Fundamentos da IA agêntica em AWS

AWS Orientação prescritiva

AWS Orientação prescritiva: Fundamentos da IA agêntica em AWS

Table of Contents

Fundamentos da IA agente em AWS

Aaron Sempf, Amazon Web Services

Julho de 2025 (histórico do documento)

Em um mundo de sistemas cada vez mais inteligentes, distribuídos e autônomos, o conceito de agente — uma entidade que pode perceber seu ambiente, raciocinar sobre seu estado e agir com intenção — tornou-se fundamental. Os agentes não são meramente programas que executam instruções; eles são entidades orientadas a objetivos e conscientes do contexto que tomam decisões em nome de usuários, sistemas ou organizações. Seu surgimento reflete uma mudança na forma como você cria e pensa sobre software: uma mudança da lógica processual e da automação reativa para sistemas que operam com autonomia e propósito.

Na interseção entre IA, sistemas distribuídos e engenharia de software está um paradigma poderoso conhecido como IA agêntica. Essa nova geração de sistemas inteligentes consiste em agentes de software capazes de comportamento adaptativo, coordenação complexa e tomada de decisão delegada.

Este guia apresenta os princípios que definem os agentes de software modernos e descreve sua evolução em direção à IA agente. Para explicar essa mudança, o guia fornece a base conceitual e, em seguida, traça a evolução dos agentes de software até a IA agente:

- A introdução aos agentes de software define os agentes de software, os compara aos componentes de software tradicionais e apresenta as características essenciais que diferenciam o comportamento do agente da automação tradicional, baseando-se em estruturas estabelecidas.
- O objetivo dos agentes de software examina por que os agentes de software existem, quais funções eles desempenham, quais problemas eles resolvem e como eles permitem a delegação inteligente, reduzem a carga cognitiva e apoiam o comportamento adaptativo em ambientes dinâmicos.
- A evolução dos agentes de software traça os marcos intelectuais e tecnológicos que moldaram os agentes de software, desde os primeiros conceitos de autonomia e concorrência até o surgimento de sistemas multiagentes e arquiteturas de agentes formais, resultando na convergência com a IA generativa.
- Os agentes de software da IA agente apresentam a IA agêntica como o ponto culminante de décadas de progresso que combina modelos de agentes distribuídos com modelos básicos, computação sem servidor e protocolos de orquestração. Esta seção descreve como essa

convergência possibilita uma nova geração de agentes inteligentes que usam ferramentas e operam com autonomia, assincronicidade e verdadeira agência em grande escala.

Público-alvo

Este guia foi desenvolvido para arquitetos, desenvolvedores e líderes de tecnologia que desejam entender a história, os principais conceitos e a evolução dos agentes de software para a IA agente antes de adotarem essa tecnologia para soluções de nuvem modernas em AWS.

Objetivos

A adoção de arquiteturas agênticas ajuda as organizações a:

- Acelere a geração de valor: automatize e escale o trabalho com conhecimento e reduza o esforço manual e a latência.
- Melhore o engajamento do cliente: forneça assistentes inteligentes em todos os domínios.
- Reduza os custos operacionais: automatize os fluxos de decisão que antes exigiam intervenção ou supervisão humana.
- Promova a inovação e a diferenciação: crie produtos inteligentes que se adaptam, aprendem e competem em tempo real.
- Modernize fluxos de trabalho antigos: reestruture scripts e monólitos em agentes de raciocínio modulares.

Sobre esta série de conteúdo

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

Introdução aos agentes de software

O conceito de agentes de software evoluiu significativamente desde sua fundação em entidades autônomas na década de 1960 até sua exploração formal no início da década de 1990. À medida que os sistemas digitais se tornam cada vez mais complexos — de scripts determinísticos a aplicativos adaptáveis e inteligentes — os agentes de software se tornam elementos essenciais para permitir um comportamento autônomo, sensível ao contexto e orientado por metas em sistemas de computação. No contexto de arquiteturas nativas da nuvem e aprimoradas por IA, particularmente com o advento da IA generativa, grandes modelos de linguagem () e plataformas como o Amazon BedrockLLMs, os agentes de software estão sendo redefinidos por meio de novas lentes de capacidade e escala.

Esta introdução se baseia no trabalho seminal Software Agents: An Overview, de Hyacinth S. Nwana (Nwana 1996). Ele define agentes de software, discute suas raízes conceituais e estende a discussão em uma estrutura contemporânea para definir três princípios abrangentes dos agentes de software modernos: autonomia, assincronicidade e agência. Esses princípios distinguem os agentes de software de outros tipos de serviços ou aplicativos e permitem que esses agentes operem com propósito, resiliência e inteligência em ambientes distribuídos em tempo real.

Nesta seção

Da autonomia à inteligência distribuída

Antes de o termo agente de software entrar no mainstream, as primeiras pesquisas em computação exploraram a ideia de entidades digitais autônomas, que são sistemas capazes de agir de forma independente, reagir às entradas e tomar decisões com base em regras ou objetivos internos. Essas ideias iniciais estabeleceram a base conceitual para o que se tornaria o paradigma do agente. (Para obter um cronograma histórico, consulte a seção A evolução dos agentes de software mais adiante neste guia.)

Conceitos iniciais de autonomia

A noção de máquinas ou programas que agem independentemente de operadores humanos intriga os projetistas de sistemas há décadas. Os primeiros trabalhos em cibernética, inteligência artificial e sistemas de controle examinaram como o software poderia exibir um comportamento autorregulado, responder dinamicamente às mudanças e operar sem supervisão humana contínua.

Essas ideias introduziram a autonomia como um atributo central dos sistemas inteligentes e prepararam o terreno para o surgimento de softwares capazes de decidir e agir, em vez de apenas reagir ou executar.

