Fundamentos de la IA agencial en AWS

AWS Guía prescriptiva

AWS Guía prescriptiva: Fundamentos de la IA agencial en AWS

Table of Contents

Fundamentos de la IA agencial en AWS

Aaron Sempf, Amazon Web Services

Julio de 2025 (historial del documento)

En un mundo de sistemas cada vez más inteligentes, distribuidos y autónomos, el concepto de agente —una entidad que puede percibir su entorno, razonar sobre su estado y actuar con intención — se ha convertido en algo fundamental. Los agentes no son simplemente programas que ejecutan instrucciones; son entidades orientadas a objetivos y sensibles al contexto que toman decisiones en nombre de los usuarios, los sistemas o las organizaciones. Su aparición refleja un cambio en la forma de crear y concebir el software: un cambio de la lógica procedimental y la automatización reactiva a sistemas que funcionan con autonomía y propósito.

En la intersección de la IA, los sistemas distribuidos y la ingeniería de software se encuentra un poderoso paradigma conocido como IA agencial. Esta nueva generación de sistemas inteligentes está formada por agentes de software que son capaces de adoptar un comportamiento adaptativo, realizar una coordinación compleja y delegar la toma de decisiones.

Esta guía presenta los principios que definen a los agentes de software modernos y describe su evolución hacia la IA agencial. Para explicar este cambio, la guía proporciona los antecedentes conceptuales y, a continuación, describe la evolución de los agentes de software hasta la IA de los agentes:

- Introducción a los agentes de software define a los agentes de software, los compara con los componentes de software tradicionales e introduce las características esenciales que diferencian el comportamiento de los agentes de la automatización tradicional basándose en marcos establecidos.
- El propósito de los agentes de software examina por qué existen los agentes de software, qué funciones desempeñan, qué problemas resuelven y cómo permiten la delegación inteligente, reducen la carga cognitiva y fomentan el comportamiento adaptativo en entornos dinámicos.
- La evolución de los agentes de software traza los hitos intelectuales y tecnológicos que dieron forma a los agentes de software, desde los primeros conceptos de autonomía y simultaneidad hasta la aparición de los sistemas multiagente y las arquitecturas de agentes formales, que desembocaron en la convergencia con la IA generativa.
- De los agentes de software a la IA de agencia se presenta la IA agencial como la culminación de décadas de progreso, que combina modelos de agentes distribuidos con modelos básicos,

computación sin servidor y protocolos de orquestación. En esta sección, se describe cómo esta convergencia posibilita una nueva generación de agentes inteligentes que utilizan herramientas y que funcionan con autonomía, asincronicidad y una verdadera capacidad de acción a escala.

Destinatarios previstos

Esta guía está diseñada para arquitectos, desarrolladores y líderes tecnológicos que desean comprender la historia, los conceptos principales y la evolución de los agentes de software hasta convertirse en una IA agente antes de adoptar esta tecnología para las soluciones de nube modernas. AWS

Objetivos

La adopción de arquitecturas de agencia ayuda a las organizaciones a:

- Acelere la generación de valor: automatice y escale el trabajo basado en el conocimiento y reduzca el esfuerzo manual y la latencia.
- Mejore la participación de los clientes: ofrezca asistentes inteligentes en todos los dominios.
- Reduzca los costos operativos: automatice los flujos de decisiones que antes requerían la participación o la supervisión de una persona.
- Impulse la innovación y la diferenciación: cree productos inteligentes que se adapten, aprendan y compitan en tiempo real.
- Modernice los flujos de trabajo tradicionales: rediseñe los scripts y los monolitos para convertirlos en agentes de razonamiento modulares.

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

Introducción a los agentes de software

El concepto de agente de software ha evolucionado considerablemente desde sus orígenes en las entidades autónomas en la década de 1960 hasta su exploración formal a principios de la década de 1990. A medida que los sistemas digitales se vuelven cada vez más complejos (desde scripts deterministas hasta aplicaciones adaptables e inteligentes), los agentes de software se han convertido en componentes esenciales para permitir un comportamiento autónomo, sensible al contexto y orientado a objetivos en los sistemas informáticos. En el contexto de las arquitecturas nativas de la nube y mejoradas con la IA, especialmente con la llegada de la IA generativa, los grandes modelos de lenguaje (LLMs) y plataformas como Amazon Bedrock, los agentes de software se están redefiniendo desde una nueva perspectiva de capacidad y escala.

Esta introducción se basa en la obra fundamental Software Agents: An Overview de Hyacinth S. Nwana (Nwana 1996). Define a los agentes de software, analiza sus raíces conceptuales y amplía la discusión a un marco contemporáneo para definir tres principios generales de los agentes de software modernos: autonomía, asincronicidad y agencia. Estos principios distinguen a los agentes de software de otros tipos de servicios o aplicaciones y les permiten operar con determinación, resiliencia e inteligencia en entornos distribuidos y en tiempo real.

En esta sección

Desde la autonomía hasta la inteligencia distribuida

Antes de que el término agente de software se generalizara, las primeras investigaciones informáticas exploraron la idea de las entidades digitales autónomas, que son sistemas que son capaces de actuar de forma independiente, reaccionar a las entradas y tomar decisiones en función de normas u objetivos internos. Estas primeras ideas sentaron las bases conceptuales de lo que se convertiría en el paradigma de los agentes. (Para obtener una cronología histórica, consulte la sección La evolución de los agentes de software, más adelante en esta guía).

Conceptos iniciales de autonomía

La noción de máquinas o programas que actúan independientemente de los operadores humanos ha intrigado a los diseñadores de sistemas durante décadas. Los primeros trabajos sobre cibernética, inteligencia artificial y sistemas de control examinaron cómo el software podía mostrar un comportamiento autorregulado, responder de forma dinámica a los cambios y funcionar sin supervisión humana continua.

Estas ideas introdujeron la autonomía como un atributo fundamental de los sistemas inteligentes y sentaron las bases para la aparición de un software capaz de decidir y actuar, en lugar de limitarse a reaccionar o ejecutar.

El modelo de actor y la ejecución asíncrona

En la década de 1970, el modelo actor, que se introdujo en el artículo A Universal Modular ACTOR Formalism for Artificial Intelligence (Hewitt et al. 1973), proporcionó un marco formal para pensar en la computación descentralizada y basada en mensajes. En este modelo, los actores son entidades independientes que se comunican exclusivamente mediante la transmisión de mensajes asíncronos y permiten sistemas escalables, concurrentes y tolerantes a errores.

El modelo de actor hizo hincapié en tres atributos clave que siguen influyendo en el diseño moderno de los agentes:

- Aislamiento del estado y el comportamiento
- Interacción asincrónica entre entidades
- Creación y delegación dinámicas de tareas

Estos atributos se alineaban con las necesidades de los sistemas distribuidos y prefiguraban las características operativas de los agentes de software en entornos nativos de la nube.

Sistemas multiagente e inteligencia distribuida

A medida que los sistemas informáticos se volvieron más interconectados después de la década de 1960, los investigadores exploraron la inteligencia artificial distribuida (DAI). Este campo se centró en cómo varias entidades autónomas podían trabajar de forma colaborativa o competitiva en un sistema. La DAI llevó al desarrollo de sistemas multiagentes, en los que cada agente tiene objetivos, percepciones y razonamientos locales, pero también opera en un entorno más amplio e interconectado.

Esta visión de la inteligencia distribuida, en la que la toma de decisiones está descentralizada y el comportamiento emergente surge de la interacción de los agentes, sigue siendo fundamental para concebir y construir los sistemas modernos basados en agentes.

La tipología de Nwana y el auge de los agentes de software

La formalización del concepto de agente de software a mediados de la década de 1990 marcó un punto de inflexión en la evolución de los sistemas inteligentes. Entre las contribuciones más influyentes a esta formalización se encuentra el artículo fundamental de Hyacinth S. Nwana, Software Agents: An Overview (Nwana 1996), que proporcionó uno de los primeros marcos integrales para categorizar y comprender los agentes de software en varias dimensiones.

En este paper, Nwana analiza el estado de la investigación de los agentes de software e identifica una divergencia creciente en la forma en que se definían e implementaban los agentes. El paper destaca la necesidad de un marco conceptual común y propone una tipología que clasifica a los agentes según sus capacidades clave. Analiza los sistemas de agentes representativos del mundo académico y de la industria, distingue los agentes de los programas y objetos tradicionales y describe los desafíos y las oportunidades de la computación basada en agentes.

Nwana hace hincapié en que los agentes de software no son un concepto monolítico, sino que existen en un espectro de sofisticación y capacidad. La tipología sirve para aclarar este panorama y guiar el diseño y la investigación futuros.

Nwana define un agente de software como una entidad de software que funciona de forma continua y autónoma en un entorno determinado, que a menudo está habitado por otros agentes y procesos. Esta definición hace hincapié en dos características clave:

- Continuidad: el agente opera de forma persistente en el tiempo, sin requerir una intervención humana constante.
- Autonomía: el agente tiene la capacidad de tomar decisiones y actuar en consecuencia de ellas de forma independiente, en función de su percepción del entorno.

Esta definición, combinada con la tipología de agentes de Nwana, hace hincapié en la autoridad delegada (a través de la autonomía) y la proactividad como características fundamentales de los agentes. Distingue entre agentes y subrutinas o servicios al destacar la capacidad del agente para actuar de forma independiente en nombre de otra entidad e iniciar un comportamiento que persiga objetivos, en lugar de responder únicamente a órdenes directas.

