucionar la forma en que se gestionan las tareas complejas en los entornos de nube.

Arquitectura basada en eventos

El siguiente diagrama muestra un sistema distribuido típico:

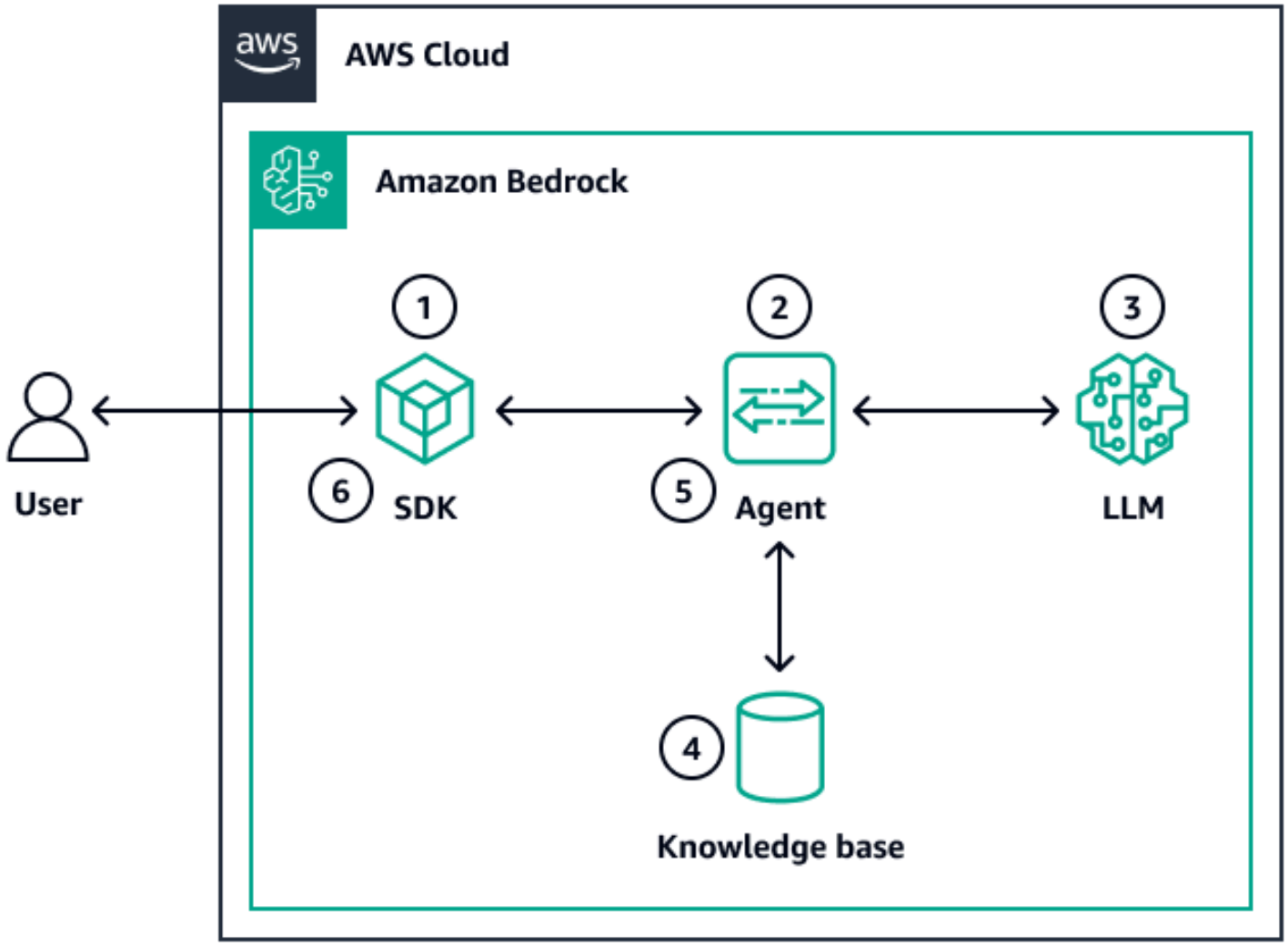


1. Un usuario envía una solicitud a Amazon API Gateway.
2. Amazon API Gateway enruta la solicitud a una AWS Lambda función.
3. AWS Lambda enriquece los datos consultando una base de datos de Amazon Aurora
4. Amazon API Gateway devuelve la carga útil enriquecida a la persona que llama.

Esta estructura es fiable y escalable, pero es fundamentalmente estática. Las reglas empresariales y las rutas lógicas deben codificarse de forma explícita, y la adaptación a los contextos cambiantes o a la información incompleta es limitada.

Flujos de trabajo con cognición aumentada

Las arquitecturas de los agentes añaden un aumento cognitivo a un sistema basado en eventos. El siguiente diagrama muestra el equivalente de un agente:



1. Un usuario envía una consulta a través de una llamada al SDK o a la API.
2. Un agente de Amazon Bedrock recibe la consulta.
3. El agente interpreta la consulta invocando un LLM
4. El agente realiza un enriquecimiento semántico buscando en la base de conocimiento de Amazon Bedrock u otras fuentes de datos externas.
5. El LLM sintetiza una respuesta rica en contexto y alineada con los objetivos.
6. El sistema devuelve una respuesta sintetizada al usuario.

En este flujo, el LLM utiliza la lógica, entiende la intención, recupera y combina el contexto relevante y, a continuación, decide la mejor manera de responder. Este patrón refleja el patrón de enriquecimiento tradicional, en el que los mensajes se complementan con datos externos antes de seguir enrutándolos. Sin embargo, en los sistemas de agencia, este enriquecimiento no es una búsqueda estática. Por el contrario, el enriquecimiento es dinámico, está guiado semánticamente y está impulsado por un propósito.

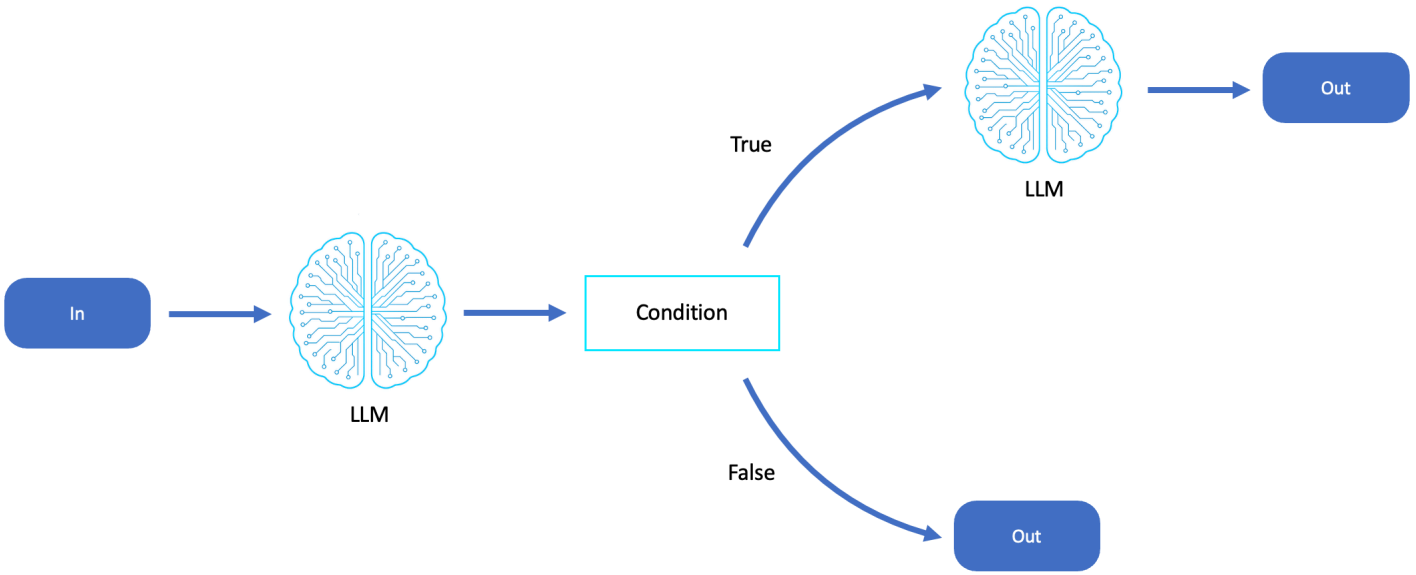
Perspectivas fundamentales

Cada flujo de trabajo de LLM se puede asignar a un patrón de flujo de trabajo de un agente, que refleja y evoluciona los estilos de arquitectura tradicionales basados en eventos. Un elemento básico de los flujos de trabajo de los agentes es la capacidad de aumentar el contexto de un LLM con datos, herramientas y memoria. Esto crea un ciclo de razonamiento informado, adaptable y alineado con la intención del usuario. Mientras que los sistemas tradicionales enriquecen los mensajes con datos de búsqueda, los sistemas de agencia permiten que el software actúe menos como guiones y más como colaboradores inteligentes.

Patrones de saga que se encadenan rápidamente

Al reinventar el encadenamiento rápido de LLM como una saga basada en eventos, descubrimos un nuevo modelo operativo: los flujos de trabajo se distribuyen, se pueden recuperar y coordinar semánticamente entre agentes autónomos. Cada paso de respuesta rápida se reformula como una tarea atómica, se emite como un evento, lo consume un agente especializado y se enriquece con metadatos contextuales.

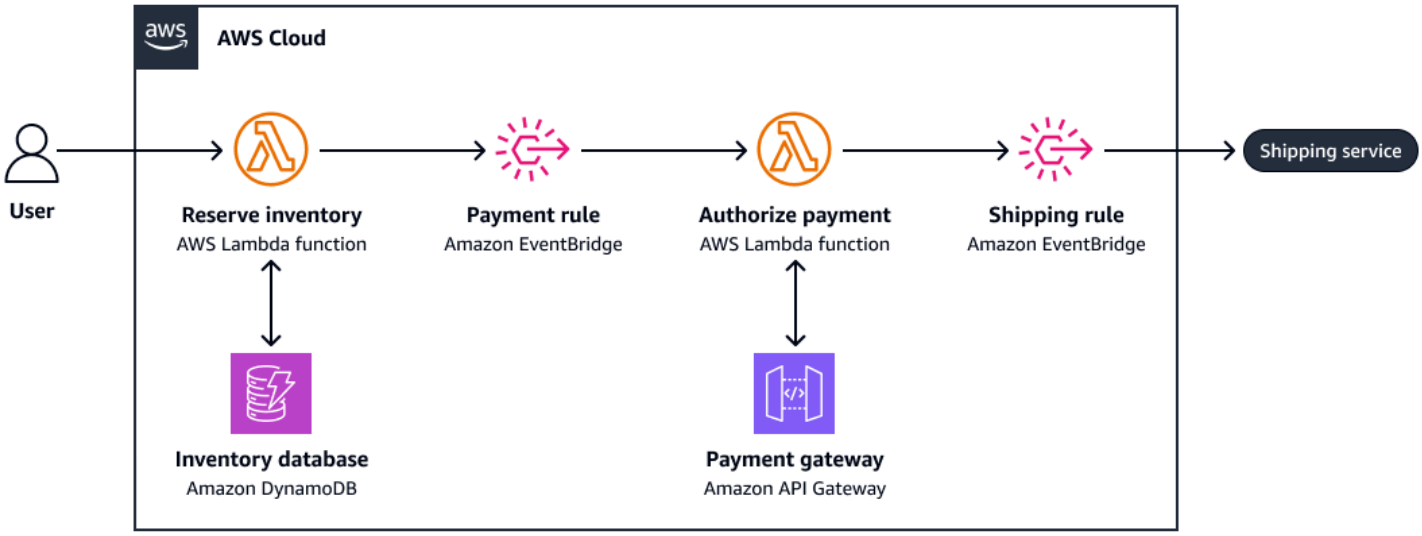
El siguiente diagrama es un ejemplo de encadenamiento de mensajes LLM:



Coreografía de la saga

El patrón coreográfico de la saga es un enfoque de implementación en sistemas distribuidos que no tiene un coordinador central. En su lugar, cada servicio o componente publica eventos que desencadenan la siguiente acción del flujo de trabajo. Este patrón se utiliza ampliamente en los sistemas distribuidos para gestionar las transacciones en varios servicios. En una saga, el sistema ejecuta una serie de transacciones locales coordinadas. Si una falla, el sistema activa acciones compensatorias para mantener la coherencia.

El siguiente diagrama es un ejemplo de coreografía de una saga:



1. Inventario de reserva
2. Autoriza el pago
3. Crea una orden de envío

Si el paso 3 no funciona, el sistema invoca acciones compensatorias (por ejemplo, cancelar un pago o liberar el inventario).

Este patrón es especialmente valioso en las arquitecturas basadas en eventos, en las que los servicios están poco acoplados y los estados deben resolverse de forma coherente a lo largo del tiempo, incluso en caso de que se produzca un fallo parcial.

Patrón de encadenamiento rápido

El encadenamiento rápido se parece al patrón de la saga tanto en estructura como en propósito. Ejecuta una serie de pasos de razonamiento que se desarrollan de forma secuencial, preservando el contexto y permitiendo retrocesos y revisiones.

Coreografía de agentes

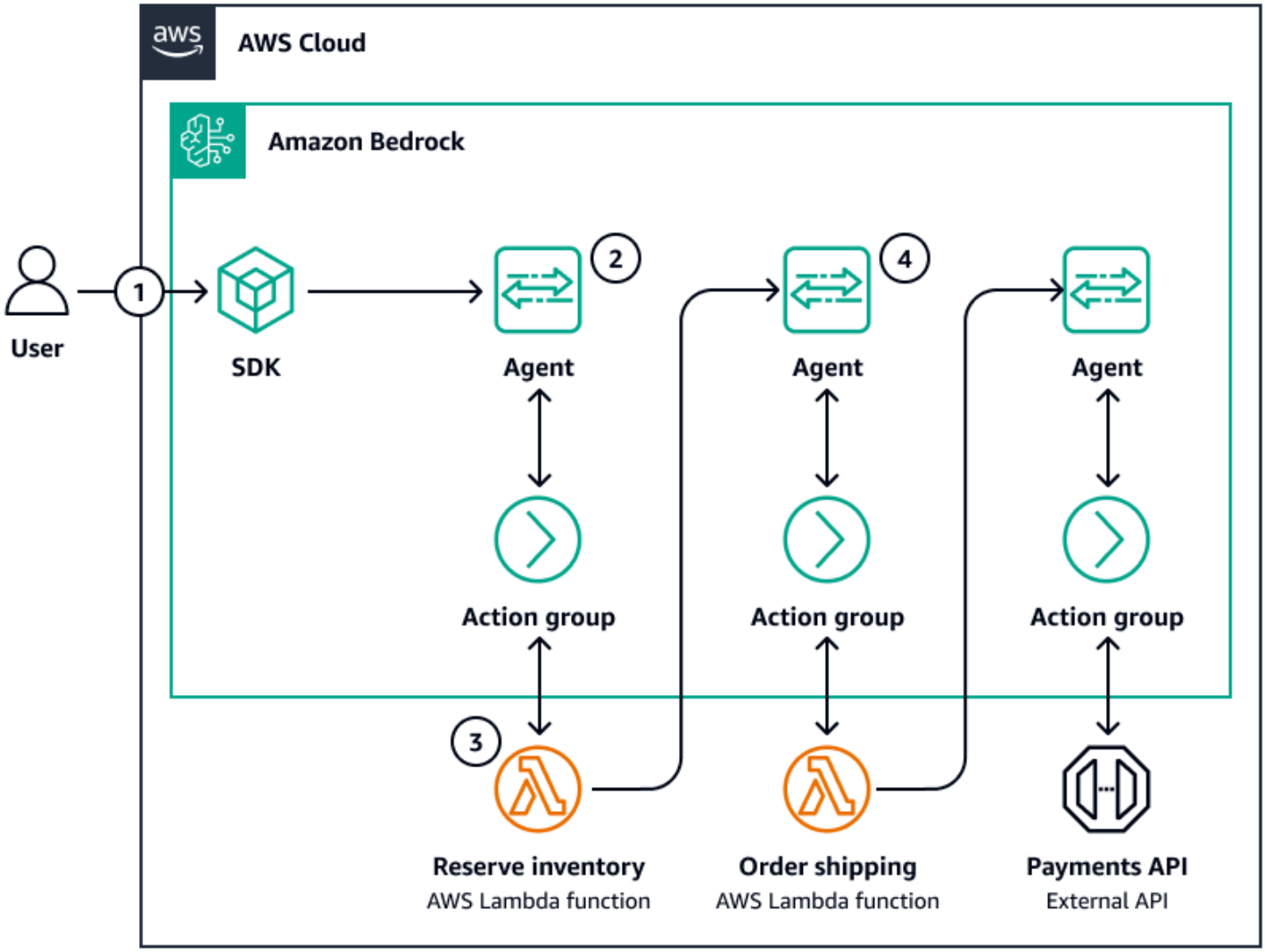
1. LLM interpreta una consulta de usuario compleja y genera una hipótesis
2. LLM elabora un plan para resolver la tarea
3. LLM ejecuta una subtarea (por ejemplo, mediante una llamada a una herramienta o recuperando conocimientos)
4. El LLM refina el resultado o retoma un paso anterior si considera que el resultado no es satisfactorio

Si un resultado intermedio es defectuoso, el sistema puede realizar una de las siguientes acciones:

- Vuelva a intentar los pasos con un enfoque diferente
- Vuelva a la solicitud anterior y vuelva a planificar
- Utilice un bucle de evaluación (por ejemplo, del patrón evaluador-optimizador) para detectar y corregir los errores

Al igual que el patrón de las sagas, el encadenamiento rápido permite avanzar parcialmente y retroceder. Esto ocurre mediante un refinamiento iterativo y una corrección dirigida por la LLM, en lugar de compensar las transacciones de la base de datos.