O modelo do ator e a execução assíncrona

Na década de 1970, o modelo de ator, introduzido no paper A Universal Modular ACTOR Formalism for Artificial Intelligence (Hewitt et al. 1973), forneceu uma estrutura formal para pensar em computação descentralizada e baseada em mensagens. Nesse modelo, os atores são entidades independentes que se comunicam exclusivamente por meio da transmissão de mensagens assíncronas e permitem sistemas escaláveis, simultâneos e tolerantes a falhas.

O modelo do ator enfatizou três atributos principais que continuam a influenciar o design moderno dos agentes:

- Isolamento de estado e comportamento
- Interação assíncrona entre entidades
- Criação dinâmica e delegação de tarefas

Esses atributos se alinharam às necessidades dos sistemas distribuídos e prefiguraram as características operacionais dos agentes de software em ambientes nativos da nuvem.

Inteligência distribuída e sistemas multiagentes

À medida que os sistemas de computação se tornaram mais interconectados após a década de 1960, os pesquisadores exploraram a inteligência artificial distribuída (DAI). Esse campo se concentrou em como várias entidades autônomas poderiam trabalhar de forma colaborativa ou competitiva em um sistema. A DAI levou ao desenvolvimento de sistemas multiagentes, em que cada agente tem metas, percepções e raciocínios locais, mas também opera em um ambiente mais amplo e interconectado.

Essa visão de inteligência distribuída, em que a tomada de decisões é descentralizada e o comportamento emergente surge da interação do agente, permanece fundamental para a forma como os sistemas modernos baseados em agentes são concebidos e construídos.

A tipologia de Nwana e a ascensão dos agentes de software

A formalização do conceito de agente de software em meados da década de 1990 marcou uma virada na evolução dos sistemas inteligentes. Entre as contribuições mais influentes para essa formalização está o artigo seminal de Hyacinth S. Nwana, Software Agents: An Overview (Nwana 1996), que forneceu uma das primeiras estruturas abrangentes para categorizar e compreender agentes de software em várias dimensões.

Neste paper, Nwana examina o estado da pesquisa de agentes de software e identifica uma divergência crescente na forma como os agentes estavam sendo definidos e implementados. O paper destaca a necessidade de uma estrutura conceitual comum e propõe uma tipologia que classifique os agentes de acordo com suas principais capacidades. Ele analisa sistemas de agentes representativos da academia e da indústria, distingue agentes de programas e objetos tradicionais e descreve os desafios e oportunidades na computação baseada em agentes.

Nwana enfatiza que os agentes de software não são um conceito monolítico, mas existem em um espectro de sofisticação e capacidade. A tipologia serve para esclarecer essa paisagem e orientar futuros projetos e pesquisas.

Nwana define um agente de software como uma entidade de software que funciona de forma contínua e autônoma em um ambiente específico, que geralmente é habitado por outros agentes e processos. Essa definição enfatiza duas características principais:

- Continuidade: O agente opera de forma persistente ao longo do tempo, sem exigir intervenção humana constante.
- Autonomia: O agente tem a capacidade de tomar decisões e agir de forma independente, com base em sua percepção do ambiente.

Essa definição, combinada com a tipologia de agentes de Nwana, enfatiza a autoridade delegada (por meio da autonomia) e a proatividade como características fundamentais dos agentes. Ele diferencia entre agentes e sub-rotinas ou serviços ao destacar a capacidade do agente de agir de forma independente em nome de outra entidade e de iniciar um comportamento em busca de metas, em vez de apenas responder a comandos diretos.

Tipologia do agente de Nwana

Para diferenciar ainda mais os vários tipos de agentes, a Nwana apresenta um sistema de classificação baseado em seis atributos principais:

- Autonomia: O agente opera sem intervenção direta de humanos ou outros.
- Habilidade social: o agente interage com outros agentes ou humanos usando mecanismos de comunicação.
- Reatividade: o agente percebe seu ambiente e responde em tempo hábil.
- Proatividade: o agente exibe um comportamento direcionado a um objetivo ao tomar a iniciativa.
- Adaptabilidade e aprendizado: o agente melhora seu desempenho ao longo do tempo por meio da experiência.
- Mobilidade: o agente pode se mover entre diferentes ambientes de sistema ou redes.

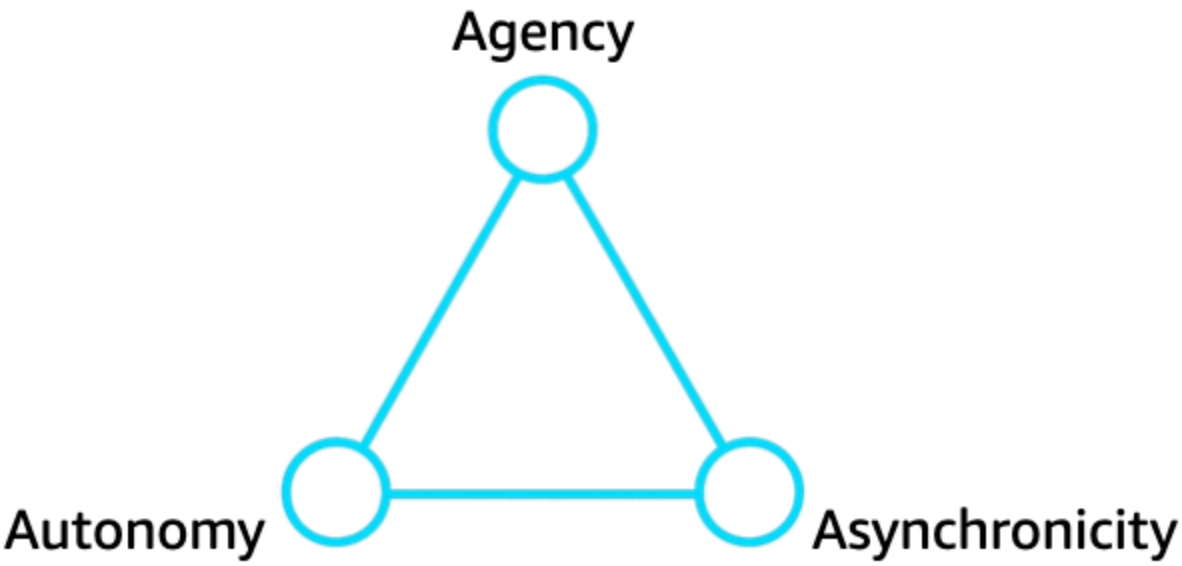
Da tipologia aos princípios agênticos modernos