La tipología de agentes de Nwana

Para diferenciar aún más entre los distintos tipos de agentes, Nwana presenta un sistema de clasificación basado en seis atributos clave:

- Autonomía: el agente opera sin la intervención directa de personas u otras personas.
- Habilidad social: el agente interactúa con otros agentes o humanos mediante el uso de mecanismos de comunicación.
- Reactividad: el agente percibe su entorno y responde de manera oportuna.
- Proactividad: el agente muestra un comportamiento orientado a un objetivo al tomar la iniciativa.
- Adaptabilidad y aprendizaje: el agente mejora su rendimiento con el tiempo gracias a la experiencia.
- Movilidad: el agente puede moverse entre distintos entornos de sistemas o redes.

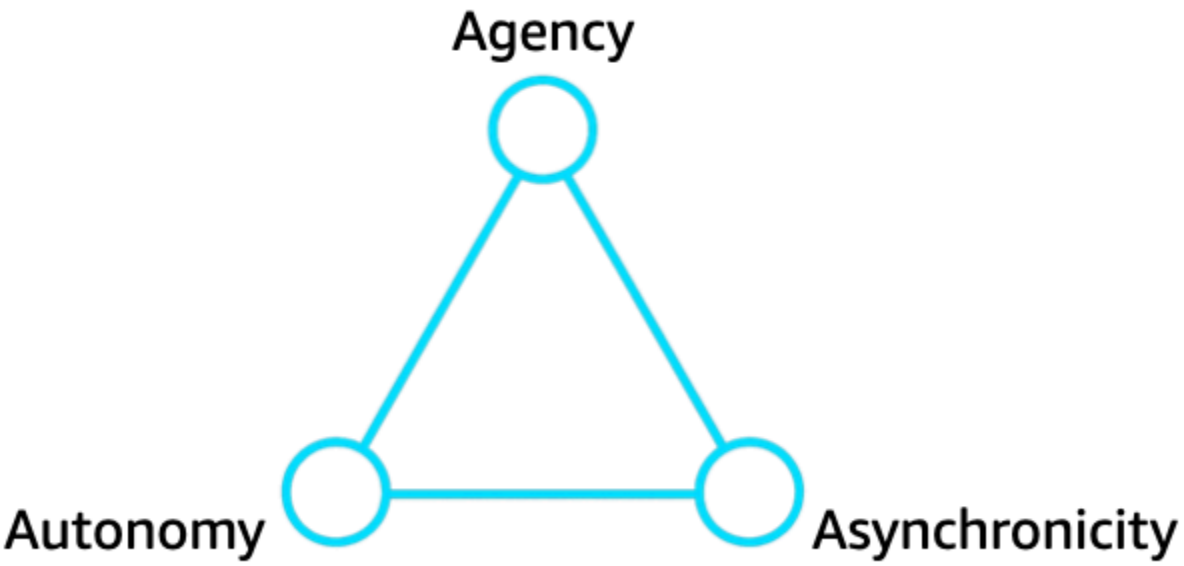
Desde la tipología hasta los principios de los agentes modernos

El trabajo de Nwana sirvió tanto de taxonomía como de lente fundamental a través de la cual la comunidad informática pudo evaluar las formas cambiantes de agencia en el software. Su énfasis en la autonomía, la proactividad y el concepto de actuar en nombre de un usuario o sistema sentó las bases para lo que ahora consideramos un comportamiento agencial.

Si bien las tecnologías y los entornos han cambiado, especialmente con el auge de la IA generativa, la infraestructura sin servidores y los marcos de orquestación multiagente, las ideas fundamentales del trabajo de Nwana siguen siendo relevantes. Proporcionan un puente fundamental entre la teoría primitiva de los agentes y los tres pilares modernos de los agentes de software.

Los tres pilares de los agentes de software modernos

En el contexto de las plataformas actuales impulsadas por la IA, las arquitecturas de microservicios y los sistemas basados en eventos, los agentes de software se pueden definir mediante tres principios interdependientes que los distinguen de los servicios estándar o los scripts de automatización: autonomía, asincronicidad y agencia. En la siguiente ilustración y en los diagramas siguientes, el triángulo representa estos tres pilares de los agentes de software modernos.



Autonomía

Los agentes modernos operan de forma independiente. Toman decisiones basadas en el estado interno y el contexto ambiental sin requerir indicaciones humanas. Esto les permite reaccionar ante los datos en tiempo real, gestionar su propio ciclo de vida y ajustar su comportamiento en función de los objetivos y los datos de la situación.

La autonomía es la base del comportamiento de los agentes. Permite a los agentes funcionar sin supervisión continua ni flujos de control codificados.

Asincronía

Los agentes son fundamentalmente asíncronos. Esto significa que responden a los eventos, señales y estímulos a medida que se producen, sin depender del bloqueo de llamadas o de flujos de trabajo lineales. Esta característica permite una comunicación escalable y sin bloqueos, una capacidad de respuesta en entornos distribuidos y un acoplamiento flexible entre los componentes.

Gracias a la asincronicidad, los agentes pueden participar en los sistemas en tiempo real y coordinarse con otros servicios o agentes de forma fluida y eficiente.

La agencia como principio definitorio

La autonomía y la asincronía son necesarias, pero estas características por sí solas no son suficientes para convertir un sistema en un verdadero agente de software. El diferenciador fundamental es la agencia, que introduce:

- Comportamiento dirigido a objetivos: los agentes persiguen objetivos y evalúan el progreso hacia ellos.
- Toma de decisiones: los agentes evalúan las opciones y eligen las acciones en función de las reglas, los modelos o las políticas aprendidas.
- Intención delegada: los agentes actúan en nombre de una persona, un sistema u organización y tienen un propósito incorporado.
- Razonamiento contextual: los agentes incorporan la memoria o los modelos de su entorno para guiar el comportamiento de forma inteligente.

Un sistema autónomo y asíncrono podría seguir siendo un servicio reactivo. Lo que lo convierte en un agente de software es su capacidad de actuar con intención y propósito, de ser agente.

Agencia con propósito

Los principios de autonomía, asincronicidad y agencia permiten que los sistemas funcionen de forma inteligente, adaptativa e independiente en entornos distribuidos. Estos principios se basan en décadas de evolución conceptual y arquitectónica, y ahora sustentan muchos de los sistemas de IA más avanzados que se están creando en la actualidad.

En esta nueva era de IA generativa, orquestación orientada a objetivos y colaboración entre múltiples agentes, es esencial entender qué hace que un agente de software sea verdaderamente agente. Reconocer la agencia como la característica que nos define nos ayuda a ir más allá de la automatización y adentrarnos en el ámbito de la inteligencia autónoma con un propósito.

El propósito de los agentes de software

A medida que los sistemas modernos se han vuelto cada vez más complejos, distribuidos e inteligentes, el papel de los agentes de software ha ido ganando protagonismo en ámbitos que van desde las operaciones autónomas hasta las tecnologías de asistencia al usuario. Pero, ¿cuál es el propósito subyacente de los agentes de software? ¿Por qué diseñamos sistemas que van más allá de los scripts, los servicios o los modelos estáticos y, en cambio, delegamos tareas en entidades que son capaces de percibir, razonar y actuar?

Esta sección explora el propósito fundamental de los agentes de software: permitir la delegación inteligente de tareas en entornos dinámicos, con un enfoque en la autonomía, la adaptabilidad y la acción decidida. Introduce los fundamentos conceptuales de los agentes de software, traza su estructura cognitiva y describe los problemas del mundo real para los que están preparados de forma única para resolver.

En esta sección

Del modelo actor a la cognición de los agentes

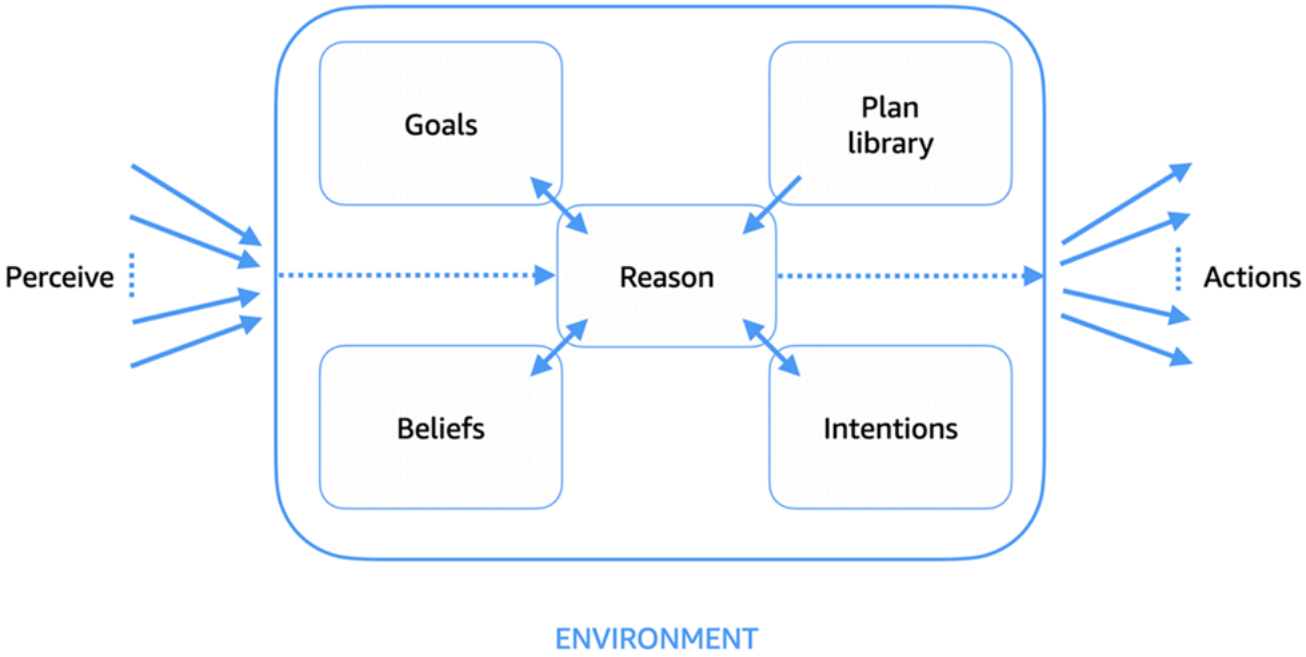
El propósito y la estructura de los agentes de software se basan en ideas que surgieron de los primeros modelos de computación, en particular el modelo actor que introdujo Carl Hewitt en la década de 1970 (Hewitt et al. 1973).