El siguiente diagrama es un ejemplo de coreografía de agentes:



1. Un usuario envía una consulta a través de un SDK.
2. Un agente de Amazon Bedrock organiza el razonamiento de la siguiente manera:
   - Interpretación (LLM)
   - Planificación (LLM)
   - Ejecución a través de una herramienta o base de conocimientos
   - Construcción de respuestas
3. Si una herramienta falla o no devuelve datos suficientes, el agente puede replanificar o reformular la tarea de forma dinámica.
4. La memoria (por ejemplo, un almacén vectorial a corto plazo) puede conservar su estado en todos los pasos

Conclusiones

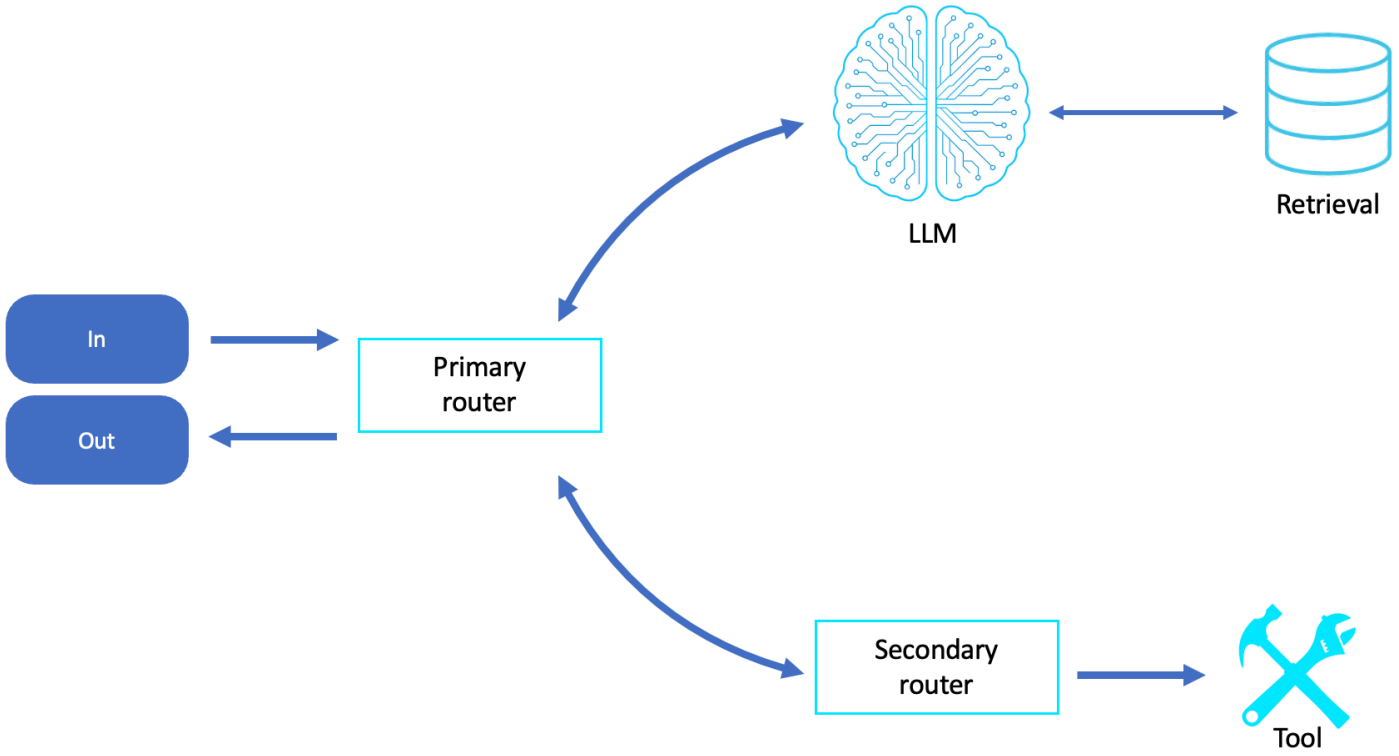
Mientras que el patrón saga gestiona las llamadas de servicio distribuidas con una lógica de compensación, el encadenamiento rápido gestiona las tareas de razonamiento mediante una secuenciación reflexiva y una replanificación adaptativa. Ambos sistemas permiten el progreso gradual, la descentralización de los puntos de decisión y la recuperación de los fallos, y todo ello mediante un razonamiento fundamentado en lugar de una reversión rígida.

El encadenamiento rápido introduce el razonamiento transaccional, que es el equivalente cognitivo de las sagas. Es decir, cada «pensamiento» es reevaluado, revisado o abandonado como parte de un diálogo más amplio dirigido a un objetivo.

Enrutamiento de patrones de despacho dinámicos

En los sistemas de agencia modernos, donde las tareas van desde el análisis de documentos hasta la generación autónoma de software, la capacidad de dirigir dinámicamente las solicitudes al agente o modelo de lenguaje grande (LLM) más competente se vuelve fundamental. La lógica de enrutamiento estática, que suele estar integrada en los scripts de orquestación o en las capas de API, carece de la adaptabilidad necesaria para los entornos en tiempo real, con varios modelos y múltiples capacidades. Para solucionar este problema, los flujos de trabajo de enrutamiento de LLM se pueden transformar en una arquitectura basada en eventos que aproveche un patrón de envío dinámico y convierta las llamadas de LLM en eventos enrutados de forma inteligente y sensibles al contexto.

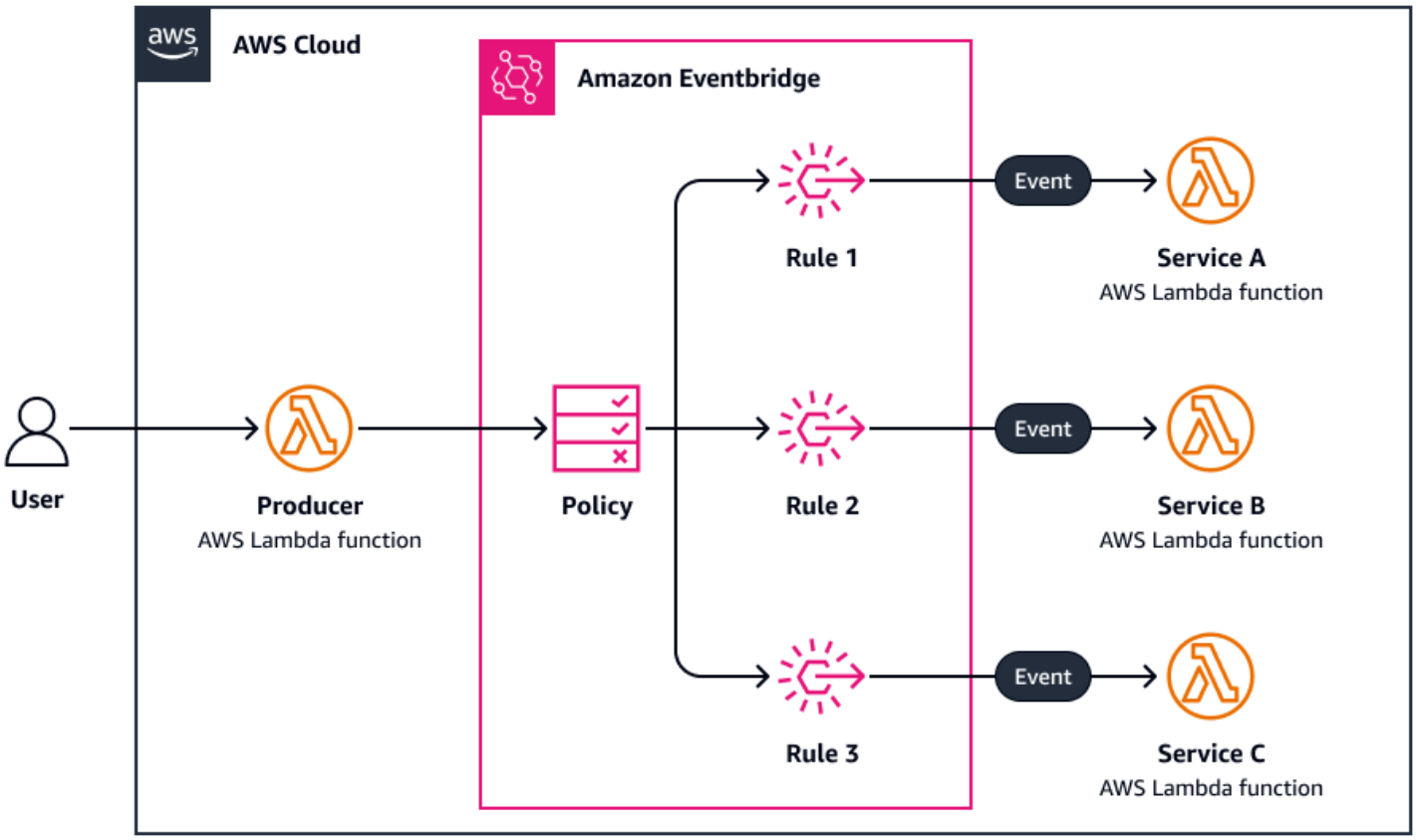
El siguiente diagrama es un ejemplo de enrutamiento LLM:



Envío dinámico

En los sistemas distribuidos tradicionales, el patrón de envío dinámico selecciona e invoca servicios específicos en tiempo de ejecución en función de los atributos del evento entrante, como el tipo de evento, la fuente y la carga útil. Por lo general, esto se implementa con Amazon EventBridge, que puede evaluar y dirigir los eventos entrantes a los objetivos adecuados (por ejemplo AWS Step Functions, AWS Lambda funciones o tareas de Amazon Elastic Container Service).

El siguiente diagrama es un ejemplo de envío dinámico:



1. Una aplicación emite un evento (por ejemplo, {"type»: «orderCreated», «priority»: «high"}).
2. Amazon EventBridge evalúa el evento según sus reglas de enrutamiento.
3. En función de los atributos de un evento, el sistema realiza los envíos de forma dinámica a lo siguiente:
   - `HighPriorityOrderProcessor`(servicio A)
   - `StandardOrderProcessor`(servicio B)
   - `UpdateOrderProcessor`(servicio C)

Este patrón admite el acoplamiento flexible, la especialización basada en el dominio y la extensibilidad del tiempo de ejecución. Esto permite que los sistemas respondan de forma inteligente a los cambios en los requisitos y a la semántica de los eventos.

Enrutamiento basado en LLM

En los sistemas de agencia, el enrutamiento también realiza una delegación dinámica de tareas, pero en lugar de EventBridge las reglas de Amazon o los filtros de metadatos, el LLM clasifica e interpreta

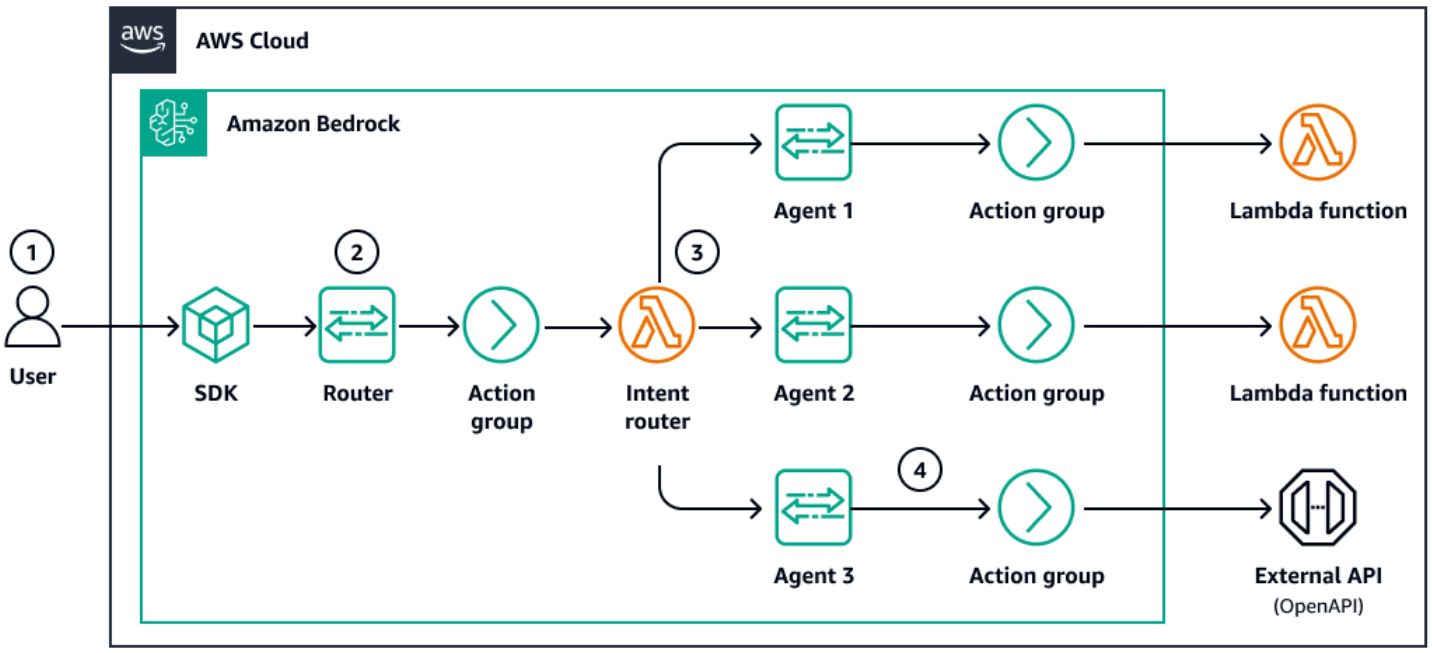
la intención del usuario a través de un lenguaje natural. El resultado es una forma de distribución flexible, semántica y adaptativa.

Agente (router)

Esta arquitectura permite un amplio envío basado en la intención sin esquemas o tipos de eventos predefinidos, lo que resulta ideal para entradas no estructuradas y consultas complejas.

1. Un usuario envía la siguiente solicitud: «¿Me pueden ayudar a revisar las condiciones de mi contrato?»
2. El LLM interpreta esto como una tarea de documentación legal.
3. El agente dirige la tarea a una o más de las siguientes opciones:
   - Plantilla de solicitud de revisión de contratos
   - Subagente de razonamiento legal
   - Herramienta de análisis de documentos

El siguiente diagrama es un ejemplo de un router de agente:



1. Un usuario envía una solicitud en lenguaje natural a través de un SDK.
2. Un agente de Amazon Bedrock utiliza un LLM para clasificar la tarea (por ejemplo, legal, técnica o de programación).

3. El agente dirige la tarea de forma dinámica a través de un grupo de acciones para invocar al agente requerido:
   - Agente de dominio específico
   - Cadena de herramientas especializada
   - Configuración rápida personalizada
4. El controlador seleccionado procesa la tarea y devuelve una respuesta personalizada.

Conclusiones

Mientras que el envío dinámico tradicional utiliza EventBridge las reglas de Amazon para el enrutamiento en función de los atributos de los eventos estructurados, el enrutamiento agencial se usa LLMs para clasificar y enrutar semánticamente las tareas en función del significado y la intención. Esto amplía la flexibilidad del sistema al permitir lo siguiente:

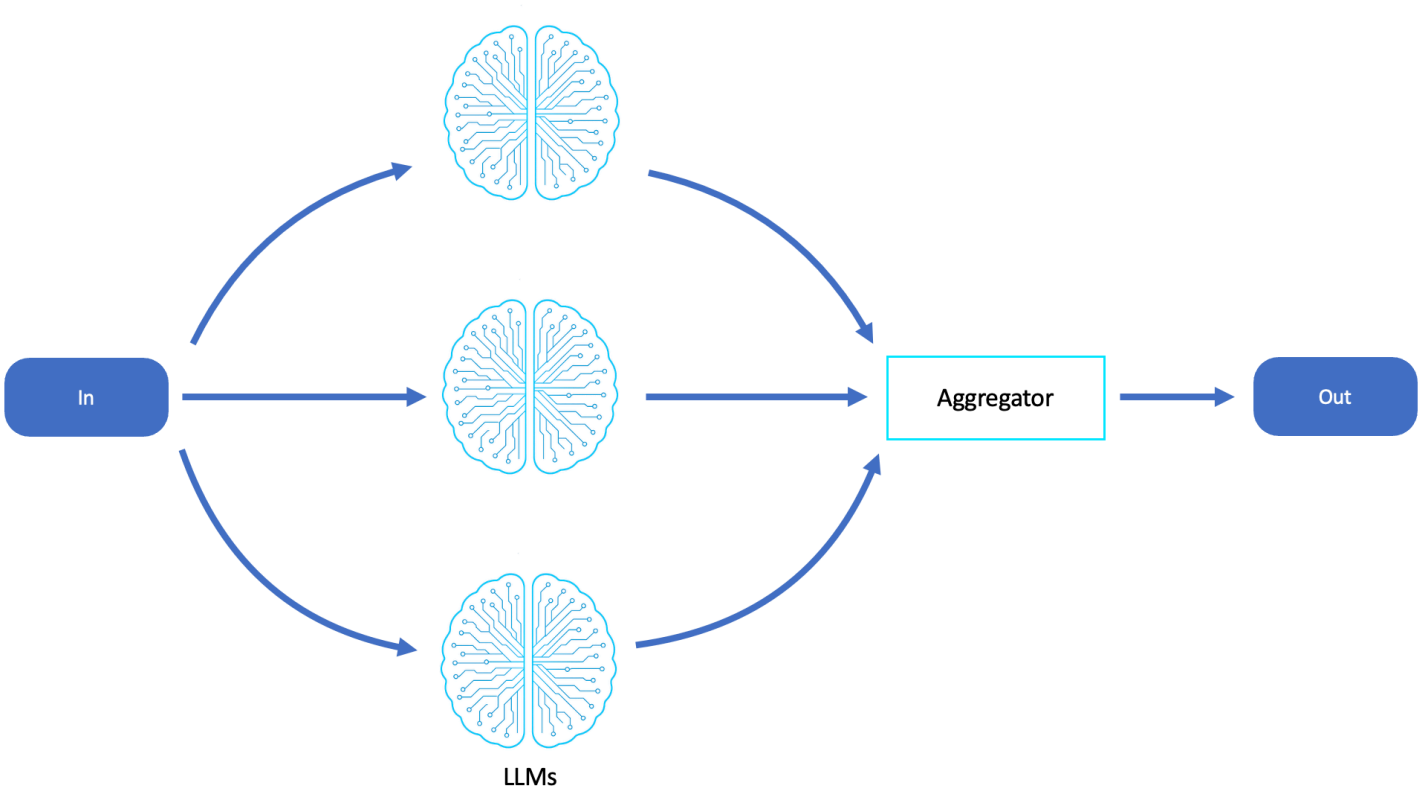
- Comprensión más amplia de las entradas
- Alternativa inteligente y selección de herramientas
- Extensibilidad natural a través de nuevos roles de agente o estilos de programación

El enrutamiento de los agentes reemplaza las reglas rígidas por una distribución cognitiva dinámica, que permite que los sistemas evolucionen con el lenguaje y no con el código.

Patrones de paralelización y dispersión

Muchas tareas avanzadas de razonamiento y generación, como resumir documentos de gran tamaño, evaluar múltiples rutas de solución o comparar diversas perspectivas, se benefician de la ejecución paralela de las solicitudes. Los flujos de trabajo secuenciales tradicionales son insuficientes cuando se requieren escalabilidad, capacidad de respuesta y tolerancia a errores. Para superar esta situación, la paralelización basada en la LLM puede reinventarse mediante un patrón de dispersión y recopilación basado en eventos, en el que las tareas se distribuyen dinámicamente entre agentes autónomos y los resultados se sintetizan de forma inteligente.