O trabalho de Nwana serviu tanto como uma taxonomia quanto como uma lente fundamental por meio da qual a comunidade de computação poderia avaliar a evolução das formas de agência em software. Sua ênfase na autonomia, proatividade e no conceito de agir em nome de um usuário ou sistema estabeleceu as bases para o que agora consideramos comportamento agente.

Embora as tecnologias e os ambientes tenham mudado, especialmente com o surgimento da IA generativa, da infraestrutura sem servidor e das estruturas de orquestração multiagente, os insights fundamentais do trabalho de Nwana permanecem relevantes. Eles fornecem uma ponte crítica entre a teoria inicial dos agentes e os três pilares modernos dos agentes de software.

Os três pilares dos agentes de software modernos

No contexto das atuais plataformas baseadas em IA, arquiteturas de microsserviços e sistemas orientados a eventos, os agentes de software podem ser definidos por três princípios interdependentes que os distinguem dos serviços padrão ou dos scripts de automação: autonomia, assincronicidade e agência. Na ilustração a seguir e nos diagramas subsequentes, o triângulo representa esses três pilares dos agentes de software modernos.



Autonomia

Agentes modernos operam de forma independente. Eles tomam decisões com base no estado interno e no contexto ambiental sem exigir instruções humanas. Isso permite que eles reajam aos dados em tempo real, gerenciem seu próprio ciclo de vida e ajustem seu comportamento com base em metas e informações situacionais.

A autonomia é a base do comportamento do agente. Ele permite que os agentes funcionem sem supervisão contínua ou fluxos de controle codificados.

Assincronicidade

Os agentes são fundamentalmente assíncronos. Isso significa que eles respondem a eventos, sinais e estímulos à medida que ocorrem, sem depender de chamadas bloqueadas ou fluxos de trabalho lineares. Essa característica permite comunicação escalável e sem bloqueio, capacidade de resposta em ambientes distribuídos e acoplamento frouxo entre componentes.

Por meio da assincronicidade, os agentes podem participar de sistemas em tempo real e se coordenar com outros serviços ou agentes de forma fluida e eficiente.

Agência como princípio definidor

Autonomia e assincronicidade são necessárias, mas esses recursos por si só não são suficientes para tornar um sistema um verdadeiro agente de software. O diferencial crítico é a agência, que introduz:

- Comportamento direcionado a metas: os agentes buscam objetivos e avaliam o progresso em relação a eles.
- Tomada de decisão: os agentes avaliam as opções e escolhem ações com base em regras, modelos ou políticas aprendidas.
- Intenção delegada: os agentes agem em nome de uma pessoa, sistema ou organização e têm um senso de propósito incorporado.
- Raciocínio contextual: os agentes incorporam memória ou modelos de seu ambiente para orientar o comportamento de forma inteligente.

Um sistema autônomo e assíncrono ainda pode ser um serviço reativo. O que o torna um agente de software é sua capacidade de agir com intenção e propósito, de ser agente.

Agência com propósito

Os princípios de autonomia, assincronicidade e agência permitem que os sistemas operem de forma inteligente, adaptativa e independente em ambientes distribuídos. Esses princípios estão enraizados em décadas de evolução conceitual e arquitetônica e agora sustentam muitos dos sistemas de IA mais avançados que estão sendo construídos atualmente.

Nessa nova era de IA generativa, orquestração orientada a objetivos e colaboração multiagente, é essencial entender o que torna um agente de software verdadeiramente agente. Reconhecer a agência como a característica definidora nos ajuda a ir além da automação e entrar no reino da inteligência autônoma com propósito.

O propósito dos agentes de software

À medida que os sistemas modernos se tornam cada vez mais complexos, distribuídos e inteligentes, o papel dos agentes de software ganhou destaque em domínios que variam de operações autônomas a tecnologias de assistência ao usuário. Mas qual é o propósito subjacente dos agentes de software? Por que projetamos sistemas que vão além de scripts, serviços ou modelos estáticos e, em vez disso, delegamos tarefas a entidades capazes de perceber, raciocinar e agir?

Esta seção explora o propósito fundamental dos agentes de software: permitir a delegação inteligente de tarefas em ambientes dinâmicos, com foco na autonomia, adaptabilidade e ação proposital. Ele apresenta a base conceitual dos agentes de software, traça sua estrutura cognitiva e descreve os problemas do mundo real que eles estão exclusivamente equipados para resolver.