El modelo de actores trata la computación como un conjunto de entidades independientes que se ejecutan simultáneamente denominadas actores. Cada actor encapsula su propio estado, interactúa únicamente mediante el paso asincrónico de mensajes y puede crear nuevos actores y delegar tareas.

Este modelo proporcionó la base conceptual para el razonamiento, la reactividad y el aislamiento descentralizados, todos los cuales sustentan la arquitectura conductual de los agentes de software modernos.

La función del agente: percibir, razonar, actuar

En el centro de cada agente de software hay un ciclo cognitivo que a menudo se describe como el ciclo de percibir, razonar y actuar. Este proceso se ilustra en el siguiente diagrama. Define cómo los agentes operan de forma autónoma en entornos dinámicos.



- Percibir: los agentes recopilan información (por ejemplo, eventos, entradas de sensores o señales de API) del entorno y actualizan su estado interno o sus creencias.
- Motivo: los agentes analizan las creencias, los objetivos y el conocimiento contextual actuales mediante una biblioteca de planes o un sistema lógico. Este proceso puede implicar la priorización de objetivos, la resolución de conflictos o la selección de intenciones.
- Actuar: los agentes seleccionan y ejecutan acciones que los acercan a la consecución de las metas que les han sido delegadas.

Esta arquitectura apoya la capacidad de los agentes para funcionar más allá de una programación rígida y permite un comportamiento flexible, sensible al contexto y orientado a los objetivos. Forma el marco mental que guía los propósitos más amplios de los agentes de software.

Colaboración e intencionalidad autónomas

El objetivo de los agentes de software es aportar a la informática moderna la autonomía, el conocimiento del contexto y la delegación inteligente. Como los agentes se basan en los principios del modelo del actor y se encarnan en el ciclo de percibir, razonar y actuar, hacen posible que los sistemas no solo sean reactivos, sino también proactivos y decididos.

Los agentes permiten que el software decida, se adapte y actúe en entornos complejos. Representan a los usuarios, interpretan los objetivos e implementan las tareas a la velocidad de una máquina. A medida que nos adentramos en la era de la IA agencial, los agentes de software se están convirtiendo en la interfaz operativa entre la intención humana y la acción digital inteligente.

¿Delegar la intención

A diferencia de los componentes de software tradicionales, los agentes de software existen para actuar en nombre de otra cosa: un usuario, otro sistema o un servicio de nivel superior. Tienen una intención delegada, lo que significa que:

- Operan de forma independiente después de la iniciación.
- Tome decisiones que estén alineadas con los objetivos de la persona que delega.
- Supere la incertidumbre y las compensaciones en la ejecución.

Los agentes cierran la brecha entre las instrucciones y los resultados, lo que permite a los usuarios expresar su intención con un mayor nivel de abstracción en lugar de requerir instrucciones explícitas.

Operan en entornos dinámicos e impredecibles

Los agentes de software están diseñados para entornos en los que las condiciones cambian constantemente, los datos llegan en tiempo real y el control y el contexto están distribuidos.

A diferencia de los programas estáticos que requieren entradas exactas o una ejecución sincrónica, los agentes se adaptan a su entorno y responden de forma dinámica. Esta es una capacidad vital en la infraestructura nativa de la nube, la computación perimetral, las redes de Internet de las cosas (IoT) y los sistemas de toma de decisiones en tiempo real.

Reducir la carga cognitiva humana

Uno de los principales objetivos de los agentes de software es reducir la carga cognitiva y operativa de los seres humanos. Los agentes pueden:

- Supervise continuamente los sistemas y los flujos de trabajo.
- Detecte y responda a condiciones predefinidas o emergentes.
- Automatice las decisiones repetitivas y de gran volumen.
- Reaccione a los cambios del entorno con una latencia mínima.

Cuando la toma de decisiones pasa de los usuarios a los agentes, los sistemas se vuelven más receptivos, resilientes y centrados en las personas, y pueden adaptarse en tiempo real a la nueva información o a las interrupciones. Esto permite una respuesta más rápida, así como una mayor continuidad operativa en entornos de alta complejidad o gran escala. El resultado es un cambio en el enfoque humano, pasando de la toma de decisiones a nivel microscópico a la supervisión estratégica y la resolución creativa de problemas.

Habilitar la inteligencia distribuida

La capacidad de los agentes de software para operar de forma individual o colectiva permite diseñar sistemas multiagente (MAS) que se coordinan entre entornos u organizaciones. Estos sistemas pueden distribuir las tareas de forma inteligente y negociar, cooperar o competir para lograr objetivos compuestos.

Por ejemplo, en un sistema de cadena de suministro global, los agentes individuales administran las fábricas, el transporte marítimo, los almacenes y las entregas de última milla. Cada agente opera con autonomía local: los agentes de fábrica optimizan la producción en función de las limitaciones de recursos, los agentes de almacén ajustan los flujos de inventario en tiempo real y los agentes de entrega redirigen los envíos en función del tráfico y la disponibilidad de los clientes.

Estos agentes se comunican y coordinan de forma dinámica y se adaptan a las interrupciones, como los retrasos en los puertos o las averías de los camiones, sin un control centralizado. La inteligencia general del sistema surge de estas interacciones y permite una logística flexible y optimizada que va más allá de las capacidades de un solo componente.

En este modelo, los agentes actúan como nodos en un tejido de inteligencia más amplio. Forman sistemas emergentes que son capaces de resolver problemas que ningún componente podría resolver por sí solo.

Actuar con un propósito, no solo con una reacción

La automatización por sí sola es insuficiente en sistemas complejos. El propósito que define a un agente de software es actuar con un propósito y evaluar los objetivos, sopesar el contexto y

tomar decisiones informadas. Esto significa que los agentes de software persiguen sus objetivos en lugar de responder únicamente a los factores desencadenantes. Pueden revisar sus creencias e intenciones en función de la experiencia o los comentarios. En este contexto, las creencias se refieren a la representación interna del entorno por parte del agente (por ejemplo, «el paquete X está en el almacén A»), en función de sus percepciones (entrada y sensores). Las intenciones se refieren a los planes que el agente elige para alcanzar un objetivo (por ejemplo, «utilizar la ruta de entrega B y avisar al destinatario»). Los agentes también pueden escalar, aplazar o adaptar las acciones según sea necesario.

Esta intencionalidad es lo que hace que los agentes de software no sean solo ejecutores reactivos, sino colaboradores autónomos en sistemas inteligentes.

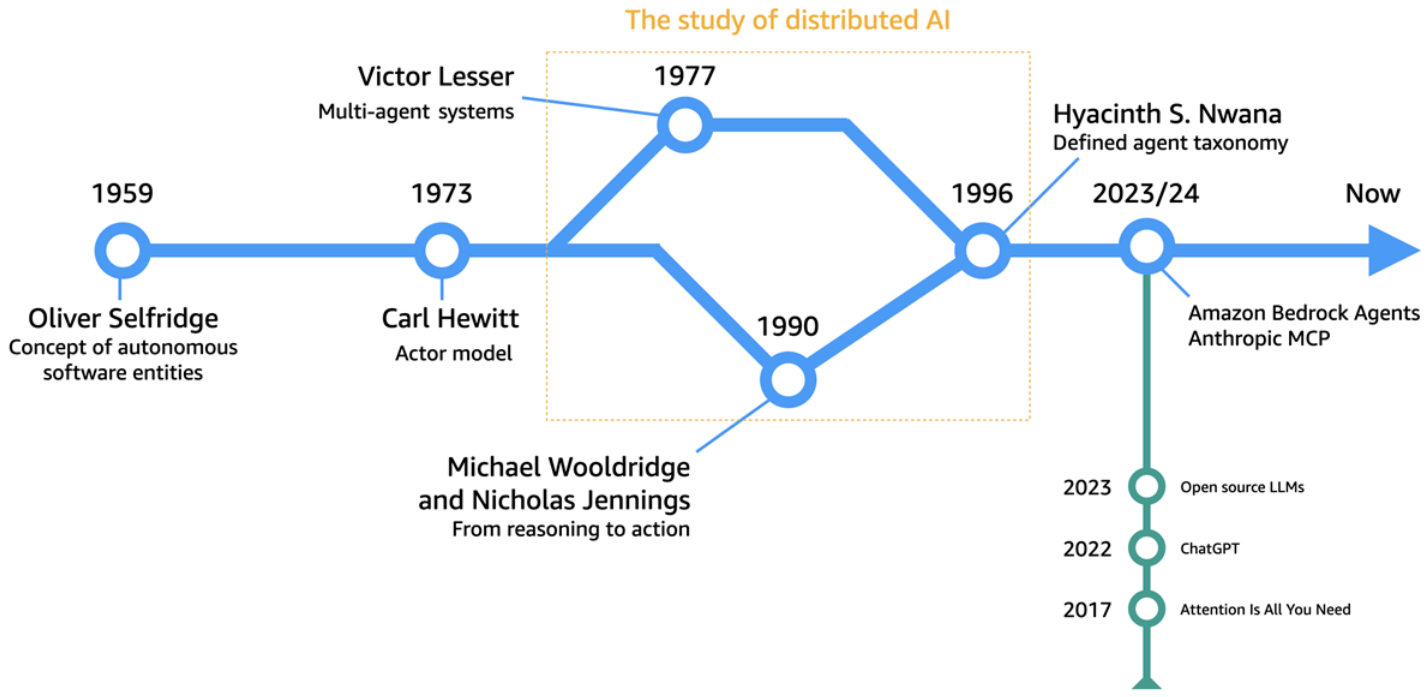
La evolución de los agentes de software

El paso de sistemas automatizados simples a agentes de software inteligentes, autónomos y orientados a objetivos refleja décadas de evolución en la informática, la inteligencia artificial y los sistemas distribuidos.