El siguiente diagrama es un ejemplo de un flujo de trabajo de paralelización de LLM:



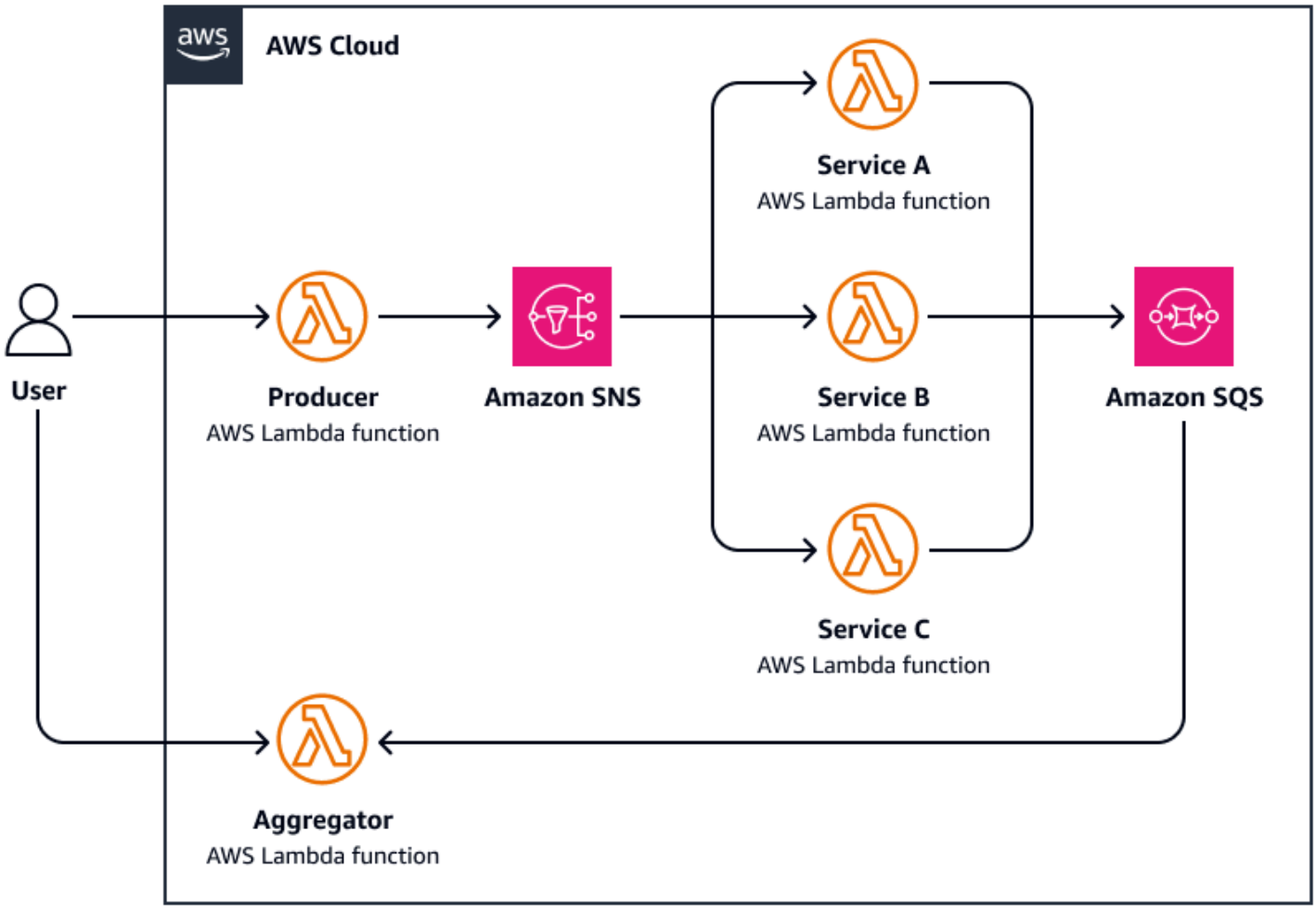
Dispersión y recopilación

En los sistemas distribuidos, un patrón de dispersión y recopilación envía las tareas a varios servicios o unidades de procesamiento en paralelo, espera sus respuestas y, a continuación, agrega los resultados en una salida consolidada. A diferencia de la dispersión, la dispersión y recolección se coordina porque espera respuestas y, por lo general, aplica la lógica para combinar, comparar y seleccionar los resultados.

Entre las implementaciones habituales de la paralelización y la recopilación de dispersión se incluyen las siguientes:

- AWS Step Functions mapear un estado para la ejecución de tareas en paralelo
- AWS Lambda con simultaneidad, coordinando los resultados de múltiples funciones invocadas
- Amazon EventBridge con flujos de trabajo de correlación IDs y agregación
- Patrón de controlador personalizado para gestionar la distribución y recopilar resultados mediante Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB o colas

El siguiente diagrama es un ejemplo de scatter-gather:



1. Un usuario envía una solicitud a una función de coordinación central que dispersa la tarea publicando mensajes paralelos en un tema del Amazon Simple Notification Service (Amazon SNS).
2. Cada mensaje incluye metadatos de la tarea y se envía a un trabajador especializado. AWS Lambda
3. Cada trabajador procesa de AWS Lambda forma independiente la subtarea asignada (por ejemplo, consulta una API externa, procesa un documento y analiza datos).
4. Los resultados se escriben en una capa de almacenamiento común, como Amazon Simple Queue Service (Amazon SQS).
5. La función de agregación espera a que se completen todas las respuestas y, a continuación, hace lo siguiente:
   - Recopila y agrega los resultados (por ejemplo, combina resúmenes y selecciona las mejores coincidencias)

- Envía una respuesta final o desencadena un flujo de trabajo posterior

Los casos de uso más comunes de los patrones de dispersión y recolección incluyen los siguientes:

- Búsqueda federada
- Motores de comparación de precios
- Análisis de datos agregados
- Inferencia multimodelo

Paralelización basada en LLM (cognición dispersa-recopilada)

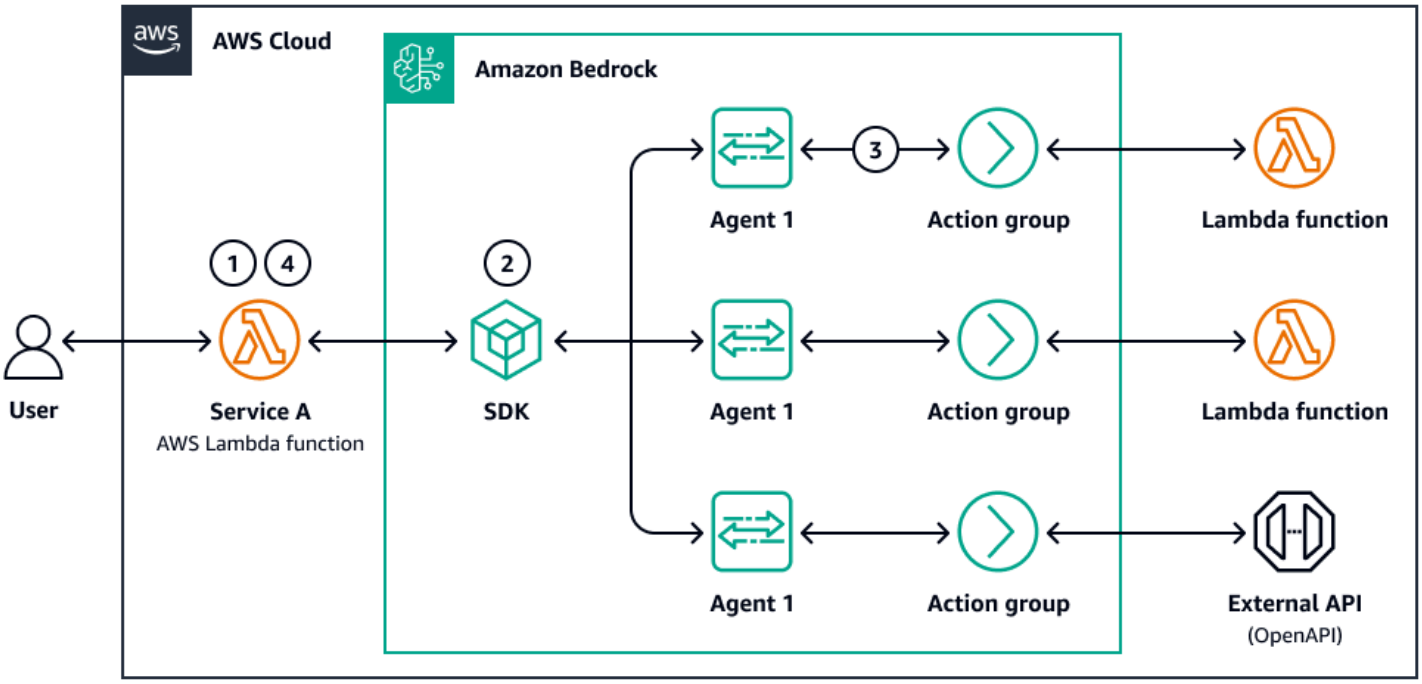
En los sistemas agenciales, la paralelización refleja de cerca la dispersión y la recopilación mediante la distribución de las subtareas entre varios agentes o llamadas de LLM, cada uno de los cuales analiza de forma independiente una parte del problema. Los resultados devueltos se recopilan y sintetizan mediante un proceso de agregación, que suele ser otro LLM o agente controlador.

Paralelización de agentes

1. Un agente envía una solicitud para «Resumir la información recopilada en estos 10 informes».
2. Distribuye los informes en 10 tareas de resumen LLM paralelas.
3. Cuando devuelve todos los resúmenes, el agente hace lo siguiente:
   - Reúne los resúmenes en un resumen unificado
   - Identifica temas o contradicciones
   - Envía la salida sintetizada al usuario

Este flujo de trabajo de agencia permite un razonamiento paralelo escalable, modular y adaptativo. Esto es ideal para casos de uso que requieren un alto rendimiento cognitivo.

El siguiente diagrama es un ejemplo de paralelización de agentes:



1. Un usuario envía una consulta o un conjunto de documentos de varias partes.
2. Un controlador AWS Lambda o una función escalonada distribuye las subtareas. Cada tarea invoca una llamada o un subagente de Amazon Bedrock LLM con su propio mensaje.
3. Cuando se completan las llamadas y las subtareas, los resultados se almacenan (por ejemplo, en Amazon S3 o en un almacén de memoria) y un paso de agregación fusiona, compara o filtra los resultados.
4. El sistema devuelve la respuesta final al usuario o al agente intermedio.

Este sistema tiene un circuito de razonamiento distribuido con trazabilidad, tolerancia a errores y lógica opcional de ponderación o selección de resultados.

Conclusiones

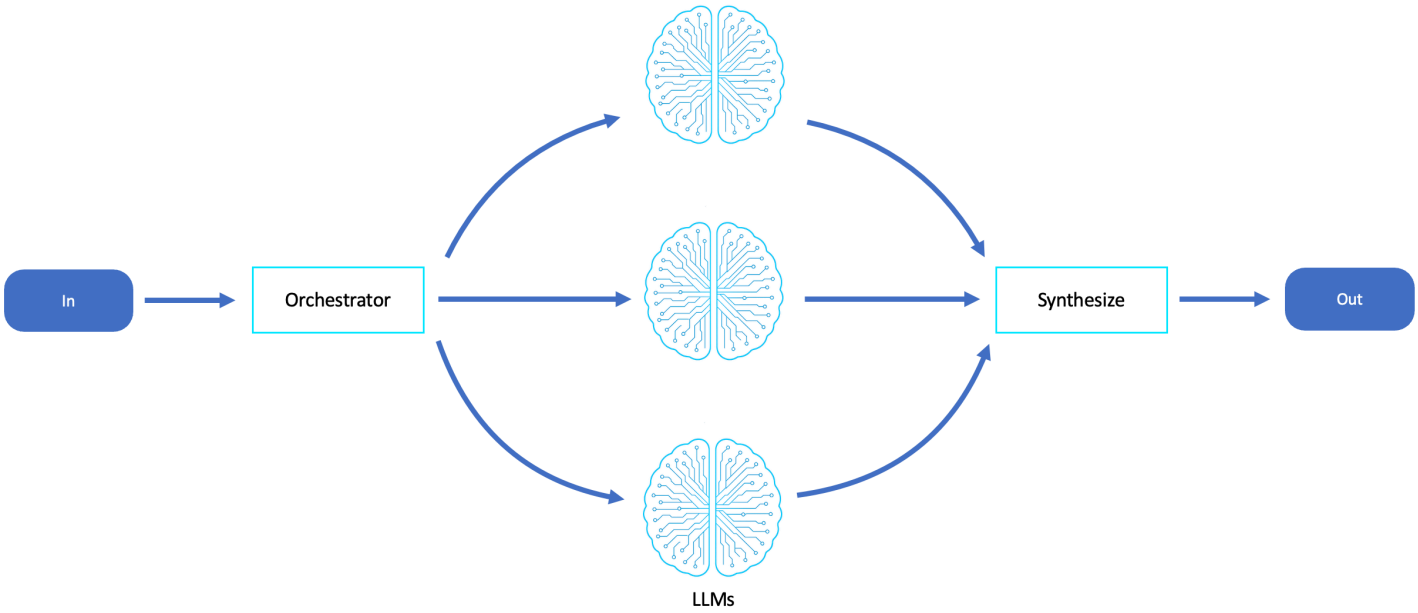
La paralelización de agentes utiliza patrones de dispersión y recolección para distribuir las tareas de LLM, lo que permite el procesamiento paralelo y la síntesis inteligente de resultados.

Patrones de orquestación de Saga

A medida que los flujos de trabajo impulsados por ellos LLMs se vuelven cada vez más complejos y abarcan cadenas de solicitudes, pasos de procesamiento de datos, invocaciones de herramientas

y colaboración entre agentes, la necesidad de una orquestación inteligente se vuelve esencial. En lugar de basarse en scripts estrechamente acoplados o flujos de ejecución estáticos predeterminados, estos flujos de trabajo se pueden implementar como patrones de orquestación basados en eventos, lo que permite a los sistemas basados en la LLM coordinar, monitorear y adaptar de forma dinámica las tareas de varios pasos entre agentes autónomos.

El siguiente diagrama es un ejemplo de un orquestador:



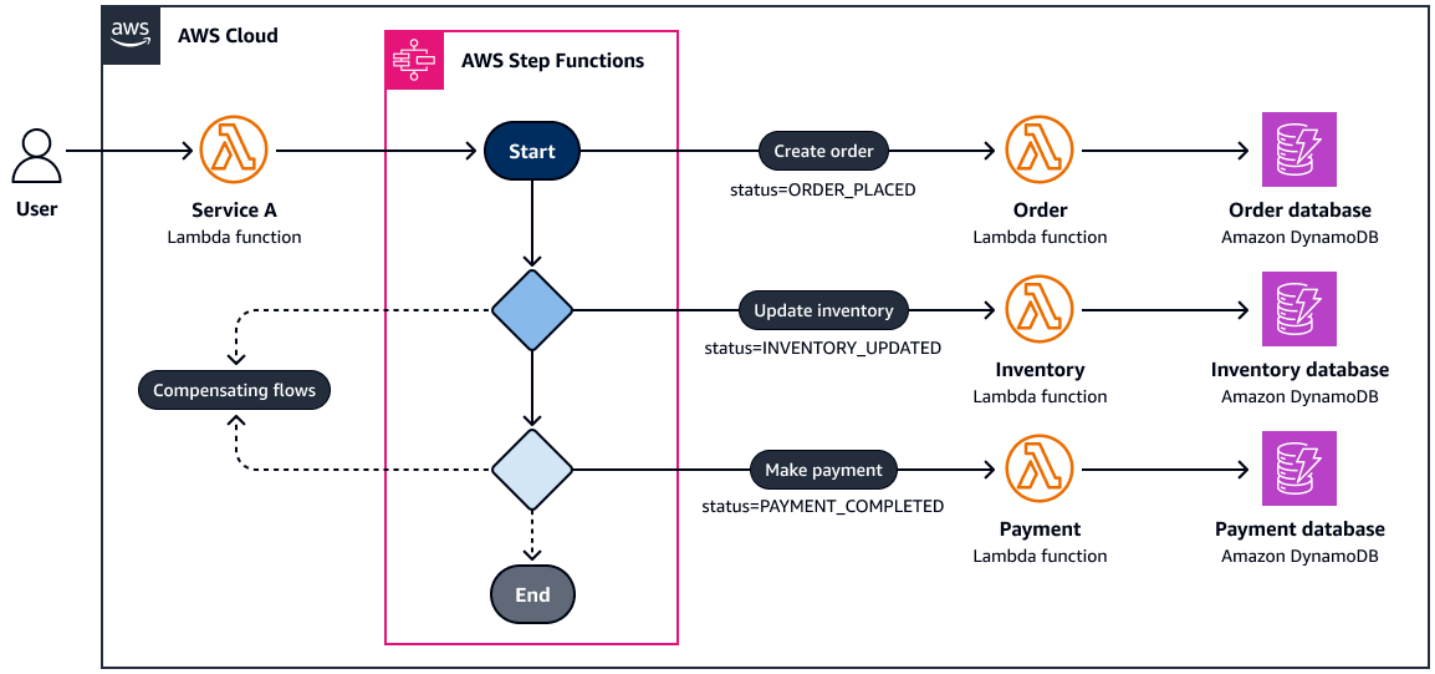
Orquestación de eventos

En los sistemas distribuidos tradicionales, la orquestación de eventos se refiere a un patrón en el que un coordinador central gestiona un flujo de trabajo complejo al dirigir explícitamente el flujo de control entre varios servicios o tareas. A diferencia de la coreografía de eventos (en la que cada servicio reacciona de forma independiente), la orquestación proporciona una lógica, visibilidad y control centralizados sobre todo el proceso.

Por lo general, esto se implementa con las siguientes herramientas:

- AWS Step Functions— Defina y ejecute flujos de trabajo con estado
- AWS Lambda— Realice tareas discretas dentro del flujo orquestado
- Amazon SQS o Amazon EventBridge: activa pasos o respuestas asíncronos

El siguiente diagrama es un ejemplo de la orquestación de una saga:



Un AWS Step Functions flujo de trabajo gestiona el proceso de pedido de un cliente:

1. Crear pedido (AWS Lambda)
2. Actualizar inventario (AWS Lambda)
3. Realizar el pago (AWS Lambda)

El orquestador coordina cada paso gestionando los reintentos, las ramificaciones paralelas, los tiempos de espera y los errores.

Sistema de agentes basado en roles (orquestador)

En los sistemas agenciales, el patrón del orquestador refleja la orquestación de eventos, pero distribuye la lógica entre varios agentes de razonamiento, cada uno con una función o especialización definida. Un agente orquestador central interpreta la tarea general, la descompone en subtareas y las delega en agentes trabajadores, cada una optimizada para un dominio concreto (por ejemplo, investigación, codificación, resumen o revisión).