Nesta seção

Do modelo do ator à cognição do agente

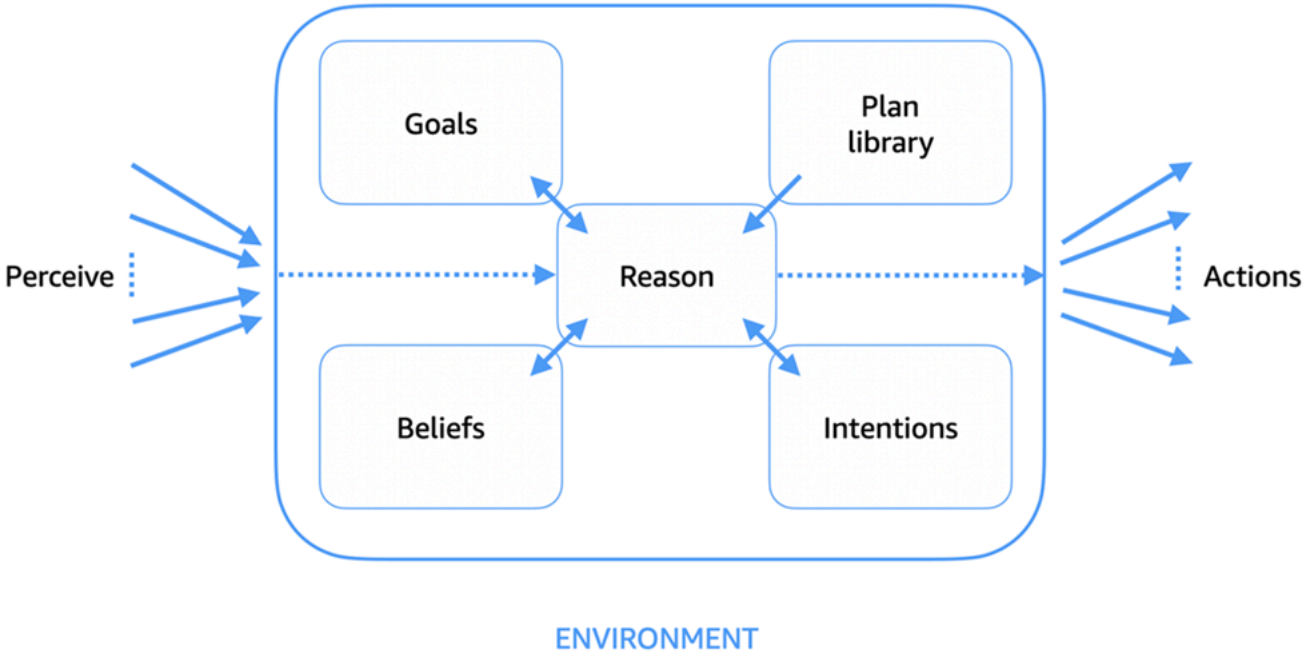
O propósito e a estrutura dos agentes de software são baseados em ideias que surgiram dos primeiros modelos de computação, particularmente o modelo de ator que foi introduzido por Carl Hewitt na década de 1970 (Hewitt et al. 1973).

O modelo de ator trata a computação como uma coleção de entidades independentes que executam simultaneamente, chamadas de atores. Cada ator encapsula seu próprio estado, interage exclusivamente por meio da transmissão assíncrona de mensagens e pode criar novos atores e delegar tarefas.

Esse modelo forneceu a base conceitual para raciocínio, reatividade e isolamento descentralizados — todos os quais sustentam a arquitetura comportamental dos agentes de software modernos.

A função do agente: perceber, raciocinar, agir

No centro de cada agente de software está um ciclo cognitivo que geralmente é descrito como o ciclo de perceber, raciocinar e agir. Esse processo é ilustrado no diagrama a seguir. Ele define como os agentes operam de forma autônoma em ambientes dinâmicos.



- Perceba: os agentes coletam informações (por exemplo, eventos, entradas de sensores ou sinais de API) do ambiente e atualizam seu estado interno ou suas crenças.
- Motivo: os agentes analisam crenças, metas e conhecimento contextual atuais usando uma biblioteca de planos ou sistema lógico. Esse processo pode envolver priorização de metas, resolução de conflitos ou seleção de intenções.
- Agir: os agentes selecionam e executam ações que os aproximam de atingir suas metas delegadas.

Essa arquitetura suporta a capacidade dos agentes de funcionarem além da programação rígida e permite um comportamento flexível, sensível ao contexto e direcionado a metas. Ele forma a estrutura mental que orienta os propósitos mais amplos dos agentes de software.

Colaboração autônoma e intencionalidade

O objetivo dos agentes de software é trazer autonomia, consciência contextual e delegação inteligente à computação moderna. Como os agentes são construídos com base nos princípios do modelo do ator e incorporados no ciclo de perceber, raciocinar e agir, eles possibilitam sistemas que não são apenas reativos, mas proativos e propositais.

Os agentes capacitam o software a decidir, adaptar e agir em ambientes complexos. Eles representam usuários, interpretam metas e implementam tarefas na velocidade da máquina. À

medida que avançamos na era da IA agente, os agentes de software estão se tornando a interface operacional entre a intenção humana e a ação digital inteligente.

Delegando intenção

Diferentemente dos componentes de software tradicionais, os agentes de software existem para agir em nome de outra coisa: um usuário, outro sistema ou um serviço de nível superior. Eles têm intenção delegada, o que significa que eles:

- Opere de forma independente após o início.
- Faça escolhas que estejam alinhadas com as metas do delegador.
- Navegue pelas incertezas e compensações na execução.

Os agentes preenchem a lacuna entre as instruções e os resultados, o que permite que os usuários expressem a intenção em um nível mais alto de abstração, em vez de exigir instruções explícitas.

Operando em ambientes dinâmicos e imprevisíveis

Os agentes de software são projetados para ambientes em que as condições mudam constantemente, os dados chegam em tempo real e o controle e o contexto são distribuídos.

Ao contrário dos programas estáticos que exigem entradas exatas ou execução síncrona, os agentes se adaptam ao ambiente e respondem dinamicamente. Esse é um recurso vital em infraestrutura nativa em nuvem, computação de ponta, redes de Internet das Coisas (IoT) e sistemas de tomada de decisão em tempo real.