A esta evolución le siguió el auge del aprendizaje automático, que cambió el paradigma de las reglas artesanales al reconocimiento estadístico de patrones. Estos sistemas podían aprender de los datos y permitían avances en la percepción, la clasificación y la toma de decisiones.

Los modelos de lenguaje LLMs extensos () representan una convergencia de escala, arquitectura y aprendizaje no supervisado. LLMs pueden razonar, generar y adaptar tareas con poca o ninguna formación específica para cada tarea. Al combinarlos LLMs con una infraestructura escalable nativa de la nube y arquitecturas componibles, ahora estamos logrando la visión completa de la IA de los agentes: agentes de software inteligentes que pueden operar con autonomía, conocimiento del contexto y adaptabilidad a escala empresarial.

Esta sección explora la historia de los agentes de software desde la teoría fundamental hasta la práctica moderna, como se ilustra en el siguiente diagrama. En él se destaca la convergencia de la inteligencia artificial distribuida (DAI) y la IA generativa basada en transformadores, y se identifican los principales hitos que han dado forma al surgimiento de la IA de agentes.



En esta sección

Fundamentos de los agentes de software

1959 — Oliver Selfridge: el nacimiento de la autonomía en el software

Los orígenes de los agentes de software se remontan a Oliver Selfridge, quien introdujo el concepto de entidades de software autónomas (demonios), programas que son capaces de percibir su entorno y actuar de forma independiente (Selfridge 1959). Sus primeros trabajos sobre la percepción y el aprendizaje de las máquinas sentaron las bases filosóficas para las nociones futuras de los agentes como sistemas inteligentes e independientes.

1973: Carl Hewitt: el modelo actor

Un avance fundamental se produjo con el modelo actor de Carl Hewitt (Hewitt et al. 1973), que es un modelo computacional formal que describe a los agentes como entidades independientes y concurrentes. En este modelo, los agentes pueden encapsular su propio estado y comportamiento, comunicarse mediante el paso asincrónico de mensajes y crear otros actores de forma dinámica y delegarles tareas.

El modelo actor proporcionó tanto la base teórica como el paradigma arquitectónico para los sistemas distribuidos basados en agentes. Este modelo prefiguró implementaciones modernas de simultaneidad, como el lenguaje de programación Erlang y el marco Akka.

Madurar el campo: del razonamiento a la acción

1977 — Victor Lesser: sistemas multiagente

A finales de la década de 1970, surgió la inteligencia artificial distribuida (DAI). Fue promovida por Victor Lesser, quien es ampliamente reconocido por ser pionero en los sistemas multiagente (MAS). Su trabajo se centró en cómo las entidades de software independientes podían cooperar, coordinarse y negociar (consulte la sección de Recursos). Este desarrollo dio lugar a sistemas

que eran capaces de resolver problemas complejos de forma colectiva, lo que supuso un avance fundamental en la creación de inteligencia distribuida.

Década de 1990: Michael Wooldridge y Nicholas Jennings: el espectro de agentes

En la década de 1990, el campo de la inteligencia distribuida había madurado gracias a las contribuciones de investigadores como Michael Wooldridge y Nicholas Jennings. Estos estudiosos clasificaron a los agentes según un espectro, desde reactivos hasta deliberativos, desde sistemas no cognitivos hasta agentes razonadores impulsados por objetivos (Wooldridge y Jennings 1995). Su trabajo hacía hincapié en que los agentes ya no eran ideas abstractas, sino que se aplicaban en una amplia gama de ámbitos prácticos, desde la robótica hasta el software empresarial.

Estos investigadores también introdujeron un cambio de enfoque: del razonamiento centralizado a la acción distribuida. Los agentes ya no eran solo pensadores, sino actores que operaban en entornos en tiempo real con autonomía y propósito.

1996: Hyacinth S. Nwana: formalizando el concepto de agente

En 1996, Hyacinth S. Nwana publicó el influyente paper Software Agents: An Overview, que proporcionaba la clasificación de agentes más completa hasta la fecha. Su tipología incluía atributos como la autonomía, la capacidad social, la reactividad, la proactividad, el aprendizaje y la movilidad, y diferenciaba entre agentes de software y construcciones de software tradicionales.

Nwana también ofreció una definición ampliamente aceptada, parafraseada: un agente de software es un programa informático basado en software que actúa en nombre de un usuario u otro programa en una relación de agencia, que deriva de la noción de delegación.

Esta formalización fue fundamental para que los agentes de software pasaran de ser construcciones teóricas a aplicaciones del mundo real. Dio lugar a una generación de sistemas basados en agentes en campos como las telecomunicaciones, la automatización del flujo de trabajo y los asistentes inteligentes.

El trabajo de Nwana se sitúa en el punto de convergencia de las primeras investigaciones sobre la IA distribuida y las arquitecturas operativas de los agentes modernos. Es un puente crucial entre la teoría cognitiva de los agentes y su despliegue práctico en los sistemas actuales.

Una línea temporal paralela: el auge de los grandes modelos lingüísticos

Mientras los marcos de agentes evolucionaban, se estaba produciendo una revolución paralela y convergente en el procesamiento del lenguaje natural y el aprendizaje automático:

- 2017 — Transformers: The paper Attention Is All You Need (Vaswani et al. 2017) presentó la arquitectura de transformadores, que mejoró drásticamente la forma en que las máquinas procesan y generan el lenguaje.
- 2022 — ChatGPT: OpenAI lanzó una interfaz de chat para GPT-3.5 llamada ChatGPT, que permitía una conversación natural e interactiva con un sistema de IA de uso general.
- 2023 LLMs: código abierto: las versiones de Llama, Falcon y Mistral hicieron que los modelos potentes fueran ampliamente accesibles y aceleraron el desarrollo de marcos de agentes en entornos empresariales y de código abierto.

Estas innovaciones convirtieron los modelos lingüísticos en motores de razonamiento capaces de analizar el contexto, planificar acciones y encadenar respuestas, y LLMs se convirtieron en elementos clave de los agentes de software inteligentes.

Los plazos convergen: el surgimiento de la IA agencial

2023-2024: plataformas de agentes de nivel empresarial

La convergencia de las arquitecturas de agentes de software distribuidas y las basadas en transformadores LLMs culminó con el auge de la IA de los agentes.

- Amazon Bedrock Agents presentó una forma totalmente gestionada de crear agentes de software basados en objetivos y que utilizan herramientas mediante modelos básicos de Amazon Bedrock.
- El Model Context Protocol (MCP) de Anthropic definió un método para que los modelos lingüísticos de gran tamaño pudieran acceder a herramientas, entornos y memoria externos e interactuar con ellos. Esto es clave para el comportamiento contextual, persistente y autónomo.

Estos dos hitos representan la síntesis de la agencia y la inteligencia. Los agentes ya no estaban limitados a flujos de trabajo estáticos o a una automatización rígida. Ahora podían razonar en varios pasos, coordinarse con las herramientas y APIs mantener el estado contextual, y aprender y adaptarse con el tiempo.

Enero-junio de 2025: capacidades empresariales ampliadas

En la primera mitad de 2025, el panorama de la IA de las agencias se expandió significativamente con nuevas capacidades empresariales. En febrero de 2025, Anthropic lanzó el Claude 3.7 Sonnet, que fue el primer modelo de razonamiento híbrido del mercado, y la especificación MCP obtuvo una amplia adopción.

Los asistentes de codificación de IA, como Amazon Q Developer, Cursor y MCP WindSurf integrado, permiten estandarizar la generación de código, el análisis de repositorios y los flujos de trabajo de desarrollo. La versión de marzo de 2025 de MCP introdujo importantes funciones listas para la empresa, como la integración de la seguridad OAuth 2.1, la ampliación de los tipos de recursos para diversos tipos de acceso a los datos y las opciones de conectividad mejoradas a través de Streamable HTTP. Sobre esta base, AWS anunció en mayo de 2025 su incorporación al comité directivo del MCP y su contribución a las nuevas capacidades de comunicación. agent-to-agent Esto refuerza aún más la posición del protocolo como estándar de la industria para la interoperabilidad de la IA entre agencias.

En mayo de 2025, AWS reforzó las opciones de los clientes para crear flujos de trabajo de IA para agencias mediante el código abierto del marco Strands Agents. Este marco independiente del proveedor y del modelo permite a los desarrolladores utilizar modelos básicos en todas las plataformas y, al mismo tiempo, mantener una profunda integración de los servicios. AWS Como se destaca en el blog de código AWS abierto, Strands Agents sigue una filosofía de diseño basada en el modelo, que sitúa los modelos básicos en el centro de la inteligencia de los agentes. Esto facilita a los clientes la creación y el despliegue de agentes de IA sofisticados para sus casos de uso específicos.