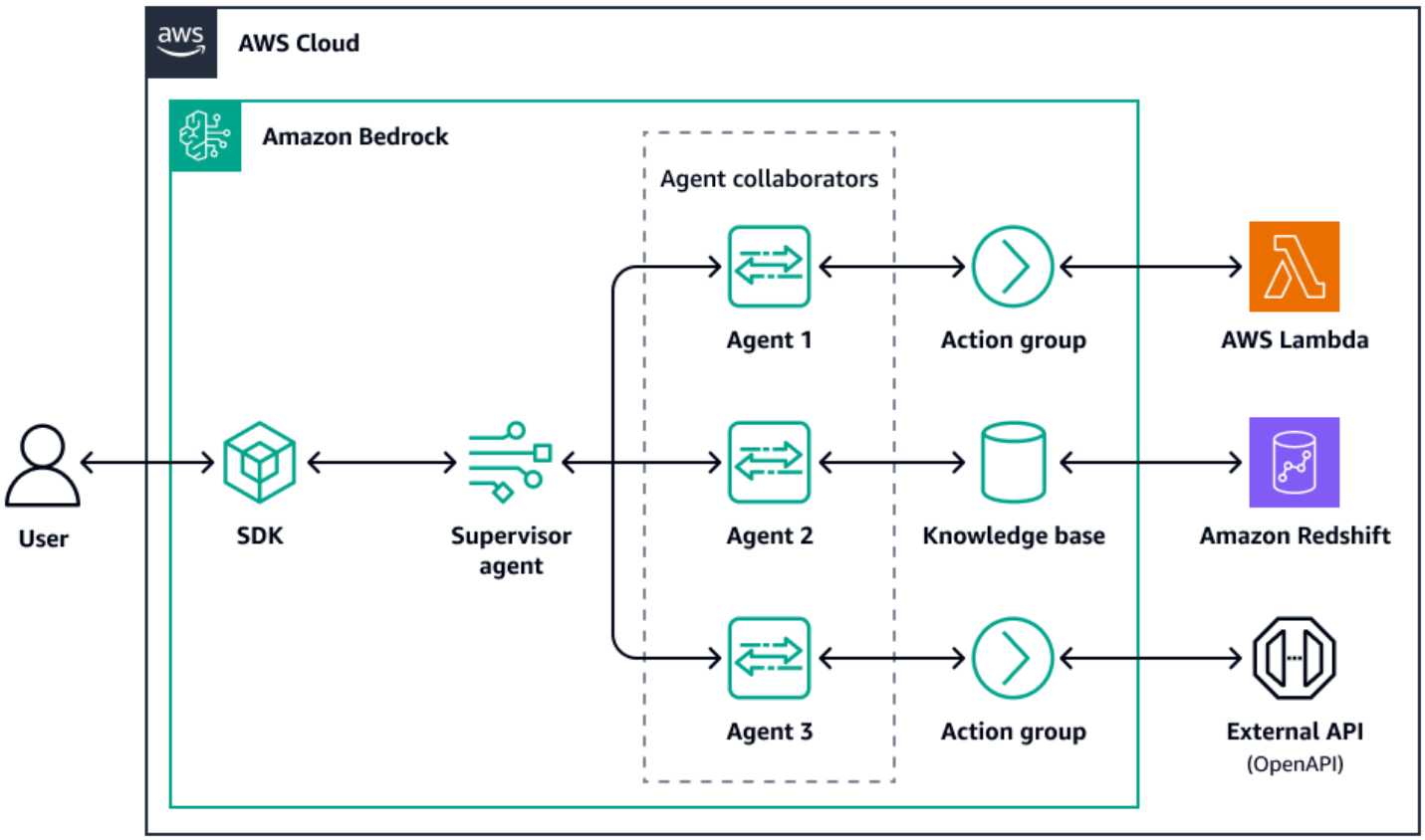
Supervisor

1. Un usuario envía la siguiente pregunta: «Cree un resumen del proyecto y resuma los cinco principales competidores».

2. El agente orquestador hace lo siguiente:
   - Asigna un agente de investigación para buscar datos de la competencia
   - Envía los resultados sin procesar a un agente de resumen
   - Pasa los resultados a un agente redactor de informes
   - Compila el resultado final para el usuario

Cada agente funciona de forma independiente, pero el orquestador coordina las tareas. Es como una función Lambda que gestiona las tareas del flujo de trabajo.

El siguiente diagrama es un ejemplo de un supervisor:



1. Un usuario envía una tarea a un agente supervisor de Amazon Bedrock.
2. El agente supervisor analiza la solicitud en subtareas para cada agente colaborador.
3. Cada subtarea se asigna a un agente colaborador con instrucciones o cadenas de herramientas específicas para cada función.
4. Los agentes de trabajo recurren a herramientas APIs o herramientas externas a través de un grupo de acción.

5. Cada agente de trabajo devuelve el resultado en un formato estructurado.
6. Cuando todos los trabajadores devuelven sus resultados, el supervisor evalúa, sintetiza y devuelve la respuesta final.

Esta estructura permite la modularidad, la adaptabilidad y la introspección en flujos de trabajo complejos de agentes de varios pasos.

Conclusiones

Mientras que la orquestación de eventos utiliza un control centralizado (por ejemplo AWS Step Functions) para dirigir la ejecución del servicio, los sistemas de agentes basados en roles utilizan un agente orquestador con tecnología LLM para razonar sobre el objetivo, delegar subtareas a los agentes trabajadores y sintetizar el resultado final.

En ambos paradigmas, el orquestador hace lo siguiente:

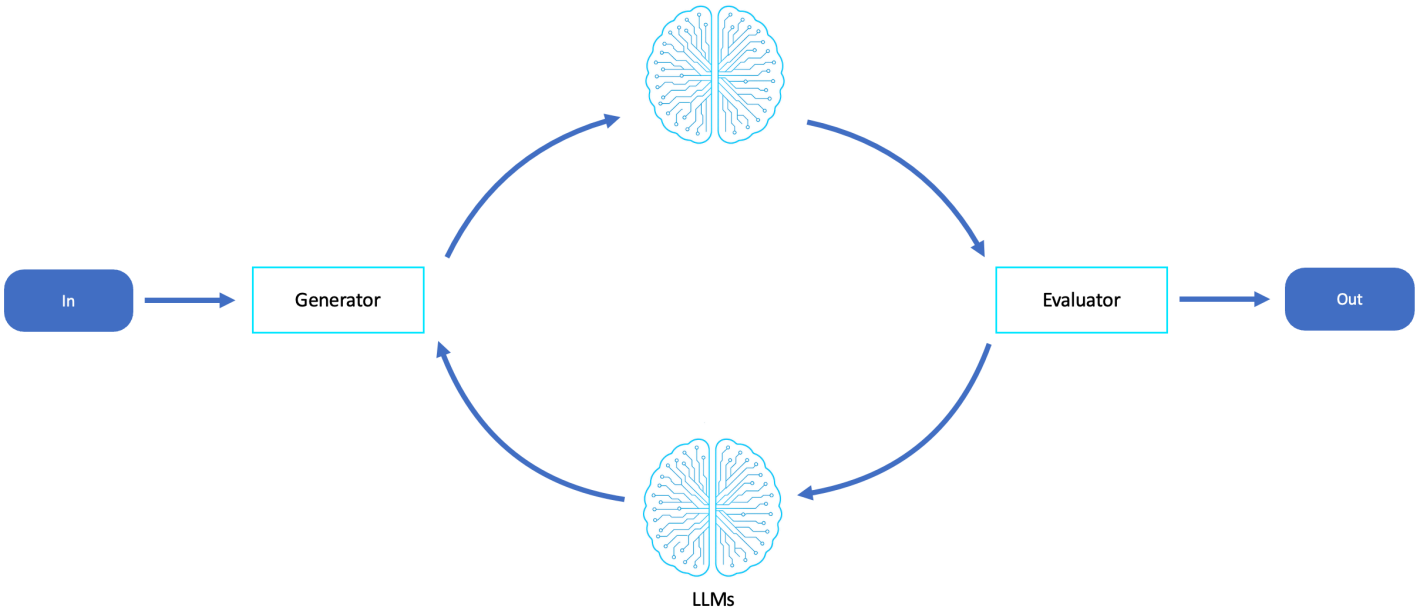
- Mantiene el contexto y el flujo de ejecución
- Gestiona la ramificación, la secuenciación y la gestión de errores
- Produce un resultado unificado a partir de componentes distribuidos

Sin embargo, la orquestación agencial añade razonamiento, adaptabilidad y delegación semántica. Esto lo hace ideal para tareas abiertas, ambiguas y en evolución.

El evaluador refleja y refina los patrones de bucles

Las tareas como la generación de código, el resumen o la toma de decisiones autónoma se benefician en gran medida de la retroalimentación en tiempo de ejecución, lo que permite que el sistema evolucione mediante la observación y el perfeccionamiento. Para ponerlo en práctica, el ciclo de reflexión y refinamiento se puede implementar como un circuito de control de retroalimentación basado en eventos, un patrón inspirado en la ingeniería de sistemas y adaptado a flujos de trabajo autónomos e inteligentes.

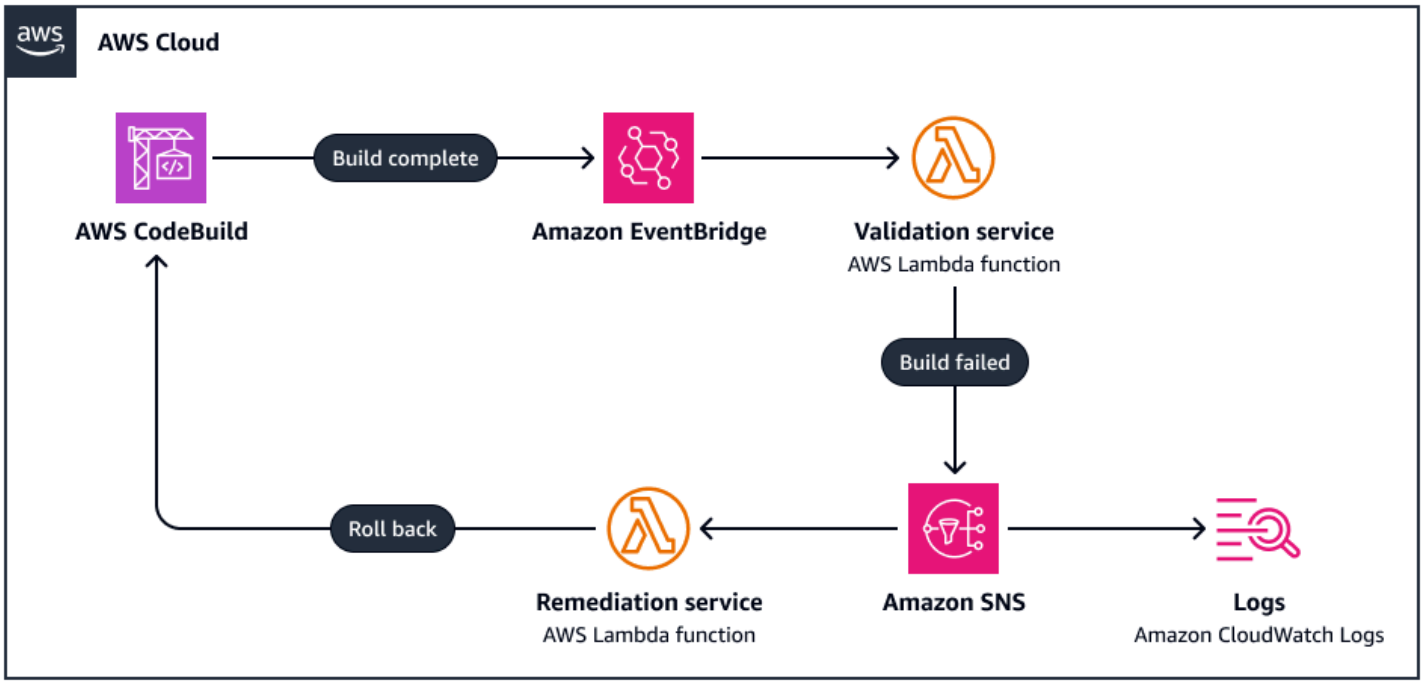
El siguiente diagrama es un ejemplo de un circuito de retroalimentación entre reflexión y refinamiento del evaluador:



Bucle de control de retroalimentación

Un circuito de control de retroalimentación es un patrón que monitorea sus propios resultados y comportamientos, los evalúa en función de criterios definidos o de un estado deseado y, a continuación, ajusta sus acciones en consecuencia. Esta arquitectura se inspira en la teoría del control y es fundamental en ámbitos como la automatización, los procesos de integración y entrega continuas (CI/CD) y las operaciones de aprendizaje automático.

El siguiente diagrama es un ejemplo de un circuito de control de retroalimentación:



1. Una canalización de despliegue emite un evento BuildComplete.
2. El evento desencadena un trabajo de prueba o evaluación automatizado que valida la compilación.
3. Si la validación falla (por ejemplo, debido a pruebas fallidas, problemas de seguridad o una infracción de una política), el sistema:
   - Emite un evento BuildComplete
   - Registra el problema o envía una notificación
   - Activa una acción correctiva o correctiva, como revertirla, parchearla o volver a intentarlo

El ciclo continúa hasta que produce un resultado o una escalada aceptables, o hasta que se agota el tiempo de espera. Este patrón se suele utilizar para lo siguiente:

- EventBridge Reglas de Amazon para dirigir los eventos a tareas de evaluación o corrección
- AWS Step Functions para la lógica de reintentos iterativos y la ramificación de los resultados de la evaluación
- Amazon Simple Notification Service (Amazon SNS) o alarmas de CloudWatch Amazon para activar comentarios y alertas
- AWS Lambda funciones o trabajadores agrupados en contenedores para aplicar medidas correctivas

Bucle de control de retroalimentación (evaluador)

El flujo de trabajo de un evaluador es un circuito de retroalimentación cognitiva impulsado por agentes LLMs de razonamiento. El proceso consiste en lo siguiente:

1. Un agente generador o LLM produce un resultado (por ejemplo, un plan, una respuesta o un borrador).
2. Un agente evaluador revisa el resultado utilizando una guía de crítica o una rúbrica de evaluación.
3. En función de los comentarios, el agente original o un nuevo agente optimizador revisan el resultado.

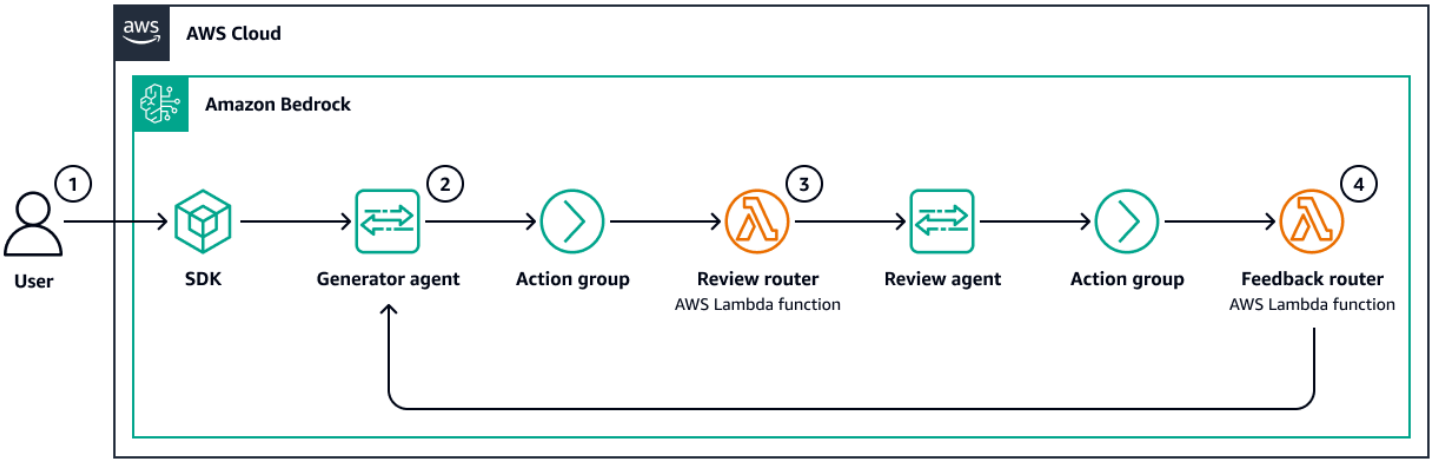
El ciclo se repite hasta que el resultado cumpla una serie de criterios, se apruebe o alcance un límite de reintentos.

Evaluador

1. Un usuario le pide a un agente que redacte un resumen de la política.
2. El agente generador lo redacta.
3. Un agente evaluador comprueba la cobertura, el tono y la corrección legal.
4. Si la respuesta es inadecuada, se refina y se vuelve a enviar hasta que converja el ciclo de retroalimentación.

Esto permite la autoevaluación, el refinamiento iterativo y el control adaptativo de los resultados, todo ello sin intervención humana.

El siguiente diagrama es un ejemplo de un circuito de control de retroalimentación (evaluador):



1. Un usuario emite una tarea (por ejemplo, redactar una estrategia empresarial).
2. Un agente de Amazon Bedrock genera un borrador inicial mediante un LLM.
3. Un segundo agente (o un agente de seguimiento) realiza una evaluación estructurada (por ejemplo, «califica este resultado según su claridad, integridad y tono»).
4. Si la calificación cae por debajo de un umbral, la respuesta se revisa de la siguiente manera:
   - Reinvocar el generador con una crítica incrustada
   - Enviar los comentarios a un agente refinador especializado
   - Iterando hasta alcanzar una respuesta aceptable

Los componentes opcionales, como AWS Lambda los controladores, AWS Step Functions pueden gestionar los umbrales de retroalimentación, los reintentos y las estrategias alternativas.

Conclusiones

Mientras que los bucles de control de retroalimentación tradicionales utilizan eventos, métricas y lógica de remediación para validar y ajustar el comportamiento del sistema, los bucles de evaluación de los agentes utilizan agentes de razonamiento para evaluar, reflejar y revisar los resultados de forma dinámica.

En ambos paradigmas:

- La salida se evalúa después de generarse
- Las acciones correctivas o de refinamiento se activan en función de los comentarios
- El sistema se adapta continuamente a un objetivo, calidad u objetivo

La versión agencial transforma la validación estática en reflexión semántica, lo que permite a los agentes que se mejoran a sí mismos evaluar su propia eficacia.

Diseñar los flujos de trabajo de los agentes en AWS

Cada patrón de esta guía se puede crear utilizando. Servicios de AWS Los agentes de Amazon Bedrock proporcionan canales de organización, acceso a datos e interacción.

Componente	Servicio de AWS	Finalidad
Razonamiento LLM	Amazon Bedrock	Lógica del agente, planificación, uso de herramientas
Ejecución de herramientas	AWS Lambda, Amazon ECS, Amazon SageMaker	Aloje herramientas externas para los agentes
Memoria y RAG	Base de conocimiento de Amazon Bedrock, Amazon S3, OpenSearch	Memoria persistente y semántica
Orquestación	AWS Step Functions	Coordinación de tareas y agentes en varios pasos
Enrutamiento de eventos	Amazon EventBridge, Amazon SQS	Mensajería interagente desacoplada
Interfaz de usuario	Amazon API Gateway AWS AppSync, SDK	Puntos de entrada para aplicaciones o sistemas
Supervisión	Amazon CloudWatch AWS X-Ray, AWS Distro para OpenTelemetry	Observabilidad e introspección de los agentes

Conclusión

Los patrones de flujo de trabajo de los agentes son la siguiente etapa evolutiva de las arquitecturas basadas en eventos, en las que la lógica empresarial no se define de forma estática, sino que se razona de forma dinámica mediante el uso de una cognición mejorada mediante un modelo de

lenguaje amplio (LLM). Al combinar las primitivas tradicionales nativas de la nube con los flujos de trabajo de la LLM y los patrones de diseño de los agentes, las organizaciones pueden crear sistemas adaptables, inteligentes y modulares que respondan con un propósito y aprendan de la experiencia.

En estos patrones, Amazon Bedrock es la puerta de entrada a la cognición de los agentes, ya que permite a los agentes basados en LLM acceder a los flujos de trabajo de los eventos, interactuar con las herramientas y la memoria y ofrecer resultados estructurados, rastreables y alineados.

Al diseñar e implementar sistemas de agencia, estos patrones de flujo de trabajo proporcionan modelos para crear arquitecturas de IA autónomas y componibles. Estos sistemas se basan en las mejores prácticas sin servidor y se complementan con modelos básicos inteligentes.

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
Publicación inicial	—	14 de julio de 2025

AWS Glosario de orientación prescriptiva

Los siguientes son términos de uso común en las estrategias, guías y patrones proporcionados por la Guía AWS prescriptiva. Para sugerir entradas, utilice el enlace Enviar comentarios al final del glosario.