Reduzindo a carga cognitiva humana

Um dos principais objetivos dos agentes de software é reduzir a carga cognitiva e operacional dos humanos. Os agentes podem:

- Monitore continuamente sistemas e fluxos de trabalho.
- Detecte e responda a condições predefinidas ou emergentes.
- Automatize decisões repetitivas e de alto volume.
- Reaja às mudanças ambientais com latência mínima.

Quando a tomada de decisão muda dos usuários para os agentes, os sistemas se tornam mais responsivos, resilientes e centrados no ser humano, e podem se adaptar em tempo real a novas

informações ou interrupções. Isso permite uma resposta mais rápida, bem como uma maior continuidade operacional em ambientes de alta complexidade ou de alta escala. O resultado é uma mudança no foco humano, da tomada de decisão em nível micro para a supervisão estratégica e a solução criativa de problemas.

Habilitando a inteligência distribuída

A capacidade dos agentes de software de operar individual ou coletivamente permite o design de sistemas multiagentes (MAS) que se coordenam entre ambientes ou organizações. Esses sistemas podem distribuir tarefas de forma inteligente e negociar, cooperar ou competir em direção a metas compostas.

Por exemplo, em um sistema global de cadeia de suprimentos, agentes individuais gerenciam fábricas, remessas, armazéns e entregas de última milha. Cada agente opera com autonomia local: os agentes da fábrica otimizam a produção com base nas restrições de recursos, os agentes do armazém ajustam os fluxos de estoque em tempo real e os agentes de entrega redirecionam as remessas com base no tráfego e na disponibilidade do cliente.

Esses agentes se comunicam e coordenam dinamicamente e se adaptam a interrupções, como atrasos em portos ou falhas em caminhões, sem controle centralizado. A inteligência geral do sistema surge dessas interações e permite uma logística resiliente e otimizada que está além das capacidades de um único componente.

Nesse modelo, os agentes atuam como nós em uma estrutura de inteligência mais ampla. Eles formam sistemas emergentes que são capazes de resolver problemas que nenhum componente sozinho poderia resolver.

Agindo com propósito, não apenas reação

A automação por si só é insuficiente em sistemas complexos. O propósito definidor de um agente de software é agir com propósito e avaliar metas, pesar o contexto e fazer escolhas informadas. Isso significa que os agentes de software buscam metas em vez de apenas responder aos gatilhos. Eles podem revisar crenças e intenções com base na experiência ou no feedback. Nesse contexto, as crenças se referem à representação interna do agente sobre o ambiente (por exemplo, “o pacote X está no armazém A”), com base em suas percepções (entrada e sensores). As intenções se referem aos planos que o agente escolhe para atingir uma meta (por exemplo, “usar a rota de entrega B e notificar o destinatário”). Os agentes também podem escalar, adiar ou adaptar as ações conforme necessário.

Essa intencionalidade é o que torna os agentes de software não apenas executores reativos, mas colaboradores autônomos em sistemas inteligentes.

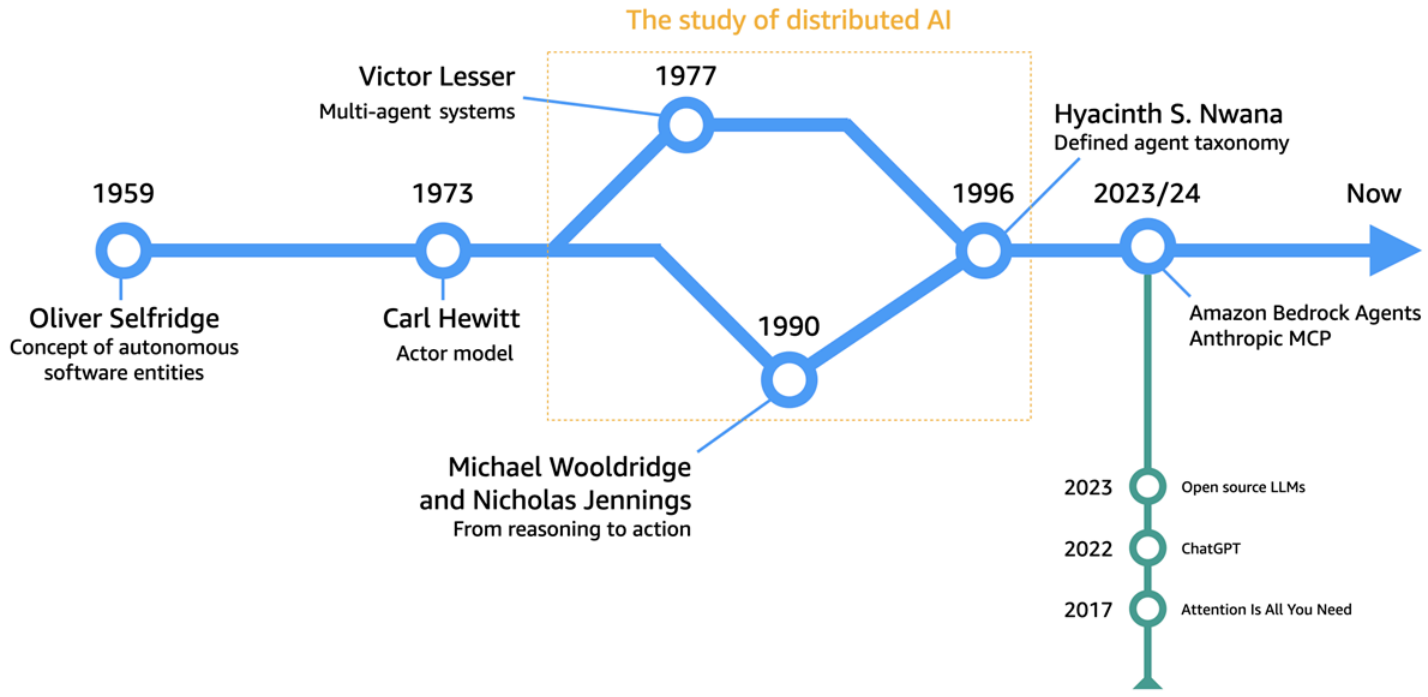
A evolução dos agentes de software

A jornada de sistemas automatizados simples para agentes de software inteligentes, autônomos e direcionados a objetivos reflete décadas de evolução em ciência da computação, inteligência artificial e sistemas distribuídos.