Emergence: IA agencial

La evolución de los agentes de software, desde las primeras ideas de autonomía hasta la moderna orquestación basada en la LLM, ha sido prolongada y escalonada. Lo que comenzó con la visión de Oliver Selfridge de percibir los programas se ha convertido en un ecosistema sólido de agentes de software inteligentes, sensibles al contexto y orientados a objetivos, que pueden colaborar, adaptarse y razonar.

La convergencia de la inteligencia artificial distribuida (DAI) y la IA generativa basada en transformadores marca el comienzo de una nueva era en la que los agentes de software ya no son solo herramientas, sino actores autónomos en los sistemas inteligentes.

La IA de Agentic representa la próxima evolución en los sistemas de software. Proporciona una clase de agentes inteligentes que son autónomos, asíncronos y agenciales, y que pueden actuar con una intención delegada y operar con determinación en entornos dinámicos y distribuidos. Agentic AI unifica lo siguiente:

- El linaje arquitectónico de los sistemas multiagente y el modelo actor
- El modelo cognitivo de percibir, razonar, actuar
- El poder generador LLMs y los transformadores
- La flexibilidad operativa de la computación nativa de la nube y sin servidor

De agentes de software a IA agentic

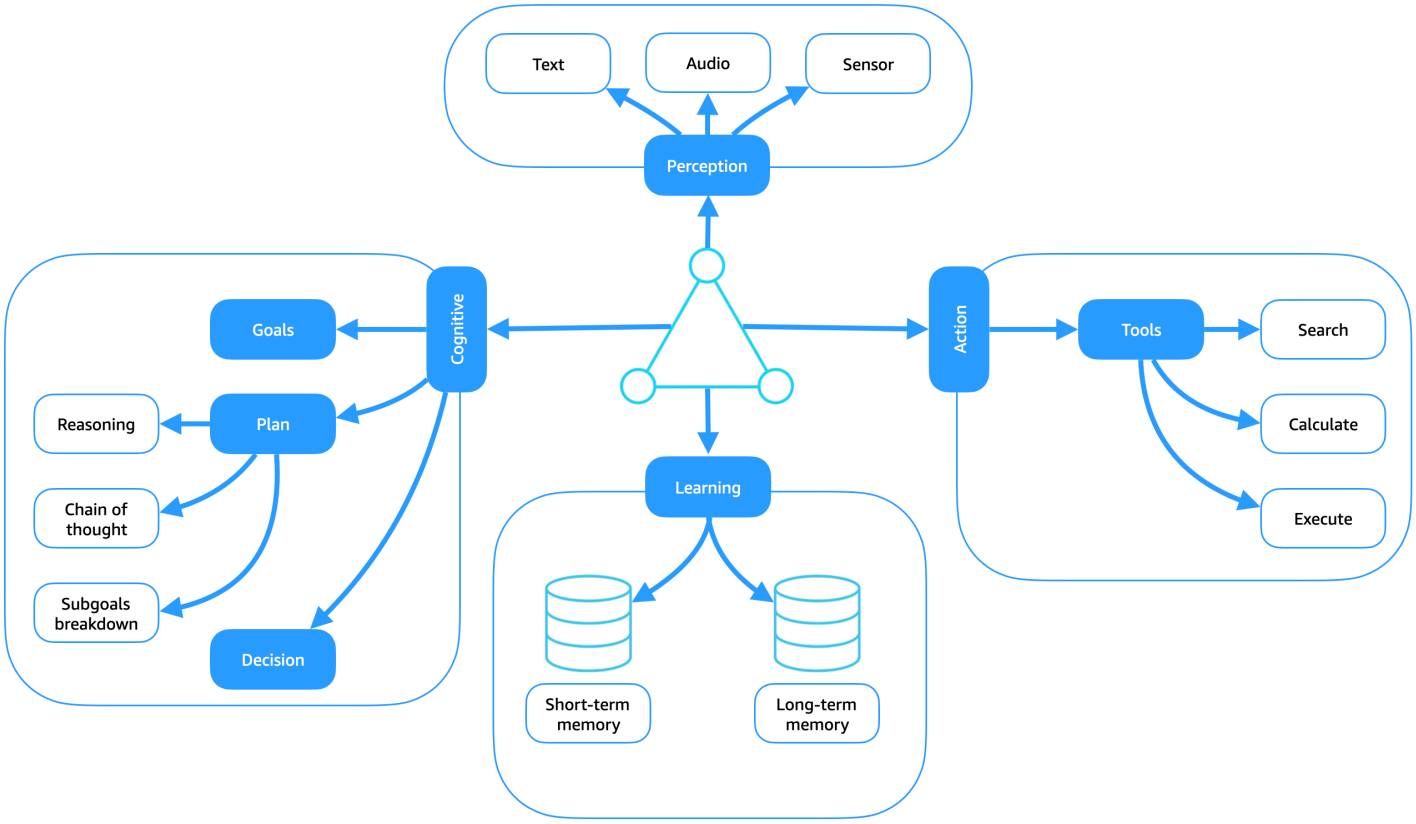
Los agentes de software son entidades digitales autónomas que están diseñadas para percibir su entorno, razonar sobre sus objetivos y actuar en consecuencia. A diferencia de los programas de software tradicionales que siguen una lógica fija, los agentes adaptan su comportamiento en función de las entradas contextuales y los marcos de decisión. Esto los hace ideales para entornos dinámicos y distribuidos, como los sistemas nativos de la nube, la robótica, la automatización inteligente y, ahora, la orquestación generativa de la IA.

En esta sección se presentan los componentes básicos de los agentes de software y se explica cómo interactúan estos componentes en las arquitecturas tradicionales en función del modelo de percibir, razonar y actuar. Se analiza cómo la IA generativa, en particular los modelos de lenguaje extensos (LLM), ha transformado la manera en que los agentes de software razonan y planifican. Esto marca un cambio fundamental de los sistemas basados en reglas a la inteligencia aprendida y basada en datos de la IA de los agentes.

En esta sección

Los componentes básicos de los agentes de software

El siguiente diagrama presenta los módulos funcionales clave que se encuentran en la mayoría de los agentes inteligentes. Cada componente contribuye a la capacidad del agente para operar de forma autónoma en entornos complejos.



En el contexto del ciclo de percibir, razonar y actuar, la capacidad de razonamiento de un agente se distribuye entre sus módulos cognitivos y de aprendizaje. Mediante la integración de la memoria y el aprendizaje, el agente desarrolla un razonamiento adaptativo basado en la experiencia pasada. A medida que el agente actúa dentro de su entorno, crea un circuito de retroalimentación emergente: cada acción influye en las percepciones futuras y la experiencia resultante se incorpora a la memoria y a los modelos internos a través del módulo de aprendizaje. Este ciclo continuo de percepción, razonamiento y acción permite al agente mejorar con el tiempo y completa el ciclo completo de percibir, razonar y actuar.

Módulo de percepción

El módulo de percepción permite al agente interactuar con su entorno a través de diversas modalidades de entrada, como texto, audio y sensores. Estas entradas forman los datos sin procesar en los que se basan todos los razonamientos y las acciones. Las entradas de texto pueden incluir indicaciones en lenguaje natural, comandos estructurados o documentos. Las entradas de audio incluyen instrucciones habladas o sonidos ambientales. Las entradas de los sensores incluyen datos físicos, como señales visuales, señales de movimiento o coordenadas GPS. La función principal de la percepción es extraer características y representaciones significativas de estos datos sin procesar. Esto permite al agente construir una comprensión precisa y procesable de su contexto actual. El

proceso puede implicar la extracción de características, el reconocimiento de objetos o eventos y la interpretación semántica, y constituye el primer paso fundamental en el ciclo de percibir, razonar y actuar. La percepción efectiva garantiza que el razonamiento y la toma de decisiones posteriores se basen en un conocimiento situacional relevante y actualizado.

Módulo cognitivo

El módulo cognitivo sirve como núcleo deliberativo del agente de software. Es responsable de interpretar las percepciones, formar la intención y guiar el comportamiento intencional mediante la planificación y la toma de decisiones basadas en objetivos. Este módulo transforma las entradas en procesos de razonamiento estructurados, lo que permite al agente operar de forma intencionada en lugar de reactiva. Estos procesos se gestionan a través de tres submódulos clave: objetivos, planificación y toma de decisiones.

Submódulo de objetivos

El submódulo de objetivos define la intención y la dirección del agente. Los objetivos pueden ser explícitos (por ejemplo, «ir a una ubicación» o «enviar un informe») o implícitos (por ejemplo, «maximizar la participación de los usuarios» o «minimizar la latencia»). Son fundamentales para el ciclo de razonamiento del agente y proporcionan un estado objetivo para su planificación y sus decisiones.

El agente evalúa continuamente el progreso hacia sus objetivos y puede cambiar las prioridades o regenerar los objetivos en función de las nuevas percepciones o el aprendizaje. Este conocimiento de los objetivos permite que el agente se adapte a entornos dinámicos.

Submódulo de planificación

El submódulo de planificación construye estrategias para alcanzar los objetivos actuales del agente. Genera secuencias de acciones, descompone las tareas jerárquicamente y selecciona planes predefinidos o generados dinámicamente.

Para operar con eficacia en entornos no deterministas o cambiantes, la planificación no es estática. Los agentes modernos pueden generar secuencias de pensamiento en cadena, introducir subobjetivos como pasos intermedios y revisar los planes en tiempo real cuando las condiciones cambian.

Este submódulo está estrechamente relacionado con la memoria y el aprendizaje, y permite al agente afinar su planificación a lo largo del tiempo en función de los resultados pasados.

Decision-making submódulo

El submódulo de toma de decisiones evalúa los planes y acciones disponibles para seleccionar el siguiente paso más apropiado. Integra los aportes de la percepción, el plan actual, los objetivos del agente y el contexto ambiental.