Números

Las 7 R

Siete estrategias de migración comunes para trasladar aplicaciones a la nube. Estas estrategias se basan en las 5 R que Gartner identificó en 2011 y consisten en lo siguiente:

- Refactor/re-architect — Mueva una aplicación y modifique su arquitectura aprovechando al máximo las funciones nativas de la nube para mejorar la agilidad, el rendimiento y la escalabilidad. Por lo general, esto implica trasladar el sistema operativo y la base de datos. Ejemplo: migre su base de datos Oracle local a la PostgreSQL-Compatible edición Amazon Aurora.
- Redefinir la plataforma (transportar y redefinir): traslade una aplicación a la nube e introduzca algún nivel de optimización para aprovechar las capacidades de la nube. Ejemplo: Migrar la base de datos Oracle en las instalaciones a Amazon Relational Database Service (Amazon RDS) para Oracle en la nube de Nube de AWS.
- Recomprar (readquirir): cambie a un producto diferente, lo cual se suele llevar a cabo al pasar de una licencia tradicional a un modelo SaaS. Ejemplo: migre su sistema de gestión de relaciones con los clientes (CRM) a Salesforce.com.
- Volver a alojar (migrar mediante lift-and-shift): traslade una aplicación a la nube sin hacer cambios para aprovechar las funcionalidades de la nube. Ejemplo: Migrar la base de datos de Oracle en las instalaciones a Oracle en una instancia de EC2 en la Nube de AWS.
- Reubicar: (migrar el hipervisor mediante lift and shift): traslade la infraestructura a la nube sin comprar equipo nuevo, reescribir aplicaciones o modificar las operaciones actuales. Los servidores se migran de una plataforma en las instalaciones a un servicio en la nube para la misma plataforma. Ejemplo: migrar una Microsoft Hyper-V aplicación a AWS.
- Retener (revisitar): conserve las aplicaciones en el entorno de origen. Estas pueden incluir las aplicaciones que requieren una refactorización importante, que desee posponer para más adelante, y las aplicaciones heredadas que desee retener, ya que no hay ninguna justificación empresarial para migrarlas.

- Retirar: retire o elimine las aplicaciones que ya no sean necesarias en un entorno de origen.

A

A2A () Agent-to-Agent

Un protocolo completo para la colaboración entre agentes que facilita la delegación de tareas y la transferencia de estados.

ABAC

Consulte control de acceso basado en atributos.

servicios abstractos

Consulte servicios administrados.

ACID

Consulte atomicidad, consistencia, aislamiento, durabilidad.

migración activa-activa

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas (mediante una herramienta de replicación bidireccional o mediante operaciones de escritura doble) y ambas bases de datos gestionan las transacciones de las aplicaciones conectadas durante la migración. Este método permite la migración en lotes pequeños y controlados, en lugar de requerir una transición única. Es más flexible, pero requiere más trabajo que una migración activa-pasiva.

migración activa-pasiva

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas, pero solo la de origen gestiona las transacciones de las aplicaciones conectadas, mientras los datos se replican en la de destino. La base de datos de destino no acepta ninguna transacción durante la migración.

Agente

Un sistema de IA que puede razonar, planificar y tomar medidas de forma autónoma utilizando herramientas para alcanzar los objetivos.

Agent Ops

Prácticas operativas para crear, probar, implementar y ejecutar agentes de IA en producción a escala.

función de agregación

Función SQL que actúa en un grupo de filas y calcula un único valor de devolución para el grupo. Entre los ejemplos de funciones de agregación se incluyen `SUM` y `MAX`.

IA

Consulte inteligencia artificial.

AIOps

Consulte operaciones de inteligencia artificial

anonimización

El proceso de eliminar permanentemente la información personal de un conjunto de datos. La anonimización puede ayudar a proteger la privacidad personal. Los datos anonimizados ya no se consideran datos personales.

antipatrones

Una solución que se utiliza con frecuencia para un problema recurrente en el que la solución es contraproducente, ineficaz o menos eficaz que una alternativa.

control de aplicaciones

Enfoque de seguridad que permite usar de manera exclusiva aplicaciones aprobadas para ayudar a proteger un sistema contra el malware.

cartera de aplicaciones

Recopilación de información detallada sobre cada aplicación que utiliza una organización, incluido el costo de creación y mantenimiento de la aplicación y su valor empresarial. Esta información es clave para el proceso de detección y análisis de la cartera y ayuda a identificar y priorizar las aplicaciones que se van a migrar, modernizar y optimizar.

inteligencia artificial (IA)

El campo de la informática que se dedica al uso de tecnologías informáticas para realizar funciones cognitivas que suelen estar asociadas a los seres humanos, como el aprendizaje, la resolución de problemas y el reconocimiento de patrones. Para más información, consulte ¿Qué es la inteligencia artificial?

operaciones de inteligencia artificial (AIOps)

El proceso de utilizar técnicas de machine learning para resolver problemas operativos, reducir los incidentes operativos y la intervención humana, y mejorar la calidad del servicio. Para obtener más información sobre cómo se utiliza AIOps en la estrategia de migración de AWS , consulte la Guía de integración de operaciones.

cifrado asimétrico

Algoritmo de cifrado que utiliza un par de claves, una clave pública para el cifrado y una clave privada para el descifrado. Puede compartir la clave pública porque no se utiliza para el descifrado, pero el acceso a la clave privada debe estar sumamente restringido.

atomicidad, consistencia, aislamiento, durabilidad (ACID)

Conjunto de propiedades de software que garantizan la validez de los datos y la fiabilidad operativa de una base de datos, incluso en caso de errores, cortes de energía u otros problemas.

control de acceso basado en atributos (ABAC)

La práctica de crear permisos detallados basados en los atributos del usuario, como el departamento, el puesto de trabajo y el nombre del equipo. Para obtener más información, consulte ABAC AWS en la documentación AWS Identity and Access Management (IAM).

origen de datos fidedigno

Ubicación en la que se almacena la versión principal de los datos, que se considera la fuente de información más fiable. Puede copiar los datos del origen de datos autorizado a otras ubicaciones con el fin de procesarlos o modificarlos, por ejemplo, anonimizarlos, redactarlos o seudonimizarlos.

Zona de disponibilidad

Una ubicación distinta dentro de una Región de AWS que está aislada de los fallos en otras zonas de disponibilidad y que proporciona una conectividad de red económica y de baja latencia a otras zonas de disponibilidad de la misma región.

AWS Marco de adopción de la nube (AWS CAF)

Un marco de directrices y mejores prácticas AWS para ayudar a las organizaciones a desarrollar un plan eficiente y eficaz para migrar con éxito a la nube. AWS CAF organiza la orientación en seis áreas de enfoque denominadas perspectivas: negocios, personas, gobierno, plataforma, seguridad y operaciones. Las perspectivas empresariales, humanas y de gobernanza se centran en las habilidades y los procesos empresariales; las perspectivas de plataforma, seguridad y

operaciones se centran en las habilidades y los procesos técnicos. Por ejemplo, la perspectiva humana se dirige a las partes interesadas que se ocupan de los Recursos Humanos (RR. HH.), las funciones del personal y la administración de las personas. Desde esta perspectiva, AWS CAF proporciona orientación para el desarrollo, la formación y la comunicación de las personas a fin de preparar a la organización para una adopción exitosa de la nube. Para obtener más información, consulte la Página web de AWS CAF y el Documento técnico de AWS CAF.

AWS Marco de calificación de la carga de trabajo (AWS WQF)

Herramienta que evalúa las cargas de trabajo de migración de bases de datos, recomienda estrategias de migración y proporciona estimaciones de trabajo. AWS WQF se incluye con AWS Schema Conversion Tool ().AWS SCT Analiza los esquemas de bases de datos y los objetos de código, el código de las aplicaciones, las dependencias y las características de rendimiento y proporciona informes de evaluación.

B

bot malicioso

Bot destinado a causar interrupciones o daños a personas u organizaciones.

BCP

Consulte planificación de la continuidad del negocio.

gráfico de comportamiento

Una vista unificada e interactiva del comportamiento de los recursos y de las interacciones a lo largo del tiempo. Puede utilizar un gráfico de comportamiento con Amazon Detective para examinar los intentos de inicio de sesión fallidos, las llamadas sospechosas a la API y acciones similares. Para obtener más información, consulte Datos en un gráfico de comportamiento en la documentación de Detective.

sistema big-endian

Un sistema que almacena primero el byte más significativo. Consulte también endianidad.

clasificación binaria

Un proceso que predice un resultado binario (una de las dos clases posibles). Por ejemplo, es posible que su modelo de ML necesite predecir problemas como “¿Este correo electrónico es spam o no es spam?” o “¿Este producto es un libro o un automóvil?”.

filtro de floración

Estructura de datos probabilística y eficiente en términos de memoria que se utiliza para comprobar si un elemento es miembro de un conjunto.

blue/green despliegue

Estrategia de implementación en la que se crean dos entornos separados, pero idénticos. La versión actual de la aplicación se ejecuta en un entorno (azul) y la nueva versión de la aplicación se ejecuta en el otro entorno (verde). Esta estrategia lo ayuda a hacer reversiones rápidas con un impacto mínimo.

bot

Aplicación de software que ejecuta tareas automatizadas a través de Internet y simula la actividad o interacción humana. Algunos bots son útiles o beneficiosos, como los rastreadores web que indexan la información de Internet. Otros bots, conocidos como bots maliciosos, tienen como objetivo causar interrupciones o daños a personas u organizaciones.

botnet

Redes de bots infectadas por malware y que están bajo el control de una sola parte, conocida como pastor de bots u operador de bots. Las botnets son el mecanismo más conocido para escalar los bots y su impacto.

branch

Área contenida de un repositorio de código. La primera rama que se crea en un repositorio es la rama principal. Puede crear una rama nueva a partir de una rama existente y, a continuación, desarrollar características o corregir errores en la rama nueva. Una rama que se genera para crear una característica se denomina comúnmente rama de característica. Cuando la característica se encuentra lista para su lanzamiento, se vuelve a combinar la rama de característica con la rama principal. Para obtener más información, consulte Acerca de las sucursales (GitHub documentación).

acceso de emergencia

En circunstancias excepcionales y mediante un proceso aprobado, es una forma rápida de que un usuario pueda acceder a un Cuenta de AWS sitio al que normalmente no tiene permisos de acceso. Para obtener más información, consulte el indicador de implementación de procedimientos rompe-cristales en la AWS Well-Architected guía.

estrategia de implementación sobre infraestructura existente

La infraestructura existente en su entorno. Al adoptar una estrategia de implementación sobre infraestructura existente para una arquitectura de sistemas, se diseña la arquitectura en función de las limitaciones de los sistemas y la infraestructura actuales. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

caché de búfer

El área de memoria donde se almacenan los datos a los que se accede con más frecuencia.

capacidad empresarial

Lo que hace una empresa para generar valor (por ejemplo, ventas, servicio al cliente o marketing). Las arquitecturas de microservicios y las decisiones de desarrollo pueden estar impulsadas por las capacidades empresariales. Para obtener más información, consulte la sección Organizado en torno a las capacidades empresariales del documento técnico Ejecutar microservicios en contenedores en AWS.

planificación de la continuidad del negocio (BCP)

Plan que aborda el posible impacto de un evento disruptivo, como una migración a gran escala en las operaciones y permite a la empresa reanudar las operaciones rápidamente.

C

CAF

Consulte AWS Cloud Adoption Framework.

implementación canario

Lanzamiento lento e incremental de una versión para los usuarios finales. Cuando tenga mayor confianza en la nueva versión, la implementa y reemplaza la versión actual en su totalidad.

CCoE

Consulte Centro de excelencia en la nube.

CDC

Consulte captura de datos de cambios.

captura de datos de cambio (CDC)

Proceso de seguimiento de los cambios en un origen de datos, como una tabla de base de datos, y registro de los metadatos relacionados con el cambio. Puede utilizar los CDC para diversos fines, como auditar o replicar los cambios en un sistema de destino para mantener la sincronización.

ingeniería del caos

Introducción intencionada de fallos o eventos disruptivos para poner a prueba la resiliencia de un sistema. Puedes usar AWS Fault Injection Service (AWS FIS) para realizar experimentos que estresen tus AWS cargas de trabajo y evalúen su respuesta.

CI/CD

Consulte integración continua y entrega continua.

clasificación

Un proceso de categorización que permite generar predicciones. Los modelos de ML para problemas de clasificación predicen un valor discreto. Los valores discretos siempre son distintos entre sí. Por ejemplo, es posible que un modelo necesite evaluar si hay o no un automóvil en una imagen.

Desarrollador ciudadano

Un usuario empresarial que crea aplicaciones de IA utilizando plataformas sin code/low código sin conocimientos técnicos especializados.

cifrado del cliente

Cifrado de datos localmente, antes de que el objetivo los Servicio de AWS reciba.

Centro de excelencia en la nube (CCoE)

Equipo multidisciplinario que impulsa los esfuerzos de adopción de la nube en toda la organización, incluido el desarrollo de las prácticas recomendadas en la nube, la movilización de recursos, el establecimiento de plazos de migración y la dirección de la organización durante las transformaciones a gran escala. Para obtener más información, consulte las publicaciones de CCoE en el blog de estrategia Nube de AWS empresarial.

computación en la nube

La tecnología en la nube que se utiliza normalmente para la administración de dispositivos de IoT y el almacenamiento de datos de forma remota. La computación en la nube suele estar relacionada con la tecnología de computación de periferia.

modelo operativo en la nube

En una organización de TI, el modelo operativo que se utiliza para crear, madurar y optimizar uno o más entornos de nube. Para obtener más información, consulte Creación de su modelo operativo de nube.

etapas de adopción de la nube

Las siguientes son las cuatro fases por las que suelen pasar las empresas cuando migran a la Nube de AWS:

- Proyecto: ejecución de algunos proyectos relacionados con la nube con fines de prueba de concepto y aprendizaje
- Fundamento: realización de inversiones fundamentales para escalar la adopción de la nube (p. ej., crear una zona de aterrizaje, definir un CCoE, establecer un modelo de operaciones)
- Migración: migración de aplicaciones individuales
- Re-invention — Optimizar los productos y servicios e innovar en la nube

Stephen Orban definió estas etapas en la entrada del blog The Journey Toward Cloud-First & the Stages of Adoption del blog Nube de AWS Enterprise Strategy. Para obtener información sobre su relación con la estrategia de AWS migración, consulte la guía de preparación para la migración.

CMDB

Consulte base de datos de administración de configuración.

repositorio de código

Una ubicación donde el código fuente y otros activos, como documentación, muestras y scripts, se almacenan y actualizan mediante procesos de control de versiones. Algunos repositorios en la nube comunes son GitHub o Bitbucket Cloud. Cada versión del código se denomina rama. En una estructura de microservicios, cada repositorio se encuentra dedicado a una única funcionalidad. Una sola CI/CD canalización puede utilizar varios repositorios.

caché en frío

Una caché de búfer que está vacía no está bien poblada o contiene datos obsoletos o irrelevantes. Esto afecta al rendimiento, ya que la instancia de la base de datos debe leer desde la memoria principal o el disco, lo que es más lento que leer desde la memoria caché del búfer.

datos fríos

Datos a los que se accede con poca frecuencia y que suelen ser históricos. Al consultar este tipo de datos, normalmente se aceptan consultas lentas. Trasladar estos datos a niveles o clases de almacenamiento de menor rendimiento y menos costosos puede reducir los costos.

visión artificial (CV)

Campo de la IA que utiliza el machine learning para analizar y extraer información de formatos visuales, como imágenes y videos digitales. Por ejemplo, Amazon SageMaker AI proporciona algoritmos de procesamiento de imágenes para CV.

deriva de configuración

En el caso de una carga de trabajo, un cambio en la configuración con respecto al estado esperado. Podría provocar que la carga de trabajo deje de cumplir las normas y, por lo general, es gradual e involuntaria.

base de datos de administración de configuración (CMDB)

Repositorio que almacena y administra información sobre una base de datos y su entorno de TI, incluidos los componentes de hardware y software y sus configuraciones. Por lo general, los datos de una CMDB se utilizan en la etapa de detección y análisis de la cartera de productos durante la migración.

paquete de conformidad

Un conjunto de AWS Config reglas y medidas correctivas que puede reunir para personalizar sus controles de conformidad y seguridad. Puede implementar un paquete de conformidad como una entidad única en una región Cuenta de AWS y, o en una organización, mediante una plantilla YAML. Para obtener más información, consulta los paquetes de conformidad en la documentación. AWS Config

integración y entrega continuas () CI/CD

El proceso de automatización de las etapas de origen, creación, prueba, puesta en escena y producción del proceso de publicación del software. CI/CD se describe comúnmente como una canalización. CI/CD puede ayudarlo a automatizar los procesos, mejorar la productividad, mejorar

la calidad del código y entregar más rápido. Para obtener más información, consulte Beneficios de la entrega continua. CD también puede significar implementación continua. Para obtener más información, consulte Entrega continua frente a implementación continua.

CV

Consulte visión artificial.

D

datos en reposo

Datos que están estacionarios en la red, como los datos que se encuentran almacenados.

clasificación de datos

Un proceso para identificar y clasificar los datos de su red en función de su importancia y sensibilidad. Es un componente fundamental de cualquier estrategia de administración de riesgos de ciberseguridad porque lo ayuda a determinar los controles de protección y retención adecuados para los datos. La clasificación de los datos es un componente del pilar de seguridad del AWS Well-Architected Framework. Para obtener más información, consulte Clasificación de datos.

deriva de datos

Una variación significativa entre los datos de producción y los datos que se utilizaron para entrenar un modelo de machine learning, o un cambio significativo en los datos de entrada a lo largo del tiempo. La deriva de datos puede reducir la calidad, la precisión y la imparcialidad generales de las predicciones de los modelos de machine learning.

datos en tránsito

Datos que se mueven de forma activa por la red, por ejemplo, entre los recursos de la red.

malla de datos

Marco de arquitectura que proporciona una propiedad de datos distribuida y descentralizada con una administración y una gobernanza centralizadas.

minimización de datos

El principio de recopilar y procesar solo los datos estrictamente necesarios. Practicar la minimización de los datos Nube de AWS puede reducir los riesgos de privacidad, los costos y la huella de carbono de la analítica.

perímetro de datos

Un conjunto de barreras preventivas en su AWS entorno que ayudan a garantizar que solo las identidades confiables accedan a los recursos confiables desde las redes esperadas. Para obtener más información, consulte Crear un perímetro de datos sobre. AWS

preprocesamiento de datos

Transformar los datos sin procesar en un formato que su modelo de ML pueda analizar fácilmente. El preprocesamiento de datos puede implicar eliminar determinadas columnas o filas y corregir los valores faltantes, incoherentes o duplicados.

procedencia de los datos

El proceso de rastrear el origen y el historial de los datos a lo largo de su ciclo de vida, por ejemplo, la forma en que se generaron, transmitieron y almacenaron los datos.

titular de los datos

Persona cuyos datos se recopilan y procesan.

almacenamiento de datos

Sistema de administración de datos que respalda la inteligencia empresarial, como los análisis. Los almacenes de datos suelen contener grandes cantidades de datos históricos y, por lo general, se utilizan para las consultas y los análisis.

lenguaje de definición de datos (DDL)

Instrucciones o comandos para crear o modificar la estructura de tablas y objetos de una base de datos.

lenguaje de manipulación de datos (DML)

Instrucciones o comandos para modificar (insertar, actualizar y eliminar) la información de una base de datos.