Essa evolução foi seguida pelo surgimento do aprendizado de máquina, que mudou o paradigma das regras artesanais para o reconhecimento estatístico de padrões. Esses sistemas poderiam aprender com os dados e possibilitar avanços na percepção, classificação e tomada de decisão.

Modelos de linguagem grandes (LLMs) representam uma convergência de escala, arquitetura e aprendizado não supervisionado. LLMs pode raciocinar, gerar e adaptar tarefas com pouco ou nenhum treinamento específico para cada tarefa. Ao combinar LLMs uma infraestrutura nativa em nuvem escalável e arquiteturas combináveis, agora estamos alcançando a visão completa da IA agêntica: agentes de software inteligentes que podem operar com autonomia, consciência do contexto e adaptabilidade em escala corporativa.

Esta seção explora a história dos agentes de software, da teoria fundamental à prática moderna, conforme ilustrado no diagrama a seguir. Ele destaca a convergência da inteligência artificial distribuída (DAI) e da IA generativa baseada em transformadores e identifica os principais marcos que moldaram o surgimento da IA agente.



Nesta seção

Fundamentos dos agentes de software

1959 — Oliver Selfridge: o nascimento da autonomia em software

As raízes dos agentes de software remontam a Oliver Selfridge, que introduziu o conceito de entidades autônomas de software (demônios) — programas capazes de perceber seu ambiente e agir de forma independente (Selfridge 1959). Seus primeiros trabalhos em percepção e aprendizado de máquinas estabeleceram as bases filosóficas para futuras noções de agentes como sistemas independentes e inteligentes.

1973 — Carl Hewitt: o ator modelo

Um avanço fundamental veio com o modelo de ator de Carl Hewitt (Hewitt et al. 1973), que é um modelo computacional formal que descreve os agentes como entidades independentes e simultâneas. Nesse modelo, os agentes podem encapsular seu próprio estado e comportamento, comunicar-se usando a passagem assíncrona de mensagens e criar dinamicamente outros atores e delegar tarefas a eles.

O modelo do ator forneceu a base teórica e o paradigma arquitetônico para sistemas distribuídos baseados em agentes. Esse modelo prefigurou implementações modernas de concorrência, como a linguagem de programação Erlang e a estrutura Akka.

Amadurecendo o campo: do raciocínio à ação

1977 — Victor Lesser: sistemas multiagentes

No final da década de 1970, surgiu a inteligência artificial distribuída (DAI). Foi promovido por Victor Lesser, que é amplamente reconhecido por ser pioneiro em sistemas multiagentes (MAS). Seu trabalho se concentrou em como entidades independentes de software poderiam cooperar, coordenar e negociar (consulte a seção Recursos). Esse desenvolvimento resultou em sistemas

capazes de resolver problemas complexos coletivamente — um salto essencial na criação de inteligência distribuída.

Década de 1990 — Michael Wooldridge e Nicholas Jennings: o espectro de agentes

Na década de 1990, o campo da inteligência distribuída amadureceu com contribuições de pesquisadores como Michael Wooldridge e Nicholas Jennings. Esses estudiosos categorizaram os agentes ao longo de um espectro, do reativo ao deliberativo, de sistemas não cognitivos a agentes de raciocínio orientados por objetivos (Wooldridge e Jennings 1995). Seu trabalho enfatizou que os agentes não eram mais ideias abstratas, mas estavam sendo aplicados em uma ampla variedade de domínios práticos, da robótica ao software corporativo.

Esses pesquisadores também introduziram uma mudança de foco: do raciocínio centralizado para a ação distribuída. Os agentes não eram mais apenas pensadores — eram agentes que operavam em ambientes em tempo real com autonomia e propósito.

1996 — Hyacinth S. Nwana: formalizando o conceito de agente

Em 1996, Hyacinth S. Nwana publicou o influente paper Software Agents: An Overview, que forneceu a classificação mais abrangente de agentes até hoje. Sua tipologia incluía atributos como autonomia, habilidade social, reatividade, proatividade, aprendizado e mobilidade, e diferenciava entre agentes de software e construções tradicionais de software.

Nwana também ofereceu uma definição agora amplamente aceita, parafraseada: Um agente de software é um programa de computador baseado em software que atua para um usuário ou outro programa em uma relação de agência, que deriva da noção de delegação.

Essa formalização foi fundamental na transição de agentes de software de construções teóricas para aplicativos do mundo real. Isso deu origem a uma geração de sistemas baseados em agentes em áreas como telecomunicações, automação de fluxo de trabalho e assistentes inteligentes.

O trabalho de Nwana está no ponto de convergência das primeiras pesquisas de IA distribuída e das arquiteturas operacionais dos agentes modernos. É uma ponte crucial entre a teoria cognitiva dos agentes e sua implantação prática nos sistemas atuais.

Uma linha do tempo paralela: o surgimento de grandes modelos de linguagem

Enquanto as estruturas de agentes evoluíam, uma revolução paralela e convergente estava acontecendo no processamento de linguagem natural e no aprendizado de máquina:

- 2017 — transformers: O paper Attention Is All You Need (Vaswani et al. 2017) apresentou a arquitetura do transformador, que melhorou drasticamente a forma como as máquinas processam e geram linguagem.
- 2022 - ChatGPT: A OpenAI lançou uma interface baseada em bate-papo para o GPT-3.5 chamada ChatGPT, que permitiu uma conversa natural e interativa com um sistema de IA de uso geral.
- 2023 — código aberto LLMs: os lançamentos do Llama, Falcon e Mistral tornaram modelos poderosos amplamente acessíveis e aceleraram o desenvolvimento de estruturas de agentes em ambientes corporativos e de código aberto.