Decision-making tiene en cuenta:

- Trade-offs entre objetivos contradictorios
- Umbrales de confianza (por ejemplo, incertidumbre en la percepción)
- Consecuencias de las acciones
- La experiencia aprendida por el agente

Según la arquitectura, los agentes pueden basarse en el razonamiento simbólico, la heurística, el aprendizaje por refuerzo o los modelos de lenguaje (LLM) para tomar decisiones informadas. Este proceso mantiene el comportamiento del agente consciente del contexto, alineado con los objetivos y adaptativo.

Módulo de acción

El módulo de acción es responsable de ejecutar las decisiones seleccionadas por el agente y de interactuar con el mundo externo o los sistemas internos para producir efectos significativos. Representa la fase de acto del ciclo de percepción, razón y acción, en la que la intención se transforma en comportamiento.

Cuando el módulo cognitivo selecciona una acción, el módulo de acción coordina la ejecución a través de submódulos especializados, en los que cada submódulo se alinea con el entorno integrado del agente:

- Actuación física: para los agentes que están integrados en sistemas robóticos o dispositivos de IoT, este submódulo traduce las decisiones en movimientos físicos del mundo real o en instrucciones a nivel de hardware.

  Ejemplos: dirigir un robot, activar una válvula o encender un sensor.
- Interacción integrada: este submódulo gestiona acciones no físicas pero visibles desde el exterior, como la interacción con sistemas de software, plataformas o API.

  Ejemplos: enviar un comando a un servicio en la nube, actualizar una base de datos o enviar un informe mediante una API.

- Invocación de herramientas: los agentes suelen ampliar sus capacidades mediante el uso de herramientas especializadas para realizar subtareas, como las siguientes:
  - Búsqueda: consulta fuentes de conocimiento estructuradas o no estructuradas
  - Resumen: comprimir entradas de texto de gran tamaño para convertirlas en descripciones generales de alto nivel
  - Cálculo: realizar cálculos lógicos, numéricos o simbólicos

  La invocación de herramientas permite componer comportamientos complejos a través de habilidades modulares e invocables.

Módulo de aprendizaje

El módulo de aprendizaje permite a los agentes adaptarse, generalizar y mejorar con el tiempo en función de la experiencia. Apoya el proceso de razonamiento al refinar continuamente los modelos internos, las estrategias y las políticas de decisión del agente mediante el uso de la retroalimentación de la percepción y la acción.

Este módulo funciona en coordinación con la memoria a corto y largo plazo:

- Short-term memoria: almacena el contexto transitorio, como el estado del diálogo, la información de la tarea actual y las observaciones recientes. Ayuda al agente a mantener la continuidad en las interacciones y tareas.
- Long-term memoria: codifica el conocimiento persistente de experiencias pasadas, incluidos los objetivos encontrados anteriormente, los resultados de las acciones y los estados ambientales. Long-term la memoria permite al agente reconocer patrones, reutilizar estrategias y evitar la repetición de errores.

Modos de aprendizaje

El módulo de aprendizaje admite una variedad de paradigmas, como el aprendizaje supervisado, no supervisado y el aprendizaje por refuerzo, que admiten diferentes entornos y funciones de los agentes:

- Aprendizaje supervisado: actualiza los modelos internos basándose en ejemplos etiquetados, a menudo a partir de comentarios humanos o conjuntos de datos de formación.

  Ejemplo: aprender a clasificar la intención de los usuarios en función de conversaciones anteriores.

- Aprendizaje no supervisado: identifica patrones o estructuras ocultos en los datos sin etiquetas explícitas.

  Ejemplo: agrupar las señales ambientales para detectar anomalías.

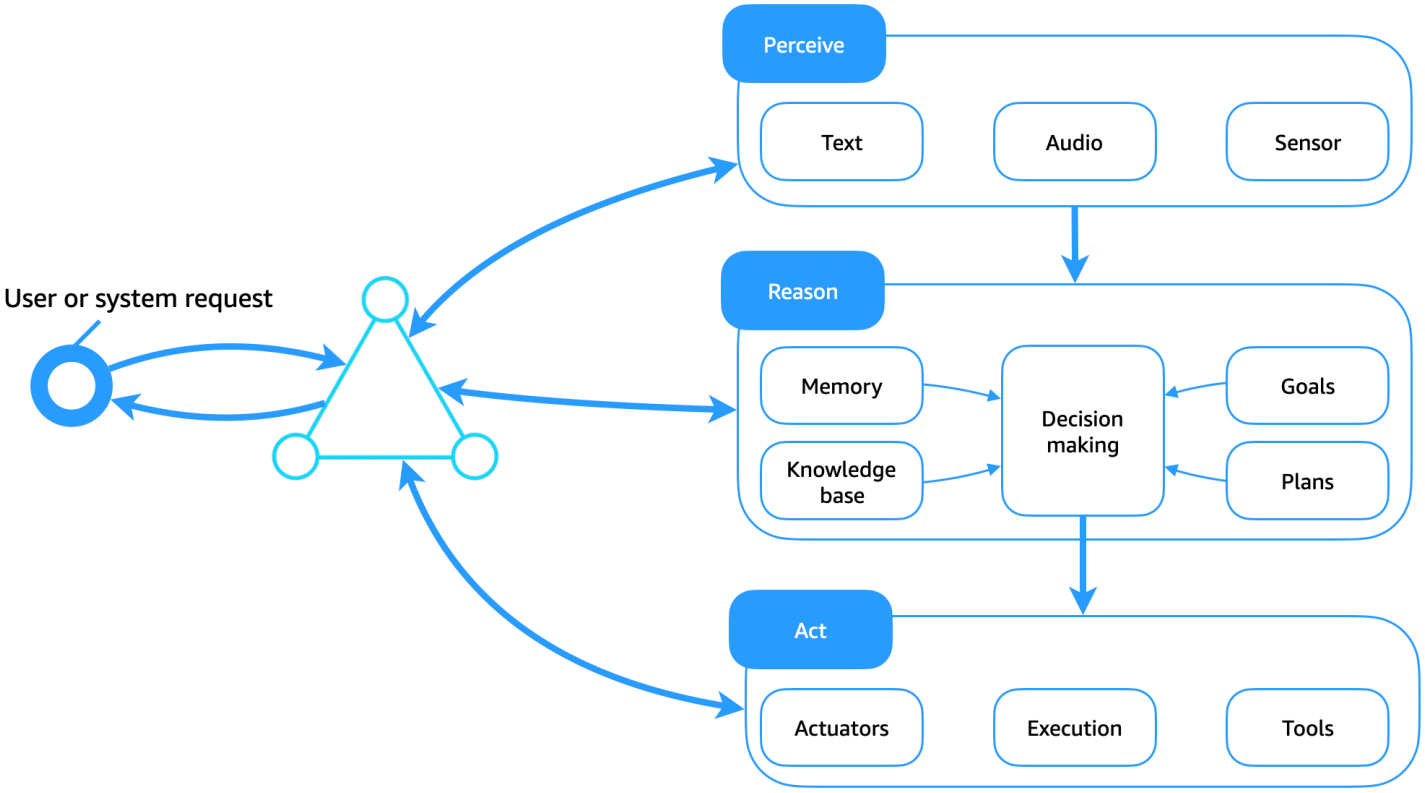
- Aprendizaje reforzado: optimiza el comportamiento mediante prueba y error al maximizar la recompensa acumulada en entornos interactivos.

  Ejemplo: aprender qué estrategia lleva a completar las tareas más rápido.

El aprendizaje se integra estrechamente con el módulo cognitivo del agente. Perfecciona las estrategias de planificación en función de los resultados pasados, mejora la toma de decisiones mediante la evaluación del éxito histórico y mejora continuamente el mapeo entre la percepción y la acción. A través de este circuito cerrado de aprendizaje y retroalimentación, los agentes evolucionan más allá de la ejecución reactiva para convertirse en sistemas que se mejoran a sí mismos y son capaces de adaptarse a nuevos objetivos, condiciones y contextos a lo largo del tiempo.

Arquitectura de agentes tradicional: percibir, razonar, actuar

El siguiente diagrama ilustra cómo funcionan los componentes básicos analizados en la sección anterior en el ciclo de percibir, razonar y actuar.



Módulo Percibe

El módulo de percepción actúa como interfaz sensorial del agente con el mundo externo. Transforma la información ambiental cruda en representaciones estructuradas que informan el razonamiento. Esto incluye el manejo de datos multimodales, como texto, audio o señales de sensores.

- La entrada de texto puede provenir de comandos, documentos o diálogos del usuario.
- La entrada de audio incluye instrucciones habladas o sonidos ambientales.
- La entrada del sensor captura señales del mundo real, como el movimiento, las señales visuales o el GPS.

Una vez ingerida la información sin procesar, el proceso de percepción consiste en extraer las características, seguida del reconocimiento de objetos o eventos y de la interpretación semántica para crear un modelo significativo de la situación actual. Estos resultados proporcionan un contexto estructurado para la toma de decisiones posteriores y anclan el razonamiento del agente en las observaciones del mundo real.

Módulo de motivos

El módulo de la razón es el núcleo cognitivo del agente. Evalúa el contexto, formula la intención y determina las acciones apropiadas. Este módulo organiza el comportamiento impulsado por objetivos utilizando tanto el conocimiento aprendido como el razonamiento.