DDL

Consulte lenguaje de definición de bases de datos.

conjunto profundo

Combinar varios modelos de aprendizaje profundo para la predicción. Puede utilizar conjuntos profundos para obtener una predicción más precisa o para estimar la incertidumbre de las predicciones.

aprendizaje profundo

Un subcampo del ML que utiliza múltiples capas de redes neuronales artificiales para identificar el mapeo entre los datos de entrada y las variables objetivo de interés.

defensa en profundidad

Un enfoque de seguridad de la información en el que se distribuyen cuidadosamente una serie de mecanismos y controles de seguridad en una red informática para proteger la confidencialidad, la integridad y la disponibilidad de la red y de los datos que contiene. Al adoptar esta estrategia AWS, se añaden varios controles en diferentes capas de la AWS Organizations estructura para ayudar a proteger los recursos. Por ejemplo, un enfoque de defensa en profundidad podría combinar la autenticación multifactor, la segmentación de la red y el cifrado.

administrador delegado

En AWS Organizations, un servicio compatible puede registrar una cuenta de AWS miembro para administrar las cuentas de la organización y gestionar los permisos de ese servicio. Esta cuenta se denomina administrador delegado para ese servicio. Para obtener más información y una lista de servicios compatibles, consulte Servicios que funcionan con AWS Organizations en la documentación de AWS Organizations .

Implementación

El proceso de hacer que una aplicación, características nuevas o correcciones de código se encuentren disponibles en el entorno de destino. La implementación abarca implementar cambios en una base de código y, a continuación, crear y ejecutar esa base en los entornos de la aplicación.

entorno de desarrollo

Consulte entorno.

control de detección

Un control de seguridad que se ha diseñado para detectar, registrar y alertar después de que se produzca un evento. Estos controles son una segunda línea de defensa, ya que lo advierten sobre los eventos de seguridad que han eludido los controles preventivos establecidos. Para obtener más información, consulte Controles de detección en Implementación de controles de seguridad en AWS.

asignación de flujos de valor para el desarrollo (DVSM)

Proceso que se utiliza para identificar y priorizar las restricciones que afectan negativamente a la velocidad y la calidad en el ciclo de vida del desarrollo de software. DVSM amplía el proceso de asignación del flujo de valor diseñado originalmente para las prácticas de fabricación ajustada. Se centra en los pasos y los equipos necesarios para crear y transferir valor a través del proceso de desarrollo de software.

gemelo digital

Representación virtual de un sistema del mundo real, como un edificio, una fábrica, un equipo industrial o una línea de producción. Los gemelos digitales son compatibles con el mantenimiento predictivo, la supervisión remota y la optimización de la producción.

tabla de dimensiones

En un esquema en estrella, tabla más pequeña que contiene los atributos de datos sobre los datos cuantitativos en una tabla de hechos. Los atributos de la tabla de dimensiones suelen ser campos de texto o números discretos que se comportan como texto. Estos atributos se suelen utilizar para restringir consultas, filtrarlas y etiquetar los conjuntos de resultados.

desastre

Un evento que impide que una carga de trabajo o un sistema cumplan sus objetivos empresariales en su ubicación principal de implementación. Estos eventos pueden ser desastres naturales, fallos técnicos o el resultado de acciones humanas, como una configuración incorrecta involuntaria o un ataque de malware.

recuperación de desastres (DR)

Estrategia y proceso que utiliza para minimizar el tiempo de inactividad y la pérdida de datos a causa de un desastre. Para obtener más información, consulte Recuperación de cargas de trabajo ante desastres en AWS: Recuperación en la nube en el AWS Well-Architected marco.

DML

Consulte lenguaje de manipulación de bases de datos.

diseño basado en el dominio

Un enfoque para desarrollar un sistema de software complejo mediante la conexión de sus componentes a dominios en evolución, o a los objetivos empresariales principales, a los que sirve cada componente. Eric Evans introdujo este concepto en su libro Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). Para

obtener información sobre cómo utilizar el diseño basado en dominios con el patrón de higos estranguladores, consulte Modernización gradual de los servicios web antiguos de ASP.NET Microsoft (ASMX) mediante contenedores y Amazon API Gateway.

DR

Consulte recuperación ante desastres.

Detección de desviaciones

Seguimiento de las desviaciones con respecto a una configuración con línea de base. Por ejemplo, puedes usarlo AWS CloudFormation para detectar desviaciones en los recursos del sistema o puedes usarlo AWS Control Tower para detectar cambios en tu landing zone que puedan afectar al cumplimiento de los requisitos de gobierno.

DVSM

Consulte asignación de flujos de valor para el desarrollo.

E

EDA

Consulte análisis de datos de tipo exploratorio.

EDI

Consulte intercambio electrónico de datos.

computación en la periferia

La tecnología que aumenta la potencia de cálculo de los dispositivos inteligentes en la periferia de una red de IoT. En comparación con la computación en la nube, la computación de periferia puede reducir la latencia de la comunicación y mejorar el tiempo de respuesta.

intercambio electrónico de datos (EDI)

Intercambio automatizado de documentos comerciales entre organizaciones. Para más información, consulte ¿Qué es el intercambio electrónico de datos?

cifrado

Proceso de computación que transforma datos de texto plano, que son legibles por humanos, en texto cifrado.

clave de cifrado

Cadena criptográfica de bits aleatorios que se genera mediante un algoritmo de cifrado. Las claves pueden variar en longitud y cada una se ha diseñado para ser impredecible y única.

endianidad

El orden en el que se almacenan los bytes en la memoria del ordenador. Big-endian los sistemas almacenan primero el byte más significativo. Little-endian los sistemas almacenan primero el byte menos significativo.

punto de conexión

Consulte punto de conexión de servicio.

servicio de punto de conexión

Servicio que puede alojar en una nube privada virtual (VPC) para compartir con otros usuarios. Puede crear un servicio de punto final con AWS PrivateLink entidades principales Cuentas de AWS o AWS Identity and Access Management (de IAM) y conceder permisos a ellas. Estas cuentas o entidades principales pueden conectarse a su servicio de punto de conexión de forma privada mediante la creación de puntos de conexión de VPC de interfaz. Para obtener más información, consulte Creación de un servicio de punto de conexión en la documentación de Amazon Virtual Private Cloud (Amazon VPC).

planificación de recursos empresariales (ERP)

Sistema que automatiza y administra los procesos empresariales clave (como la contabilidad, MES y la administración de proyectos) de una empresa.

cifrado de sobre

El proceso de cifrar una clave de cifrado con otra clave de cifrado. Para obtener más información, consulte el cifrado de sobres en la documentación de AWS Key Management Service (AWS KMS).

entorno

Una instancia de una aplicación en ejecución. Los siguientes son los tipos de entornos más comunes en la computación en la nube:

- entorno de desarrollo: instancia de una aplicación en ejecución que solo se encuentra disponible para el equipo principal responsable del mantenimiento de la aplicación. Los

entornos de desarrollo se utilizan para probar los cambios antes de promocionarlos a los entornos superiores. Este tipo de entorno a veces se denomina entorno de prueba.

- entornos inferiores: todos los entornos de desarrollo de una aplicación, como los que se utilizan para las compilaciones y pruebas iniciales.
- entorno de producción: instancia de una aplicación en ejecución a la que pueden acceder los usuarios finales. En un CI/CD proceso, el entorno de producción es el último entorno de implementación.
- entornos superiores: todos los entornos a los que pueden acceder usuarios que no sean del equipo de desarrollo principal. Esto puede incluir un entorno de producción, entornos de preproducción y entornos para las pruebas de aceptación por parte de los usuarios.

epopeya

En las metodologías ágiles, son categorías funcionales que ayudan a organizar y priorizar el trabajo. Las epopeyas brindan una descripción detallada de los requisitos y las tareas de implementación. Por ejemplo, las epopeyas AWS de seguridad de CAF incluyen la gestión de identidades y accesos, los controles de detección, la seguridad de la infraestructura, la protección de datos y la respuesta a incidentes. Para obtener más información sobre las epopeyas en la estrategia de migración de AWS , consulte la Guía de implementación del programa.

ERP

Consulte planificación de recursos empresariales.

análisis de datos de tipo exploratorio (EDA)

El proceso de analizar un conjunto de datos para comprender sus características principales. Se recopilan o agregan datos y, a continuación, se realizan las investigaciones iniciales para encontrar patrones, detectar anomalías y comprobar las suposiciones. El EDA se realiza mediante el cálculo de estadísticas resumidas y la creación de visualizaciones de datos.

F

tabla de hechos

Tabla central de un esquema en estrella. Almacena datos cuantitativos sobre operaciones empresariales. Por lo general, una tabla de hechos contiene dos tipos de columnas: las que contienen medidas y las que contienen una clave externa para una tabla de dimensiones.

Fail Fast

Filosofía que utiliza pruebas frecuentes e incrementales para reducir el ciclo de vida del desarrollo. Es una parte fundamental de los enfoques ágiles.

límite de aislamiento de errores

En el Nube de AWS, un límite, como una zona de disponibilidad Región de AWS, un plano de control o un plano de datos, que limita el efecto de una falla y ayuda a mejorar la resiliencia de las cargas de trabajo. Para más información, consulte AWS Fault Isolation Boundaries.

rama de característica

Consulte rama.

características

Los datos de entrada que se utilizan para hacer una predicción. Por ejemplo, en un contexto de fabricación, las características pueden ser imágenes que se capturan periódicamente desde la línea de fabricación.

importancia de las características

La importancia que tiene una característica para las predicciones de un modelo. Por lo general, esto se expresa como una puntuación numérica que se puede calcular mediante diversas técnicas, como las explicaciones aditivas de Shapley (SHAP) y los gradientes integrados. Para obtener más información, consulte Interpretabilidad del modelo de aprendizaje automático con. AWS

transformación de funciones

Optimizar los datos para el proceso de ML, lo que incluye enriquecer los datos con fuentes adicionales, escalar los valores o extraer varios conjuntos de información de un solo campo de datos. Esto permite que el modelo de ML se beneficie de los datos. Por ejemplo, si divide la fecha del “27 de mayo de 2021 00:15:37” en “jueves”, “mayo”, “2021” y “15”, puede ayudar al algoritmo de aprendizaje a aprender patrones matizados asociados a los diferentes componentes de los datos.

peticiones con pocos pasos

Proporcionar a un LLM una pequeña cantidad de ejemplos que demuestren la tarea y el resultado deseado antes de pedirle que lleve a cabo una tarea similar. Esta técnica es una aplicación del aprendizaje contextual, en el que los modelos aprenden a partir de ejemplos (tomas) integrados en las instrucciones. Few-shot Las indicaciones pueden ser eficaces para tareas que requieren

un formato, un razonamiento o un conocimiento del dominio específicos. Consulte también peticiones desde cero.

FGAC

Consulte control de acceso detallado.

control de acceso preciso (FGAC)

El uso de varias condiciones que tienen por objetivo permitir o denegar una solicitud de acceso.

migración relámpago

Método de migración de bases de datos que utiliza la replicación continua de datos mediante la captura de datos de cambio para migrar los datos en el menor tiempo posible, en lugar de utilizar un enfoque gradual. El objetivo es reducir al mínimo el tiempo de inactividad.

FM

Consulte modelo fundacional.

Modelo fundacional (FM)

Gran red neuronal de aprendizaje profundo que se entrenó con conjuntos de datos masivos de datos generalizados y no etiquetados. Los FM pueden hacer una amplia variedad de tareas generales, como comprender el lenguaje, generar texto e imágenes y conversar en lenguaje natural. Para más información, consulte ¿Qué son los modelos fundacionales?

Puerta de enlace FM

Un intermediario centralizado que controla y normaliza el acceso a los modelos básicos. También se conoce como puerta de enlace LLM.

G

IA generativa

Subconjunto de modelos de IA que se entrenaron con grandes cantidades de datos y que pueden utilizar una simple petición de texto para crear contenido y artefactos nuevos, como imágenes, videos, texto y audio. Para más información, consulte ¿Qué es la IA generativa?

bloqueo geográfico

Consulte restricciones geográficas.

restricciones geográficas (bloqueo geográfico)

En Amazon CloudFront, una opción para impedir que los usuarios de países específicos accedan a las distribuciones de contenido. Puede utilizar una lista de permitidos o bloqueados para especificar los países aprobados y prohibidos. Para obtener más información, consulta Restringir la distribución geográfica del contenido en la CloudFront documentación.

Flujo de trabajo de Gitflow

Un enfoque en el que los entornos inferiores y superiores utilizan diferentes ramas en un repositorio de código fuente. El flujo de trabajo de Gitflow se considera heredado, mientras que el flujo de trabajo basado en enlaces troncales es el enfoque moderno preferido.

imagen dorada

Instantánea de un sistema o software que se usa como plantilla para implementar nuevas instancias de ese sistema o software. Por ejemplo, en la fabricación, una imagen dorada se puede utilizar para aprovisionar software en varios dispositivos y ayuda a mejorar la velocidad, la escalabilidad y la productividad de las operaciones de fabricación de dispositivos.

estrategia de implementación desde cero

La ausencia de infraestructura existente en un entorno nuevo. Al adoptar una estrategia de implementación desde cero para una arquitectura de sistemas, puede seleccionar todas las tecnologías nuevas sin que estas deban ser compatibles con una infraestructura existente, lo que también se conoce como implementación sobre infraestructura existente. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

barrera de protección

Una regla de alto nivel que ayuda a regular los recursos, las políticas y la conformidad en todas las unidades organizativas (OU). Las barreras de protección preventivas aplican políticas para garantizar la alineación con los estándares de conformidad. Se implementan mediante políticas de control de servicios y límites de permisos de IAM. Las barreras de protección de detección detectan las vulneraciones de las políticas y los problemas de conformidad, y generan alertas para su corrección. Se implementan mediante Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, Amazon Inspector y AWS Lambda cheques personalizados.

barandas (AI)

Mecanismos de seguridad que filtran, validan y restringen las entradas y salidas de los agentes para ayudar a garantizar un comportamiento responsable y seguro de la IA.

H

HA

Consulte alta disponibilidad.

migración heterogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que utilice un motor de base de datos diferente (por ejemplo, de Oracle a Amazon Aurora). La migración heterogénea suele ser parte de un esfuerzo de rediseño de la arquitectura y convertir el esquema puede ser una tarea compleja. AWS ofrece AWS SCT, lo cual ayuda con las conversiones de esquemas.

alta disponibilidad (HA)

La capacidad de una carga de trabajo para funcionar de forma continua, sin intervención, en caso de desafíos o desastres. Los sistemas de alta disponibilidad están diseñados para realizar una conmutación por error automática, ofrecer un rendimiento de alta calidad de forma constante y gestionar diferentes cargas y fallos con un impacto mínimo en el rendimiento.

modernización histórica

Un enfoque utilizado para modernizar y actualizar los sistemas de tecnología operativa (TO) a fin de satisfacer mejor las necesidades de la industria manufacturera. Un histórico es un tipo de base de datos que se utiliza para recopilar y almacenar datos de diversas fuentes en una fábrica.

datos de reserva

Parte de los datos históricos etiquetados que se ocultan de un conjunto de datos que se utiliza para entrenar un modelo de machine learning. Puede utilizar los datos de reserva para evaluar el rendimiento del modelo mediante la comparación de las predicciones del modelo con los datos de reserva.

human-in-the-loop (HiTL)

Un patrón de flujo de trabajo en el que la ejecución de los agentes se detiene para su revisión y aprobación por parte de una persona en los puntos de decisión críticos.

migración homogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que comparte el mismo motor de base de datos (por ejemplo, Microsoft SQL Server a Amazon RDS para SQL Server).

La migración homogénea suele formar parte de un esfuerzo para volver a alojar o redefinir la plataforma. Puede utilizar las utilidades de bases de datos nativas para migrar el esquema.

datos recientes

Datos a los que se accede con frecuencia, como datos en tiempo real o datos traslacionales recientes. Por lo general, estos datos requieren un nivel o una clase de almacenamiento de alto rendimiento para proporcionar respuestas rápidas a las consultas.

hotfix

Una solución urgente para un problema crítico en un entorno de producción. Debido a su urgencia, una revisión suele realizarse fuera del flujo de trabajo habitual de las DevOps versiones.

periodo de hiperatención

Periodo, inmediatamente después de la transición, durante el cual un equipo de migración administra y monitorea las aplicaciones migradas en la nube para solucionar cualquier problema. Por lo general, este periodo dura de 1 a 4 días. Al final del periodo de hiperatención, el equipo de migración suele transferir la responsabilidad de las aplicaciones al equipo de operaciones en la nube.

I

IaC

Consulte infraestructura como código.

políticas basadas en identidades

Política asociada a uno o más directores de IAM que define sus permisos en el entorno. Nube de AWS

aplicación inactiva

Aplicación que utiliza un promedio de CPU y memoria de entre 5 y 20 por ciento durante un periodo de 90 días. En un proyecto de migración, es habitual retirar estas aplicaciones o mantenerlas en las instalaciones.

IIoT

Consulte Internet de las cosas industrial.

infraestructura inmutable

Modelo que implementa una nueva infraestructura para las cargas de trabajo de producción en lugar de actualizar o modificar la infraestructura existente o aplicarle revisiones. Las infraestructuras inmutables son de manera intrínseca más coherentes, fiables y predecibles que las infraestructuras mutables. Para obtener más información, consulte las mejores prácticas del Framework para implementar con una infraestructura inmutable. AWS Well-Architected

VPC entrante (de entrada)

En una arquitectura de AWS cuentas múltiples, una VPC que acepta, inspecciona y enruta las conexiones de red desde fuera de una aplicación. La Arquitectura de referencia de seguridad de AWS recomienda configurar su cuenta de red con VPC entrantes, salientes y de inspección para proteger la interfaz bidireccional entre su aplicación e Internet en general.

migración gradual

Estrategia de transición en la que se migra la aplicación en partes pequeñas en lugar de realizar una transición única y completa. Por ejemplo, puede trasladar inicialmente solo unos pocos microservicios o usuarios al nuevo sistema. Tras comprobar que todo funciona correctamente, puede trasladar microservicios o usuarios adicionales de forma gradual hasta que pueda retirar su sistema heredado. Esta estrategia reduce los riesgos asociados a las grandes migraciones.

Industria 4.0

Un término que Klaus Schwab introdujo en 2016 para referirse a la modernización de los procesos de fabricación mediante avances en la conectividad, los datos en tiempo real, la automatización, el análisis y. AI/ML

infraestructura

Todos los recursos y activos que se encuentran en el entorno de una aplicación.

infraestructura como código (IaC)

Proceso de aprovisionamiento y administración de la infraestructura de una aplicación mediante un conjunto de archivos de configuración. La IaC se ha diseñado para ayudarlo a centralizar la administración de la infraestructura, estandarizar los recursos y escalar con rapidez a fin de que los entornos nuevos sean repetibles, fiables y consistentes.

Internet de las cosas industrial (IIoT)

El uso de sensores y dispositivos conectados a Internet en los sectores industriales, como el productivo, el eléctrico, el automotriz, el sanitario, el de las ciencias de la vida y el de la

agricultura. Para obtener más información, consulte Creación de una estrategia de transformación digital del Internet de las cosas industrial (IIoT).