Essas inovações transformaram os modelos de linguagem em mecanismos de raciocínio capazes de analisar o contexto, planejar ações e encadear respostas, além de LLMs se tornarem os principais facilitadores de agentes de software inteligentes.

Os cronogramas convergem: o surgimento da IA agente

2023-2024 — plataformas de agentes de nível corporativo

A convergência de arquiteturas de agentes de software distribuídos e baseadas em transformadores LLMs culminou no surgimento da IA agente.

- O Amazon Bedrock Agents introduziu uma forma totalmente gerenciada de criar agentes de software orientados por metas e usando modelos básicos do Amazon Bedrock.
- O Model Context Protocol (MCP) da Anthropic definiu um método para grandes modelos de linguagem acessarem e interagirem com ferramentas, ambientes e memória externos. Isso é fundamental para um comportamento contextual, persistente e autônomo.

Esses dois marcos representam a síntese de agência e inteligência. Os agentes não estavam mais limitados a fluxos de trabalho estáticos ou automação rígida. Agora, eles podiam raciocinar em várias etapas, coordenar com ferramentas e APIs manter o estado contextual e aprender e se adaptar ao longo do tempo.

De janeiro a junho de 2025 — capacidades corporativas expandidas

No primeiro semestre de 2025, o cenário de IA agente se expandiu significativamente com novos recursos corporativos. Em fevereiro de 2025, a Anthropic lançou o Claude 3.7 Sonnet, que foi o primeiro modelo de raciocínio híbrido no mercado, e a especificação MCP ganhou ampla adoção.

Assistentes de codificação de IA, como Amazon Q Developer, Cursor e MCP WindSurf integrado, para padronizar a geração de código, a análise de repositórios e os fluxos de trabalho de desenvolvimento. A versão do MCP de março de 2025 introduziu recursos significativos prontos para uso corporativo, incluindo integração de segurança OAuth 2.1, tipos de recursos expandidos para acesso diversificado a dados e opções aprimoradas de conectividade por meio de HTTP Streamable. Com base nessa base, AWS anunciou em maio de 2025 que estava se juntando ao comitê diretor do MCP e contribuindo para novas capacidades de agent-to-agent comunicação. Isso fortalece ainda mais a posição do protocolo como um padrão do setor para interoperabilidade de IA agente.

Em maio de 2025, AWS fortaleceu as opções dos clientes para criar fluxos de trabalho de IA agentes ao abrir o código-fonte da estrutura Strands Agents. Essa estrutura independente do provedor e independente do modelo permite que os desenvolvedores usem modelos básicos em todas as plataformas, mantendo a profunda integração de serviços. AWS Conforme destacado no blog de código AWS aberto, a Strands Agents segue uma filosofia de design que prioriza o modelo, que coloca os modelos básicos no centro da inteligência dos agentes. Isso torna mais fácil para os clientes criar e implantar agentes de IA sofisticados para seus casos de uso específicos.

Emergência — IA agente

A evolução dos agentes de software, desde as ideias iniciais de autonomia até a orquestração moderna baseada em LLM, foi longa e complexa. O que começou com a visão de Oliver Selfridge de perceber programas se transformou em um ecossistema robusto de agentes de software inteligentes, conscientes do contexto e orientados por metas que podem colaborar, se adaptar e raciocinar.

A convergência da inteligência artificial distribuída (DAI) e da IA generativa baseada em transformadores marca o início de uma nova era na qual os agentes de software não são mais apenas ferramentas, mas atores autônomos em sistemas inteligentes.

A IA agente representa a próxima evolução em sistemas de software. Ele fornece uma classe de agentes inteligentes que são autônomos, assíncronos e agentes, que podem agir com intenção delegada e operar propositalmente em ambientes dinâmicos e distribuídos. A inteligência artificial da Agentic unifica o seguinte:

- A linhagem arquitetônica de sistemas multiagentes e o modelo de ator
- O modelo cognitivo de perceber, raciocinar, agir
- O poder generativo dos transformadores LLMs e
- A flexibilidade operacional da computação nativa em nuvem e sem servidor