El módulo Reason consta de submódulos estrechamente integrados:

- Memoria: mantiene el estado del diálogo, el contexto de la tarea y el historial episódico tanto a corto como a largo plazo.
- Base de conocimientos: proporciona acceso a reglas simbólicas, ontologías o modelos aprendidos (como incrustaciones, hechos y políticas).
- Objetivos y planes: define los resultados deseados y elabora estrategias de acción para lograrlos. Los objetivos se pueden actualizar de forma dinámica y los planes se pueden modificar de forma adaptativa en función de los comentarios.
- Decision-making: Actúa como el motor central de arbitraje al sopesar las opciones, evaluar las compensaciones y seleccionar la siguiente acción. Este submódulo tiene en cuenta los umbrales de confianza, la alineación de los objetivos y las restricciones contextuales.

Juntos, estos componentes permiten al agente razonar sobre su entorno, actualizar sus creencias, seleccionar caminos y comportarse de manera coherente y adaptativa. El módulo de la razón cierra la brecha entre la percepción y el comportamiento.

Módulo Act

El módulo act ejecuta la decisión seleccionada por el agente mediante una interfaz con el entorno digital o físico para llevar a cabo las tareas. Aquí es donde la intención se convierte en acción.

Este módulo incluye tres canales funcionales:

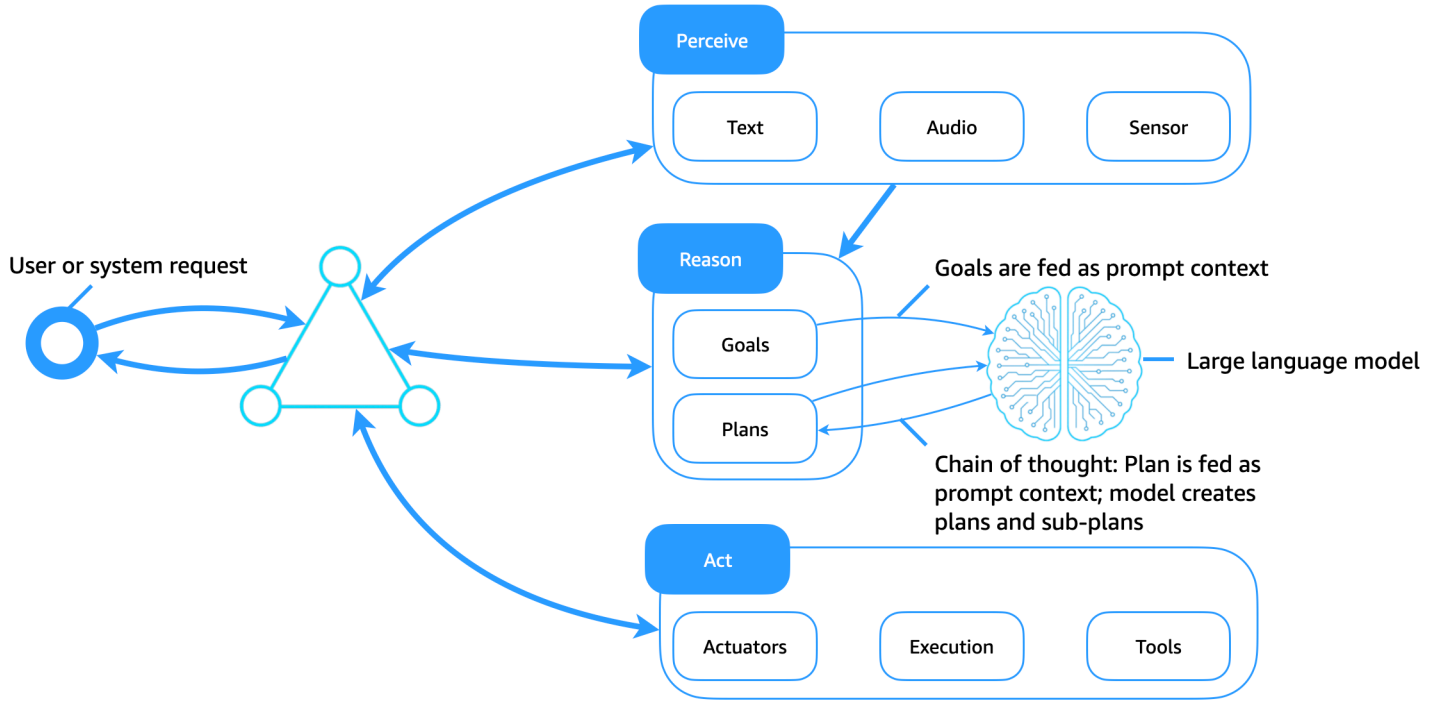
- Actuadores: para los agentes que tienen presencia física (como robots y dispositivos de IoT), controlan las interacciones a nivel de hardware, como el movimiento, la manipulación o la señalización.
- Ejecución: gestiona las acciones basadas en el software, como la invocación de las API, el envío de comandos y la actualización de los sistemas.

- Herramientas: habilita funciones funcionales como la búsqueda, el resumen, la ejecución de código, el cálculo y la gestión de documentos. Estas herramientas suelen ser dinámicas y sensibles al contexto, lo que amplía la utilidad del agente.

Las salidas del módulo act se retroalimentan al entorno y cierran el ciclo. El agente vuelve a percibir estos resultados. Actualizan el estado interno del agente e informan las decisiones futuras, completando así el ciclo de percibir, razonar y actuar.

Agentes de IA generativa: sustituyendo la lógica simbólica por los LLM

El siguiente diagrama ilustra cómo los modelos de lenguaje de gran tamaño (LLM) sirven ahora como un núcleo cognitivo flexible e inteligente para los agentes de software. A diferencia de los sistemas lógicos simbólicos tradicionales, que se basan en bibliotecas de planes estáticos y reglas codificadas a mano, los LLM permiten el razonamiento adaptativo, la planificación contextual y el uso dinámico de herramientas, lo que transforma la forma en que los agentes perciben, razonan y actúan.



Mejoras clave

Esta arquitectura mejora la arquitectura de agentes tradicional de la siguiente manera:

- Los LLM como motores cognitivos: los objetivos, los planes y las consultas se transfieren al modelo como un contexto rápido. El LLM genera vías de razonamiento (como una cadena de pensamiento), descompone las tareas en subobjetivos y decide las próximas acciones.
- Uso de herramientas mediante indicaciones: los LLM se pueden dirigir mediante agentes de uso de herramientas o mediante indicaciones de razonamiento y actuación (ReAct) para que llamen a las API y busquen, consulten, calculen e interpreten los resultados.
- Context-aware planificación: los agentes generan o revisan los planes de forma dinámica en función del objetivo actual del agente, el entorno de entrada y los comentarios, sin necesidad de bibliotecas de planes codificadas.
- El contexto rápido como memoria: en lugar de utilizar bases de conocimiento simbólicas, los agentes codifican la memoria, los planes y los objetivos como símbolos instantáneos que se transmiten al modelo.
- Aprendizaje mediante un aprendizaje breve y contextualizado: los LLM adaptan los comportamientos mediante una ingeniería rápida, lo que reduce la necesidad de readiestramiento explícito o de bibliotecas de planes rígidas.

Lograr una memoria a largo plazo en los agentes LLM-based

A diferencia de los agentes tradicionales, que almacenaban la memoria a largo plazo en bases de conocimiento estructuradas, los agentes de IA generativa deben trabajar dentro de las limitaciones del contexto de las LLM. Para ampliar la memoria y fomentar la inteligencia persistente, los agentes de IA generativa utilizan varias técnicas complementarias: el almacenamiento de agentes, la Retrieval-Augmented generación (RAG), el aprendizaje contextual y el encadenamiento rápido, y la formación previa.

Almacén de agentes: memoria externa a largo plazo

El estado del agente, el historial del usuario, las decisiones y los resultados se almacenan en un almacén de memoria de agente a largo plazo (como una base de datos vectorial, un almacén de objetos o un almacén de documentos). Las memorias relevantes se recuperan a pedido y se insertan en el contexto de los mensajes de LLM durante el tiempo de ejecución. Esto crea un bucle de memoria persistente, en el que el agente conserva la continuidad entre sesiones, tareas o interacciones.

TRAPO

RAG mejora el rendimiento de la LLM al combinar el conocimiento recuperado con las capacidades generativas. Cuando se establece un objetivo o una consulta, el agente busca en un índice de recuperación (por ejemplo, mediante una búsqueda semántica de documentos, conversaciones anteriores o conocimiento estructurado). Los resultados recuperados se adjuntan a la solicitud de LLM, lo que basa la generación en hechos externos o en un contexto personalizado. Este método amplía la memoria efectiva del agente y mejora la confiabilidad y la exactitud de los hechos.

In-context aprendizaje y encadenamiento rápido

Los agentes mantienen la memoria a corto plazo utilizando el contexto simbólico de la sesión y el encadenamiento rápido estructurado. Los elementos contextuales, como el plan actual, los resultados de las acciones anteriores y el estado del agente, se transmiten entre llamadas para guiar el comportamiento.

Capacitación previa y ajustes continuos

En el caso de los agentes de dominios específicos, los LLM pueden seguir formándose previamente sobre recopilaciones personalizadas, como registros, datos empresariales o documentación de productos. Como alternativa, el ajuste fino de la instrucción o el aprendizaje por refuerzo a partir de la retroalimentación humana (RLHF) pueden incorporar un comportamiento similar al de los agentes directamente en el modelo. Esto hace que los patrones de razonamiento pasen de la lógica del tiempo de respuesta a la representación interna del modelo, reduce la longitud de las indicaciones y mejora la eficiencia.

Beneficios combinados de la IA de los agentes

Estas técnicas, cuando se utilizan juntas, permiten a los agentes de IA generativa:

- Mantenga el conocimiento contextual a lo largo del tiempo.
- Adapte el comportamiento en función del historial o las preferencias del usuario.
- Tome decisiones utilizando conocimientos actualizados, fácticos o privados.
- Amplíese a los casos de uso empresarial con comportamientos persistentes, conformes y explicables.