VPC de inspección

En una arquitectura de AWS cuentas múltiples, una VPC centralizada que gestiona las inspecciones del tráfico de red entre las VPC (iguales o Regiones de AWS diferentes), Internet y las redes locales. La Arquitectura de referencia de seguridad de AWS recomienda configurar su cuenta de red con VPC entrantes, salientes y de inspección para proteger la interfaz bidireccional entre su aplicación e Internet en general.

Internet de las cosas (IoT)

Red de objetos físicos conectados con sensores o procesadores integrados que se comunican con otros dispositivos y sistemas a través de Internet o de una red de comunicación local. Para obtener más información, consulte ¿Qué es IoT?.

interpretabilidad

Característica de un modelo de machine learning que describe el grado en que un ser humano puede entender cómo las predicciones del modelo dependen de sus entradas. Para obtener más información, consulte Interpretabilidad del modelo de aprendizaje automático con. AWS

IoT

Consulte Internet de las cosas.

biblioteca de información de TI (ITIL)

Conjunto de prácticas recomendadas para ofrecer servicios de TI y alinearlos con los requisitos empresariales. La ITIL proporciona la base para la ITSM.

administración de servicios de TI (ITSM)

Actividades asociadas con el diseño, la implementación, la administración y el soporte de los servicios de TI para una organización. Para obtener información sobre la integración de las operaciones en la nube con las herramientas de ITSM, consulte la Guía de integración de operaciones.

ITIL

Consulte biblioteca de información de TI.

ITSM

Consulte administración de servicios de TI.

L

control de acceso basado en etiquetas (LBAC)

Una implementación del control de acceso obligatorio (MAC) en la que a los usuarios y a los propios datos se les asigna explícitamente un valor de etiqueta de seguridad. La intersección entre la etiqueta de seguridad del usuario y la etiqueta de seguridad de los datos determina qué filas y columnas puede ver el usuario.

zona de aterrizaje

Una landing zone es un AWS entorno multicuenta bien diseñado, escalable y seguro. Este es un punto de partida desde el cual las empresas pueden lanzar e implementar rápidamente cargas de trabajo y aplicaciones con confianza en su entorno de seguridad e infraestructura. Para obtener más información sobre las zonas de aterrizaje, consulte Configuración de un entorno de AWS seguro y escalable con varias cuentas.

modelo de lenguaje de gran tamaño (LLM)

Modelo de IA de aprendizaje profundo que se entrenó previamente con una gran cantidad de datos. Un LLM puede llevar a cabo varias tareas, como responder preguntas, resumir documentos, traducir textos a otros idiomas y completar oraciones. Para más información, consulte ¿Qué es un LLM (modelo de lenguaje de gran tamaño)?

migración grande

Migración de 300 servidores o más.

LBAC

Consulte control de acceso basado en etiquetas.

privilegio mínimo

La práctica recomendada de seguridad que consiste en conceder los permisos mínimos necesarios para realizar una tarea. Para obtener más información, consulte Aplicar permisos de privilegio mínimo en la documentación de IAM.

migrar mediante lift-and-shift

Consulte Las 7 R.

sistema little-endian

Un sistema que almacena primero el byte menos significativo. Consulte también endianidad.

LLM

Consulte modelo de lenguaje de gran tamaño.

entornos inferiores

Consulte entorno.

M

machine learning (ML)

Un tipo de inteligencia artificial que utiliza algoritmos y técnicas para el reconocimiento y el aprendizaje de patrones. El ML analiza y aprende de los datos registrados, como los datos del Internet de las cosas (IoT), para generar un modelo estadístico basado en patrones. Para más información, consulte Machine learning.

rama principal

Consulte rama.

malware

Software diseñado para comprometer la seguridad o la privacidad de la computadora. El malware podría interrumpir los sistemas informáticos, filtrar información confidencial u obtener acceso no autorizado. Algunos ejemplos de malware son los virus, los gusanos, el ransomware, los troyanos, el spyware y los registradores de pulsaciones de teclas.

Servicios administrados

Servicios de AWS en el que AWS opera la capa de infraestructura, el sistema operativo y las plataformas, y se accede a los puntos finales para almacenar y recuperar datos. Amazon Simple Storage Service (Amazon S3) y Amazon DynamoDB son ejemplos de servicios administrados. También se conocen como servicios abstractos.

sistema de ejecución de fabricación (MES)

Sistema de software para seguir, supervisar, documentar y controlar los procesos de producción que convierten las materias primas en productos acabados en la zona de producción.

MAP

Consulte Programa de aceleración de la migración.

MCP

Consulte Model Context Protocol.

Protocolo de contexto para modelos (MCP)

Un protocolo sin estado para la comunicación entre el agente y la herramienta.

Servidor MCP

Un servicio que expone una o más herramientas a través del protocolo Model Context.

mecanismo

Proceso completo mediante el que se crea una herramienta, se impulsa su adopción y, a continuación, se inspeccionan los resultados para hacer ajustes. Un mecanismo es un ciclo que se refuerza y mejora por sí mismo a medida que funciona. Para obtener más información, consulte Creación de mecanismos en el AWS Well-Architected marco.

cuenta de miembro

Todas las Cuentas de AWS demás cuentas, excepto la de administración, que forman parte de una organización AWS Organizations. Una cuenta no puede pertenecer a más de una organización a la vez.

MES

Consulte sistema de ejecución de fabricación.

Message Queuing Telemetry Transport (MQTT)

Un protocolo de comunicación ligero de máquina a máquina (M2M), basado en el publish/subscribepatrón, para dispositivos de IoT con recursos limitados.

microservicio

Un servicio pequeño e independiente que se comunica a través de API bien definidas y que, por lo general, es propiedad de equipos pequeños e independientes. Por ejemplo, un sistema de seguros puede incluir microservicios que se adapten a las capacidades empresariales, como las de ventas o marketing, o a subdominios, como las de compras, reclamaciones o análisis. Los beneficios de los microservicios incluyen la agilidad, la escalabilidad flexible, la facilidad de implementación, el código reutilizable y la resiliencia. Para obtener más información, consulte Integrar microservicios mediante servicios sin servidor. AWS

arquitectura de microservicios

Un enfoque para crear una aplicación con componentes independientes que ejecutan cada proceso de la aplicación como un microservicio. Estos microservicios se comunican a través de una interfaz bien definida mediante API ligeras. Cada microservicio de esta arquitectura se puede actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación. Para obtener más información, consulte Implementación de microservicios en. AWS

Programa de aceleración de la migración (MAP)

Un AWS programa que proporciona soporte de consultoría, formación y servicios para ayudar a las organizaciones a crear una base operativa sólida para migrar a la nube y para ayudar a compensar el costo inicial de las migraciones. El MAP incluye una metodología de migración para ejecutar las migraciones antiguas de forma metódica y un conjunto de herramientas para automatizar y acelerar los escenarios de migración más comunes.

migración a escala

Proceso de transferencia de la mayoría de la cartera de aplicaciones a la nube en oleadas, con más aplicaciones desplazadas a un ritmo más rápido en cada oleada. En esta fase, se utilizan las prácticas recomendadas y las lecciones aprendidas en las fases anteriores para implementar una fábrica de migración de equipos, herramientas y procesos con el fin de agilizar la migración de las cargas de trabajo mediante la automatización y la entrega ágil. Esta es la tercera fase de la estrategia de migración de AWS.

fábrica de migración

Cross-functional equipos que agilizan la migración de las cargas de trabajo mediante enfoques ágiles y automatizados. Los equipos de las fábricas de migración suelen estar compuestos por analistas y propietarios de operaciones, ingenieros de migración, desarrolladores y DevOps profesionales que trabajan a pasos agigantados. Entre el 20 y el 50 por ciento de la cartera de aplicaciones empresariales se compone de patrones repetidos que pueden optimizarse mediante un enfoque de fábrica. Para obtener más información, consulte la discusión sobre las fábricas de migración y la Guía de fábricas de migración a la nube en este contenido.

metadatos de migración

Información sobre la aplicación y el servidor que se necesita para completar la migración. Cada patrón de migración requiere un conjunto diferente de metadatos de migración. Algunos ejemplos de metadatos de migración son la subred de destino, el grupo de seguridad y AWS la cuenta.

patrón de migración

Tarea de migración repetible que detalla la estrategia de migración, el destino de la migración y la aplicación o el servicio de migración utilizados. Ejemplo: rehospede la migración a Amazon EC2 AWS con Application Migration Service.

Migration Portfolio Assessment (MPA)

Herramienta en línea que proporciona información a fin de validar los argumentos comerciales necesarios para migrar a la Nube de AWS. La MPA ofrece una evaluación detallada de la cartera (adecuación del tamaño de los servidores, precios, comparaciones del costo total de propiedad, análisis de los costos de migración), así como una planificación de la migración (análisis y recopilación de datos de aplicaciones, agrupación de aplicaciones, priorización de la migración y planificación de oleadas). La herramienta MPA (requiere iniciar sesión) está disponible de forma gratuita para todos los AWS consultores y consultores de los socios de APN.

Evaluación de la preparación para la migración (MRA)

Proceso que consiste en obtener información sobre el estado de preparación de una organización para la nube, identificar sus puntos fuertes y débiles y elaborar un plan de acción para cerrar las brechas identificadas mediante el AWS CAF. Para obtener más información, consulte la Guía de preparación para la migración. La MRA es la primera fase de la estrategia de migración de AWS.

estrategia de migración

Enfoque utilizado para migrar una carga de trabajo a la Nube de AWS. Para más información, consulte la entrada Las 7 R de este glosario y también Mobilize your organization to accelerate large-scale migrations.

ML

Consulte machine learning.

modernización

Transformar una aplicación obsoleta (antigua o monolítica) y su infraestructura en un sistema ágil, elástico y de alta disponibilidad en la nube para reducir los gastos, aumentar la eficiencia y aprovechar las innovaciones. Para más información, consulte Strategy for modernizing applications in the Nube de AWS.

evaluación de la preparación para la modernización

Evaluación que ayuda a determinar la preparación para la modernización de las aplicaciones de una organización; identifica los beneficios, los riesgos y las dependencias; y determina qué

tan bien la organización puede soportar el estado futuro de esas aplicaciones. El resultado de la evaluación es un esquema de la arquitectura objetivo, una hoja de ruta que detalla las fases de desarrollo y los hitos del proceso de modernización y un plan de acción para abordar las brechas identificadas. Para más información, consulte Evaluating modernization readiness for applications in the Nube de AWS.

aplicaciones monolíticas (monolitos)

Aplicaciones que se ejecutan como un único servicio con procesos estrechamente acoplados. Las aplicaciones monolíticas presentan varios inconvenientes. Si una característica de la aplicación experimenta un aumento en la demanda, se debe escalar toda la arquitectura. Agregar o mejorar las características de una aplicación monolítica también se vuelve más complejo a medida que crece la base de código. Para solucionar problemas con la aplicación, puede utilizar una arquitectura de microservicios. Para obtener más información, consulte Descomposición de monolitos en microservicios.

MPA

Consulte Migration Portfolio Assessment.

MQTT

Consulte Message Queuing Telemetry Transport.

clasificación multiclase

Un proceso que ayuda a generar predicciones para varias clases (predice uno de más de dos resultados). Por ejemplo, un modelo de ML podría preguntar “¿Este producto es un libro, un automóvil o un teléfono?” o “¿Qué categoría de productos es más interesante para este cliente?”.

infraestructura mutable

Modelo que actualiza y modifica la infraestructura actual para las cargas de trabajo de producción. Para mejorar la coherencia, la confiabilidad y la previsibilidad, el AWS Well-Architected Marco recomienda el uso de una infraestructura inmutable como práctica recomendada.

O

OAC

Consulte control de acceso de origen.

OAI

Consulte identidad de acceso de origen.

OCM

Consulte administración del cambio organizacional.

migración fuera de línea

Método de migración en el que la carga de trabajo de origen se elimina durante el proceso de migración. Este método implica un tiempo de inactividad prolongado y, por lo general, se utiliza para cargas de trabajo pequeñas y no críticas.

OI

Consulte integración de operaciones.

OLA

Consulte acuerdo de nivel operativo.

migración en línea

Método de migración en el que la carga de trabajo de origen se copia al sistema de destino sin que se desconecte. Las aplicaciones que están conectadas a la carga de trabajo pueden seguir funcionando durante la migración. Este método implica un tiempo de inactividad nulo o mínimo y, por lo general, se utiliza para cargas de trabajo de producción críticas.

OPC-UA

Consulte Open Process Communications: arquitectura unificada.

Comunicaciones de proceso abierto: arquitectura unificada () OPC-UA

Un protocolo de comunicación de máquina a máquina (M2M) para la automatización industrial. OPC-UA proporciona un estándar de interoperabilidad con esquemas de cifrado, autenticación y autorización de datos.

acuerdo de nivel operativo (OLA)

Acuerdo que aclara lo que los grupos de TI operativos se comprometen a ofrecerse entre sí, para respaldar un acuerdo de nivel de servicio (SLA).

revisión de la preparación operativa (ORR)

Lista de comprobación de preguntas y prácticas recomendadas asociadas que son útiles para comprender, evaluar, prevenir o reducir el alcance de los incidentes y posibles errores. Para

obtener más información, consulte las revisiones de preparación operativa (ORR) en el AWS Well-Architected marco.

tecnología operativa (TO)

Sistemas de hardware y software que funcionan con el entorno físico para controlar las operaciones, los equipos y la infraestructura industriales. En el sector de la fabricación, la integración de los sistemas de TO y tecnología de la información (TI) es un enfoque clave para las transformaciones de la industria 4.0.

integración de operaciones (OI)

Proceso de modernización de las operaciones en la nube, que implica la planificación de la preparación, la automatización y la integración. Para obtener más información, consulte la Guía de integración de las operaciones.

registro de seguimiento organizativo

Un registro creado por y AWS CloudTrail que registra todos los eventos Cuentas de AWS de una organización AWS Organizations. Este registro de seguimiento se crea en cada Cuenta de AWS que forma parte de la organización y realiza un seguimiento de la actividad en cada cuenta. Para obtener más información, consulte Crear un registro para una organización en la CloudTrail documentación.

administración del cambio organizacional (OCM)

Marco para administrar las transformaciones empresariales importantes y disruptivas desde la perspectiva de las personas, la cultura y el liderazgo. La OCM ayuda a las empresas a prepararse para nuevos sistemas y estrategias y a realizar la transición a ellos, al acelerar la adopción de cambios, abordar los problemas de transición e impulsar cambios culturales y organizacionales. En la estrategia de AWS migración, este marco se denomina aceleración de personal, debido a la velocidad de cambio que requieren los proyectos de adopción de la nube. Para obtener más información, consulte la Guía de OCM.

control de acceso de origen (OAC)

En CloudFront, una opción mejorada para restringir el acceso y proteger el contenido del Amazon Simple Storage Service (Amazon S3). El OAC admite todos los buckets de S3 Regiones de AWS, el cifrado del lado del servidor con AWS KMS (SSE-KMS) y `DELETE` las solicitudes `PUT` y dinámicas al bucket de S3.

identidad de acceso de origen (OAI)

En CloudFront, una opción para restringir el acceso y proteger el contenido de Amazon S3. Cuando utiliza OAI, CloudFront crea un principal con el que Amazon S3 puede autenticarse. Los directores autenticados solo pueden acceder al contenido de un bucket de S3 a través de una distribución específica. CloudFront Consulte también el OAC, que proporciona un control de acceso más detallado y mejorado.

ORR

Consulte revisión de la preparación operativa.

OT

Consulte tecnología operativa.

VPC saliente (de salida)

En una arquitectura de AWS cuentas múltiples, una VPC que gestiona las conexiones de red que se inician desde una aplicación. La Arquitectura de referencia de seguridad de AWS recomienda configurar su cuenta de red con VPC entrantes, salientes y de inspección para proteger la interfaz bidireccional entre su aplicación e Internet en general.

P

límite de permisos

Una política de administración de IAM que se adjunta a las entidades principales de IAM para establecer los permisos máximos que puede tener el usuario o el rol. Para obtener más información, consulte Límites de permisos en la documentación de IAM.

información de identificación personal (PII)

Información que, vista directamente o combinada con otros datos relacionados, puede utilizarse para deducir de manera razonable la identidad de una persona. Algunos ejemplos de información de identificación personal son los nombres, las direcciones y la información de contacto.

PII

Consulte información de identificación personal.

manual de estrategias

Conjunto de pasos predefinidos que capturan el trabajo asociado a las migraciones, como la entrega de las funciones de operaciones principales en la nube. Un manual puede adoptar la forma de scripts, manuales de procedimientos automatizados o resúmenes de los procesos o pasos necesarios para operar un entorno modernizado.

PLC

Consulte controlador lógico programable.

PLM

Consulte administración del ciclo de vida del producto.

policy

Objeto que puede definir permisos (consulte política basada en identidad), especificar las condiciones de acceso (consulte política basada en recursos) o definir los permisos máximos para todas las cuentas de una organización de AWS Organizations (consulte política de control de servicio).

persistencia políglota

Elegir de forma independiente la tecnología de almacenamiento de datos de un microservicio en función de los patrones de acceso a los datos y otros requisitos. Si sus microservicios tienen la misma tecnología de almacenamiento de datos, pueden enfrentarse a desafíos de implementación o experimentar un rendimiento deficiente. Los microservicios se implementan más fácilmente y logran un mejor rendimiento y escalabilidad si utilizan el almacén de datos que mejor se adapte a sus necesidades.

evaluación de cartera

Proceso de detección, análisis y priorización de la cartera de aplicaciones para planificar la migración. Para obtener más información, consulte la Evaluación de la preparación para la migración.

predicate

Condición de consulta que devuelve `true` o `false`. En general, se encuentra en una cláusula `WHERE`.

inserción de predicados

Técnica de optimización de consultas en bases de datos que filtra los datos de la consulta antes de transferirlos. Esta técnica reduce la cantidad de datos de la base de datos relacional que se tienen que recuperar y procesar. Además, mejora el rendimiento de las consultas.

control preventivo

Un control de seguridad diseñado para evitar que ocurra un evento. Estos controles son la primera línea de defensa para evitar el acceso no autorizado o los cambios no deseados en la red. Para obtener más información, consulte Controles preventivos en Implementación de controles de seguridad en AWS.

entidad principal

Una entidad AWS que puede realizar acciones y acceder a los recursos. Esta entidad suele ser un usuario raíz para un Cuenta de AWS rol de IAM o un usuario. Para obtener más información, consulte Entidad principal en Términos y conceptos de roles en la documentación de IAM.