Al aumentar los LLM con memoria externa, niveles de recuperación y formación continua, los agentes pueden lograr un nivel de continuidad cognitiva y un propósito que antes no podían lograr únicamente con sistemas simbólicos.

Comparación de la IA tradicional con los agentes de software y la IA agencial

La siguiente tabla proporciona una comparación detallada de la IA tradicional, los agentes de software y la IA de los agentes.

Característica	IA tradicional	Agentes de software	IA de agencia
Ejemplos	Filtros de spam, clasificadores de imágenes, motores de recomendación	Chatbots, programadores de tareas, agentes de monitorización	Asistentes de IA, agentes desarrolladores autónomos, orquestaciones LLM multiagente
Modelo de ejecución	Batch o sincrónico	Event-driven o programado	Asincrónico, impulsado por eventos y orientado a objetivos
Autonomía	Limitada; a menudo requiere una orquestación humana o externa	Medio; funciona de forma independiente dentro de límites predefinidos	Alto; actúa de forma independiente con estrategias de adaptación
Reactividad	Reactiva a los datos de entrada	Reactivo al entorno y a los eventos	Reactivo y proactivo; anticipa e inicia acciones
Proactividad	Raro	Presente en algunos sistemas	Atributo principal: impulsa el comportamiento dirigido a objetivos
Comunicación	Mínimo; por lo general, independiente o API-bound	Inter-agent o mensajería entre un agente y un humano	Interacción rica entre múltiples agentes y entre personas informadas

Característica	IA tradicional	Agentes de software	IA de agencia
Decision-making	Modele únicamente la inferencia (clasificación, predicción, etc.)	Razonamiento simbólico o decisiones basadas en reglas o guionadas	Razonamiento contextual, dinámico y basado en objetivos (a menudo) LLM-enhanced
Intención delegada	No; realiza tareas definidas directamente por el usuario	Parcial; actúa en nombre de usuarios o sistemas con un alcance limitado	Sí; actúa con objetivos delegados, a menudo en todos los servicios, usuarios o sistemas
Aprendizaje y adaptación	A menudo se centra en el modelo (por ejemplo, la formación en aprendizaje automático)	A veces adaptativo	Aprendizaje, memoria o razonamiento integrados (por ejemplo, retroalimentación, autocorrección)
Agencia	Ninguna; herramientas para humanos	¿Implícito o básico	Explícito; opera con un propósito, metas y autodirección
Conciencia del contexto	Bajo; apátrida o basado en instantáneas	Moderado; algún seguimiento estatal	Alto; utiliza modelos de memoria, contexto situacional y entorno
Rol de infraestructura	Integrado en aplicaciones o canales de análisis	Componente de middleware o capa de servicio	Malla de agentes componible integrada con sistemas en la nube, sin servidor o periféricos

En resumen:

- La IA tradicional se centra en las herramientas y es limitada desde el punto de vista funcional. Se centra en la predicción o la clasificación.
- Los agentes de software tradicionales introducen la autonomía y la comunicación básica, pero a menudo están sujetos a reglas o son estáticos.
- La IA de Agentic aúna autonomía, asincronía y agencia. Permite que las entidades inteligentes y orientadas a objetivos puedan razonar, actuar y adaptarse dentro de sistemas complejos. Esto hace que la IA agentic sea ideal para el futuro nativo de la nube. AI-driven

Pasos siguientes

Esta guía analiza la historia y los fundamentos de la IA de los agentes, que representa la evolución de los agentes de software tradicionales hacia sistemas autónomos e inteligentes que funcionan con la IA generativa. Describió cómo los primeros agentes de software seguían reglas y lógicas predefinidas para automatizar las tareas dentro de límites fijos, y explicó cómo la IA de los agentes se basa en esta base al incorporar modelos de lenguaje de gran tamaño, que permiten a los agentes razonar, aprender y adaptarse de forma dinámica en entornos abiertos.

Puede explorar la IA de los agentes en profundidad consultando las siguientes publicaciones de esta serie:

- La puesta en marcha de la IA de los agentes AWS proporciona una estrategia organizativa para transformar la IA de los agentes, pasando de ser experimentos aislados a convertirse en una infraestructura a escala empresarial que genera valor.
- Los patrones y flujos de trabajo de la IA de Agentic AWS analizan los planos fundamentales y las estructuras modulares que se utilizan para diseñar, componer y organizar agentes de IA orientados a objetivos.
- Los marcos, protocolos y herramientas de inteligencia artificial de las agencias abordan los fundamentos, los conjuntos de herramientas y los protocolos del software que AWS hay que tener en cuenta a la hora de crear las soluciones de IA de las agencias.
- La creación de arquitecturas sin servidor para la IA de los agentes AWS analiza las arquitecturas sin servidor como base natural de las cargas de trabajo de IA modernas y describe cómo se pueden crear arquitecturas sin servidor nativas de la IA en el. Nube de AWS
- La creación de arquitecturas multiusuario para la IA de los agentes AWS describe el uso de agentes de IA en entornos con varios usuarios, incluidas las consideraciones de alojamiento, los modelos de despliegue y los planos de control.

Recursos

Para obtener más información sobre los conceptos analizados en esta guía, consulte las siguientes guías y artículos.

AWS referencias

- Agentes de Amazon Bedrock
- Amazon Q Developer
- SDK de Strands Agents

Otras referencias

- Hewitt, Carl, Peter Bishop y Richard Steiger. «Un formalismo ACTOR modular universal para la inteligencia artificial». Actas de la 3ª Conferencia Internacional Conjunta sobre Inteligencia Artificial (1973): 235-245. https://www.ijcai. org/Proceedings/73/Papers/027B.pdf
- Lesser, Victor R., publicaciones relevantes (ver lista completa):
  - Lesser, Victor R. y Daniel D. Corkill. «Sistemas distribuidos cooperativos y funcionalmente precisos». Transacciones del IEEE sobre sistemas, hombre y cibernética 11, núm. 1 (1981): 81-96. https://ieeexplore.ieee. org/abstract/document/4308581
  - Decker, Keith S. y Victor R. Lesser. «La comunicación al servicio de la coordinación». Taller de la AAAI sobre planificación de la comunicación interagente (1994). https://www.researchgate. net/profile/Victor-Lesser/publication/2768884_Communication_in_the_Service_of_Coordination/links/00b7d51cc2a0750cb4000000/Communication-in-the-Service-of-Coordination.pdf
  - Durfee, Edmund H., Victor R. Lesser y Daniel D. Corkill. «Tendencias en la resolución cooperativa de problemas distribuida». Transacciones del IEEE sobre ingeniería del conocimiento y los datos (1989). http://mas.cs.umass. edu/Documents/ieee-tkde89.pdf
  - Durfee, Edmund H., V.R. Lesser y D.D. Corkill, «Inteligencia artificial distribuida». La cooperación a través de la comunicación en una red distribuida de resolución de problemas (1987): 29-58. https://www.academia. edu/download/79885643/durf94_1.pdf
  - Lâasri, Brigitte, Hassan Lâasri, Susan Lander y Victor Lesser. «Un modelo genérico para agentes negociadores inteligentes». Revista Internacional de Sistemas de Información Cooperativa 01, núm. 02 (1992): 291-317. https://doi.org/10.1142/S0218215792000210

  - Lander, Susan E. y Victor R. Lesser. «Comprender el papel de la negociación en la búsqueda distribuida entre agentes heterogéneos». IJCAI'93: Actas de la 13ª conferencia internacional conjunta sobre inteligencia artificial (1993): 438-444. https://www.ijcai. org/Proceedings/93-1/Papers/062.pdf
  - Lander, Susan, Victor R. Lesser y Margaret E. Connell. «Estrategias de resolución de conflictos para agentes expertos que cooperan» CKBS'90: Actas de la conferencia internacional de trabajo sobre sistemas cooperantes basados en el conocimiento (octubre de 1990): 183 a 200. https://doi.org/10.1007/978-1-4471-1831-2_10
  - Prasad, M. V. Nagendra, Victor Lesser y Susan E. Lander. «Experimentos de aprendizaje en un sistema heterogéneo de múltiples agentes». Taller IJCAI-95 sobre adaptación y aprendizaje en sistemas multiagentes (1995): 59-64. https://www.researchgate.net/publication/2784280_Learning_Experiments_in_a_Heterogeneous_Multi-agent_System
- Nwana, Hyacinth S. «Agentes de software: una visión general». Knowledge Engineering Review 11, núm. 3 (octubre/noviembre de 1996): 205-244. https://teaching.shu.ac. uk/aces/rh1/elearning/multiagents/introduction/nwana.pdf
- Selfridge, Oliver G. «El pandemonio: un paradigma para el aprendizaje». Mecanización de los procesos de pensamiento: actas de un simposio celebrado en el Laboratorio Nacional de Física 1 (1959): 511—529. https://aitopics. org/download/classics. ----SEP----:504E1BAC
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser e Illia Polosukhin. «Todo lo que necesitas es atención». Actas de la 31ª Conferencia sobre Sistemas de Procesamiento de Información Neural (NIPS). Avances en los sistemas de procesamiento de información neuronal 30 (2017): 5998-6008. https://proceedings.neurips. cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- Wooldridge, Michael y Nicholas R. Jennings. «Agentes inteligentes: teoría y práctica». Revista de ingeniería del conocimiento 10, núm. 2 (enero de 1995): 115-152. https://www.cs.cmu.edu/ ~ _intelligent_agents.pdf motionplanning/papers/sbp_papers/integrated1/woodridge

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.