Privacidad desde el diseño

Enfoque de ingeniería de sistemas que tiene en cuenta la privacidad durante todo el proceso de desarrollo.

zonas alojadas privadas

Contenedor que aloja información acerca de cómo desea que responda Amazon Route 53 a las consultas de DNS de un dominio y sus subdominios en una o varias VPC. Para obtener más información, consulte Uso de zonas alojadas privadas en la documentación de Route 53.

control proactivo

Control de seguridad que se diseñó para evitar la implementación de recursos que no cumplan con la normativa. Estos controles analizan los recursos antes de aprovisionarlos. Si el recurso no cumple con los requisitos del control, no se aprovisiona. Para obtener más información, consulte la guía de referencia de controles en la AWS Control Tower documentación y consulte Controles proactivos en Implementación de controles de seguridad en AWS.

administración del ciclo de vida del producto (PLM)

Administración de los datos y los procesos de un producto a lo largo de todo su ciclo de vida, desde el diseño, el desarrollo y el lanzamiento, pasando por el crecimiento y la madurez, hasta la reducción de su uso y su retirada.

entorno de producción

Consulte entorno.

controlador lógico programable (PLC)

En el sector de la fabricación, computadora adaptable y altamente fiable que supervisa las máquinas y automatiza los procesos de fabricación.

encadenamiento de peticiones

Uso de la salida de una petición de LLM como entrada para la siguiente petición a fin de generar mejores respuestas. Esta técnica se utiliza para dividir una tarea compleja en tareas secundarias o para refinar o ampliar de forma iterativa una respuesta preliminar. Ayuda a mejorar la precisión y la relevancia de las respuestas de un modelo y permite obtener resultados más detallados y personalizados.

seudonimización

El proceso de reemplazar los identificadores personales de un conjunto de datos por valores de marcadores de posición. La seudonimización puede ayudar a proteger la privacidad personal. Los datos seudonimizados siguen considerándose datos personales.

publish/subscribe (pub/sub)

Patrón que permite establecer comunicaciones asíncronas entre microservicios para mejorar la escalabilidad y la capacidad de respuesta. Por ejemplo, en un MES basado en microservicios, un microservicio puede publicar mensajes de eventos en un canal al que se pueden suscribir otros microservicios. El sistema puede agregar nuevos microservicios sin cambiar el servicio de publicación.

Q

plan de consulta

Serie de pasos, como instrucciones, que se utilizan para acceder a los datos de un sistema de base de datos relacional SQL.

regresión del plan de consulta

El optimizador de servicios de la base de datos elige un plan menos óptimo que antes de un cambio determinado en el entorno de la base de datos. Los cambios en estadísticas,

restricciones, configuración del entorno, enlaces de parámetros de consultas y actualizaciones del motor de base de datos PostgreSQL pueden provocar una regresión del plan.

R

Matriz RACI

Consulte responsable, fiable, consultada e informada (RACI).

RAG

Consulte generación aumentada por recuperación.

ransomware

Software malicioso que se ha diseñado para bloquear el acceso a un sistema informático o a los datos hasta que se efectúe un pago.

Matriz RASCI

Consulte responsable, fiable, consultada e informada (RACI).

RCAC

Consulte control de acceso por filas y columnas.

réplica de lectura

Una copia de una base de datos que se utiliza con fines de solo lectura. Puede enrutar las consultas a la réplica de lectura para reducir la carga en la base de datos principal.

rediseñar

Consulte Las 7 R.

objetivo de punto de recuperación (RPO)

La cantidad de tiempo máximo aceptable desde el último punto de recuperación de datos. Esto determina qué se considera una pérdida de datos aceptable entre el último punto de recuperación y la interrupción del servicio.

objetivo de tiempo de recuperación (RTO)

La demora máxima aceptable entre la interrupción del servicio y el restablecimiento del servicio.

refactorizar

Consulte Las 7 R.

Region

Conjunto de AWS recursos en un área geográfica. Cada uno Región de AWS está aislado e independiente de los demás para proporcionar tolerancia a las fallas, estabilidad y resiliencia. Para más información, consulte Specify which Regiones de AWS your account can use.

regresión

Una técnica de ML que predice un valor numérico. Por ejemplo, para resolver el problema de "¿A qué precio se venderá esta casa?", un modelo de ML podría utilizar un modelo de regresión lineal para predecir el precio de venta de una vivienda en función de datos conocidos sobre ella (por ejemplo, los metros cuadrados).

volver a alojar

Consulte Las 7 R.

versión

En un proceso de implementación, el acto de promover cambios en un entorno de producción.

reubicar

Consulte Las 7 R.

redefinir la plataforma

Consulte Las 7 R.

recomprar

Consulte Las 7 R.

resiliencia

Capacidad de una aplicación para resistir interrupciones o recuperarse de ellas. Al planificar la resiliencia en la Nube de AWS, la alta disponibilidad y la recuperación ante desastres son consideraciones comunes. Para más información, consulte Resiliencia en la Nube de AWS.

política basada en recursos

Una política asociada a un recurso, como un bucket de Amazon S3, un punto de conexión o una clave de cifrado. Este tipo de política especifica a qué entidades principales se les permite el acceso, las acciones compatibles y cualquier otra condición que deba cumplirse.

matriz responsable, confiable, consultada e informada (RACI)

Una matriz que define las funciones y responsabilidades de todas las partes involucradas en las actividades de migración y las operaciones de la nube. El nombre de la matriz se deriva de los tipos de responsabilidad definidos en la matriz: responsable (R), contable (A), consultado (C) e informado (I). El tipo de soporte (S) es opcional. Si incluye el soporte, la matriz se denomina matriz RASCI y, si la excluye, se denomina matriz RACI.

control receptivo

Un control de seguridad que se ha diseñado para corregir los eventos adversos o las desviaciones con respecto a su base de seguridad. Para obtener más información, consulte Controles receptivos en Implementación de controles de seguridad en AWS.

retain

Consulte Las 7 R.

retirar

Consulte Las 7 R.

Generación aumentada de recuperación (RAG)

Tecnología de IA generativa mediante la que un LLM hace referencia a un origen de datos autorizado que se encuentra fuera de sus orígenes de datos de entrenamiento antes de generar una respuesta. Por ejemplo, un modelo de RAG podría hacer una búsqueda semántica en la base de conocimientos o en los datos personalizados de una organización. Para más información, consulte ¿Qué es RAG (generación aumentada por recuperación)?

rotación

Proceso mediante el que periódicamente se actualiza un secreto para que resulte más difícil que un atacante pueda acceder a las credenciales.

control de acceso por filas y columnas (RCAC)

El uso de expresiones SQL básicas y flexibles que tienen reglas de acceso definidas. El RCAC consta de permisos de fila y máscaras de columnas.

RPO

Consulte objetivo de punto de recuperación.

RTO

Consulte objetivo de tiempo de recuperación.

manual de procedimientos

Conjunto de procedimientos manuales o automatizados necesarios para realizar una tarea específica. Por lo general, se diseñan para agilizar las operaciones o los procedimientos repetitivos con altas tasas de error.

S

SAML 2.0

Un estándar abierto que utilizan muchos proveedores de identidad (IdPs). Esta función permite el inicio de sesión único (SSO) federado, de modo que los usuarios pueden iniciar sesión Consola de administración de AWS o llamar a las operaciones de la AWS API sin tener que crear un usuario en IAM para todos los miembros de la organización. Para obtener más información sobre la federación basada en SAML 2.0, consulte Acerca de la federación basada en SAML 2.0 en la documentación de IAM.

SCADA

Consulte control de supervisión y adquisición de datos.

SCP

Consulte política de control de servicio.

secreta

En AWS Secrets Manager, información confidencial o restringida, como una contraseña o credenciales de usuario, que se almacena de forma cifrada. Se compone del valor del secreto y de sus metadatos. El valor del secreto puede ser binario, una sola cadena o varias cadenas. Para más información, consulte What's in a Secrets Manager secret? en la documentación de Secrets Manager.

seguridad desde el diseño

Enfoque de ingeniería de sistemas que tiene en cuenta la seguridad durante todo el proceso de desarrollo.

control de seguridad

Barrera de protección técnica o administrativa que impide, detecta o reduce la capacidad de un agente de amenazas para aprovechar una vulnerabilidad de seguridad. Existen cuatro tipos de controles de seguridad principales: preventivos, de detección, de respuesta y proactivos.

refuerzo de la seguridad

Proceso de reducir la superficie expuesta a ataques para hacerla más resistente a los ataques. Esto puede incluir acciones, como la eliminación de los recursos que ya no se necesitan, la implementación de prácticas recomendadas de seguridad consistente en conceder privilegios mínimos o la desactivación de características innecesarias en los archivos de configuración.

sistema de información sobre seguridad y administración de eventos (SIEM)

Herramientas y servicios que combinan sistemas de administración de información sobre seguridad (SIM) y de administración de eventos de seguridad (SEM). Un sistema de SIEM recopila, monitorea y analiza los datos de servidores, redes, dispositivos y otras fuentes para detectar amenazas y brechas de seguridad y generar alertas.

automatización de la respuesta de seguridad

Acción predefinida y programada que está diseñada para responder automáticamente a un evento de seguridad o corregirlo. Estas automatizaciones sirven como controles de seguridad preventivos o adaptables que le ayudan a implementar las mejores prácticas AWS de seguridad. La modificación de un grupo de seguridad de VPC, la aplicación de revisiones a una instancia de Amazon EC2 o la rotación de credenciales son algunos ejemplos de acciones de respuesta automatizadas.

cifrado del servidor

Cifrado de los datos en su destino, por parte de Servicio de AWS quien los recibe.

política de control de servicio (SCP)

Una política que proporciona un control centralizado de los permisos de todas las cuentas de una organización en AWS Organizations. Las SCP definen barreras de protección o establecen límites a las acciones que un administrador puede delegar en los usuarios o roles. Puede utilizar las SCP como listas de permitidos o rechazados, para especificar qué servicios o acciones se encuentra permitidos o prohibidos. Para obtener más información, consulte las políticas de control del servicio en la AWS Organizations documentación.

punto de enlace de servicio

La URL del punto de entrada de un Servicio de AWS. Para conectarse mediante programación a un servicio de destino, puede utilizar un punto de conexión. Para obtener más información, consulte Puntos de conexión de Servicio de AWS en Referencia general de AWS.

acuerdo de nivel de servicio (SLA)

Acuerdo que aclara lo que un equipo de TI se compromete a ofrecer a los clientes, como el tiempo de actividad y el rendimiento del servicio.

indicador de nivel de servicio (SLI)

Medición de un aspecto del rendimiento de un servicio, como la tasa de errores, la disponibilidad o el rendimiento.

objetivo de nivel de servicio (SLO)

Métrica objetivo que representa el estado de un servicio medido mediante un indicador de nivel de servicio.

modelo de responsabilidad compartida

Un modelo que describe la responsabilidad con AWS la que compartes la seguridad y el cumplimiento de la nube. AWS es responsable de la seguridad de la nube, mientras que usted es responsable de la seguridad en la nube. Para obtener más información, consulte el Modelo de responsabilidad compartida.

Shadow AI

Aplicaciones de IA no autorizadas creadas o utilizadas fuera de los canales regulados dentro de una organización.

SIEM

Consulte sistema de administración de eventos e información de seguridad.

único punto de error (SPOF)

Error en un único componente crítico de una aplicación que puede interrumpir el sistema.

SLA

Consulte acuerdo de nivel de servicio.

SLI

Consulte indicador de nivel de servicio.

SLO

Consulte objetivo de nivel de servicio.

modelo de dividir y sembrar

Un patrón para escalar y acelerar los proyectos de modernización. A medida que se definen las nuevas funciones y los lanzamientos de los productos, el equipo principal se divide para crear nuevos equipos de productos. Esto ayuda a ampliar las capacidades y los servicios de su organización, mejora la productividad de los desarrolladores y apoya la innovación rápida. Para más información, consulte Phased approach to modernizing applications in the Nube de AWS.

SPOF

Consulte único punto de error.

esquema en estrella

Estructura organizativa de una base de datos que utiliza una tabla de hechos de gran tamaño para almacenar datos transaccionales o medidos y una o varias tablas dimensionales más pequeñas para almacenar los atributos de los datos. Esta estructura está diseñada para utilizarse en un almacén de datos o con fines de inteligencia empresarial.

patrón de higo estrangulador

Un enfoque para modernizar los sistemas monolíticos mediante la reescritura y el reemplazo gradual de las funciones del sistema hasta que se pueda desmantelar el sistema heredado. Este patrón utiliza la analogía de una higuera que crece hasta convertirse en un árbol estable y, finalmente, se apodera y reemplaza a su host. El patrón fue presentado por Martin Fowler como una forma de gestionar el riesgo al reescribir sistemas monolíticos. Para ver un ejemplo de cómo aplicar este patrón, consulte Modernización gradual de los servicios web antiguos de Microsoft ASP.NET (ASMX) mediante contenedores y Amazon API Gateway.

subred

Un intervalo de direcciones IP en la VPC. Una subred debe residir en una sola zona de disponibilidad.

control de supervisión y adquisición de datos (SCADA)

En el sector de la fabricación, sistema que utiliza hardware y software para supervisar los activos físicos y las operaciones de producción.

cifrado simétrico

Un algoritmo de cifrado que utiliza la misma clave para cifrar y descifrar los datos.

pruebas sintéticas

Prueba de un sistema de manera que simule las interacciones de los usuarios para detectar posibles problemas o supervisar el rendimiento. Puede usar Amazon CloudWatch Synthetics para crear estas pruebas.

petición del sistema

Técnica para proporcionar contexto, instrucciones o pautas a un LLM para dirigir su comportamiento. Las peticiones del sistema ayudan a establecer el contexto y las reglas para las interacciones con los usuarios.

T

etiquetas

Key-value pares que actúan como metadatos para organizar sus AWS recursos. Las etiquetas pueden ayudar a administrar, identificar, organizar, buscar y filtrar recursos de . Para obtener más información, consulte Etiquetado de los recursos de AWS.

variable de destino

El valor que intenta predecir en el ML supervisado. Esto también se conoce como variable de resultado. Por ejemplo, en un entorno de fabricación, la variable objetivo podría ser un defecto del producto.

lista de tareas

Herramienta que se utiliza para hacer un seguimiento del progreso mediante un manual de procedimientos. La lista de tareas contiene una descripción general del manual de procedimientos y una lista de las tareas generales que deben completarse. Para cada tarea general, se incluye la cantidad estimada de tiempo necesario, el propietario y el progreso.

entorno de prueba

Consulte entorno.

entrenamiento

Proporcionar datos de los que pueda aprender su modelo de ML. Los datos de entrenamiento deben contener la respuesta correcta. El algoritmo de aprendizaje encuentra patrones en los

datos de entrenamiento que asignan los atributos de los datos de entrada al destino (la respuesta que desea predecir). Genera un modelo de ML que captura estos patrones. Luego, el modelo de ML se puede utilizar para obtener predicciones sobre datos nuevos para los que no se conoce el destino.

herramienta

Una función o API que un agente puede invocar para realizar operaciones en sistemas externos.

puerta de enlace de tránsito

Centro de tránsito de red que puede utilizar para interconectar las VPC y las redes en las instalaciones. Para obtener más información, consulte Qué es una pasarela de tránsito en la AWS Transit Gateway documentación.

flujo de trabajo basado en enlaces troncales

Un enfoque en el que los desarrolladores crean y prueban características de forma local en una rama de característica y, a continuación, combinan esos cambios en la rama principal. Luego, la rama principal se adapta a los entornos de desarrollo, preproducción y producción, de forma secuencial.

acceso de confianza

Otorgar permisos a un servicio que especifique para realizar tareas en su organización AWS Organizations y en sus cuentas en su nombre. El servicio de confianza crea un rol vinculado al servicio en cada cuenta, cuando ese rol es necesario, para realizar las tareas de administración por usted. Para obtener más información, consulte AWS Organizations Utilización con otros AWS servicios en la AWS Organizations documentación.

ajuste

Cambiar aspectos de su proceso de formación a fin de mejorar la precisión del modelo de ML. Por ejemplo, puede entrenar el modelo de ML al generar un conjunto de etiquetas, incorporar etiquetas y, luego, repetir estos pasos varias veces con diferentes ajustes para optimizar el modelo.

equipo de dos pizzas

Un DevOps equipo pequeño al que puedes alimentar con dos pizzas. Un equipo formado por dos integrantes garantiza la mejor oportunidad posible de colaboración en el desarrollo de software.

U

incertidumbre

Un concepto que hace referencia a información imprecisa, incompleta o desconocida que puede socavar la fiabilidad de los modelos predictivos de ML. Hay dos tipos de incertidumbre: la incertidumbre epistémica se debe a datos limitados e incompletos, mientras que la incertidumbre aleatoria se debe al ruido y la aleatoriedad inherentes a los datos.

tareas indiferenciadas

También conocido como tareas arduas, es el trabajo que es necesario para crear y operar una aplicación, pero que no proporciona un valor directo al usuario final ni proporciona una ventaja competitiva. Algunos ejemplos de tareas indiferenciadas son la adquisición, el mantenimiento y la planificación de la capacidad.

entornos superiores

Consulte entorno.

V

succión

Una operación de mantenimiento de bases de datos que implica limpiar después de las actualizaciones incrementales para recuperar espacio de almacenamiento y mejorar el rendimiento.

control de versión

Procesos y herramientas que realizan un seguimiento de los cambios, como los cambios en el código fuente de un repositorio.

Emparejamiento de VPC

Conexión entre dos VPC que permite enrutar el tráfico mediante direcciones IP privadas. Para obtener más información, consulte ¿Qué es una interconexión de VPC? en la documentación de Amazon VPC.

vulnerabilidad

Defecto de software o hardware que pone en peligro la seguridad del sistema.

W

caché caliente

Un búfer caché que contiene datos actuales y relevantes a los que se accede con frecuencia. La instancia de base de datos puede leer desde la caché del búfer, lo que es más rápido que leer desde la memoria principal o el disco.

datos templados

Datos a los que el acceso es infrecuente. Al consultar este tipo de datos, normalmente se aceptan consultas moderadamente lentas.

función de ventana

Función SQL que hace un cálculo en un grupo de filas que se relacionan de alguna manera con el registro actual. Las funciones de ventana son útiles para las tareas de procesamiento, como calcular una media móvil o acceder al valor de las filas en función de la posición relativa de la fila actual.

carga de trabajo

Conjunto de recursos y código que ofrece valor comercial, como una aplicación orientada al cliente o un proceso de backend.

flujo de trabajo

Grupos funcionales de un proyecto de migración que son responsables de un conjunto específico de tareas. Cada flujo de trabajo es independiente, pero respalda a los demás flujos de trabajo del proyecto. Por ejemplo, el flujo de trabajo de la cartera es responsable de priorizar las aplicaciones, planificar las oleadas y recopilar los metadatos de migración. El flujo de trabajo de la cartera entrega estos recursos al flujo de trabajo de migración, que luego migra los servidores y las aplicaciones.

WORM

Consulte escritura única y lectura múltiple.

WQF

Consulte AWS Workload Qualification Framework.

escritura única y lectura múltiple (WORM)

Modelo de almacenamiento que escribe los datos una sola vez y evita que se eliminen o modifiquen. Los usuarios autorizados pueden leer los datos tantas veces como sea necesario, pero no los pueden cambiar. Esta infraestructura de almacenamiento de datos se considera inmutable.

Z

ataque de día cero

Ataque, normalmente de malware, que se aprovecha de una vulnerabilidad de día cero.

vulnerabilidad de día cero

Un defecto o una vulnerabilidad sin mitigación en un sistema de producción. Los agentes de amenazas pueden usar este tipo de vulnerabilidad para atacar el sistema. Los desarrolladores suelen darse cuenta de la vulnerabilidad a raíz del ataque.

peticiones desde cero

Proporcionar a un LLM instrucciones para llevar a cabo una tarea, pero sin ejemplos (pasos) que puedan ayudar a guiarlo. El LLM debe usar los conocimientos del entrenamiento previo para llevar a cabo la tarea. La eficacia de la petición desde cero depende de la complejidad de la tarea y de la calidad de la petición. Consulte también peticiones con pocos pasos.

aplicación zombi

Aplicación que utiliza un promedio de CPU y memoria menor al 5 por ciento. En un proyecto de migración, es habitual retirar estas aplicaciones.

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.