Agentes de software para agenciar a IA

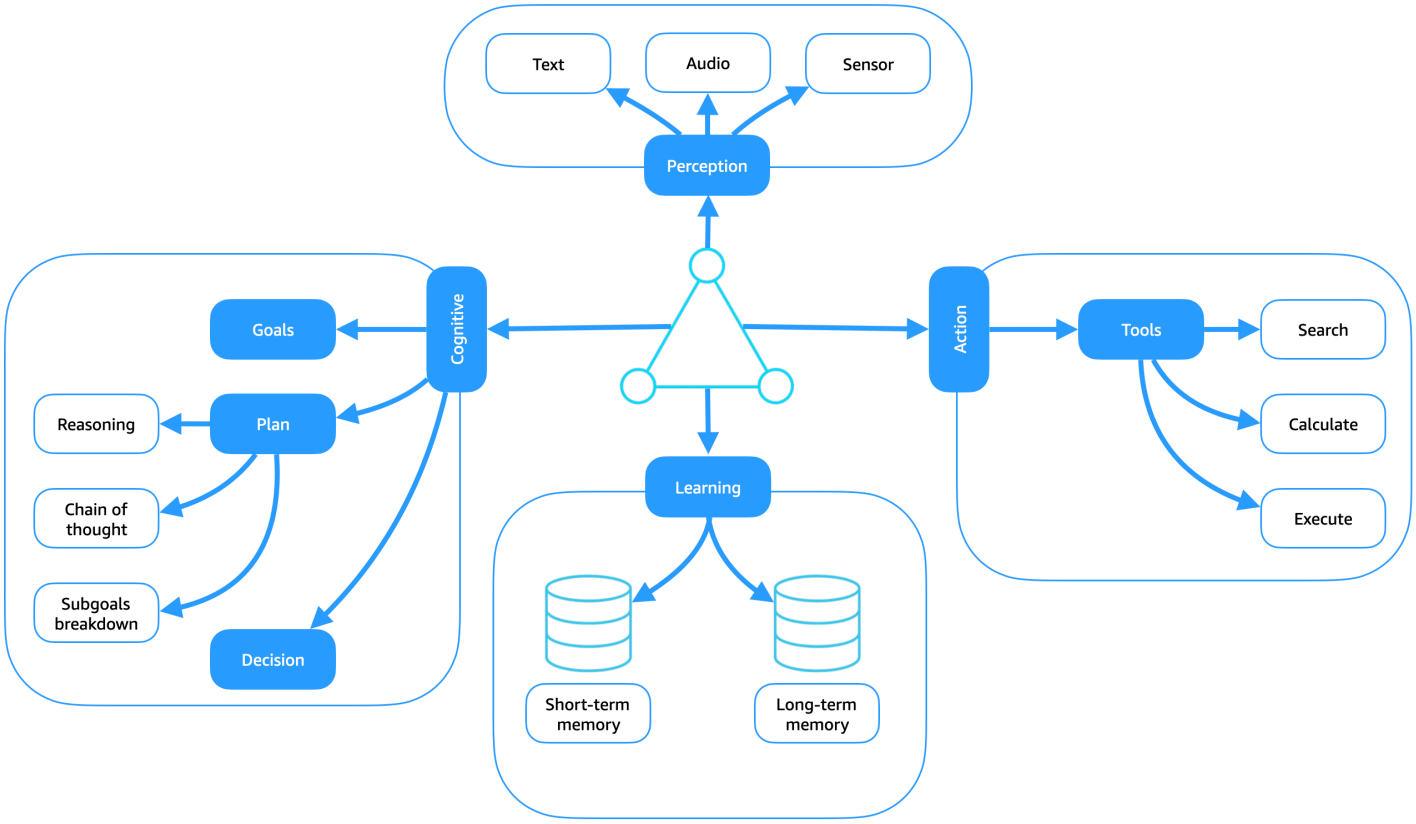
Agentes de software são entidades digitais autônomas projetadas para perceber seu ambiente, raciocinar sobre seus objetivos e agir de acordo. Diferentemente dos programas de software tradicionais que seguem uma lógica fixa, os agentes adaptam seu comportamento com base em entradas contextuais e estruturas de decisão. Isso os torna ideais para ambientes dinâmicos e distribuídos, como sistemas nativos da nuvem, robótica, automação inteligente e, agora, orquestração generativa de IA.

Esta seção apresenta os principais componentes dos agentes de software e explica como esses componentes interagem nas arquiteturas tradicionais com base no modelo de perceber, raciocinar e agir. Ele discute como a IA generativa, particularmente os grandes modelos de linguagem (LLMs), transformou a forma como os agentes de software raciocinam e planejam. Isso marca uma mudança fundamental dos sistemas baseados em regras para a inteligência aprendida e orientada por dados da IA agente.

Nesta seção

Elementos básicos dos agentes de software

O diagrama a seguir apresenta os principais módulos funcionais encontrados na maioria dos agentes inteligentes. Cada componente contribui para a capacidade do agente de operar de forma autônoma em ambientes complexos.



No contexto do ciclo de perceber, raciocinar e agir, a capacidade de raciocínio de um agente é distribuída em seus módulos cognitivo e de aprendizagem. Por meio da integração da memória e do aprendizado, o agente desenvolve um raciocínio adaptativo baseado em experiências passadas. À medida que o agente age em seu ambiente, ele cria um ciclo de feedback emergente: cada ação influencia as percepções futuras, e a experiência resultante é incorporada à memória e aos modelos internos por meio do módulo de aprendizagem. Esse ciclo contínuo de percepção, raciocínio e ação permite que o agente melhore com o tempo e complete o ciclo completo de perceber, raciocinar e agir.

Módulo de percepção

O módulo de percepção permite que o agente interaja com seu ambiente por meio de diversas modalidades de entrada, como texto, áudio e sensores. Essas entradas formam os dados brutos nos quais todo raciocínio e ação se baseiam. As entradas de texto podem incluir solicitações em linguagem natural, comandos estruturados ou documentos. As entradas de áudio abrangem instruções faladas ou sons ambientais. As entradas do sensor incluem dados físicos, como feeds visuais, sinais de movimento ou coordenadas GPS. A função principal da percepção é extrair características e representações significativas desses dados brutos. Isso permite que o agente construa uma compreensão precisa e acionável de seu contexto atual. O processo pode envolver

extração de características, reconhecimento de objetos ou eventos e interpretação semântica, e constitui a primeira etapa crítica no ciclo de perceber, raciocinar e agir. A percepção eficaz garante que o raciocínio e a tomada de decisões posteriores sejam baseados em uma consciência situacional relevante e atualizada.

Módulo cognitivo

O módulo cognitivo serve como o núcleo deliberativo do agente de software. É responsável por interpretar as percepções, formar a intenção e orientar o comportamento proposital por meio de planejamento e tomada de decisões orientados por metas. Esse módulo transforma as entradas em processos estruturados de raciocínio, o que permite que o agente opere intencionalmente em vez de reativamente. Esses processos são gerenciados por meio de três submódulos principais: metas, planejamento e tomada de decisão.

Submódulo de metas

O submódulo de metas define a intenção e a direção do agente. As metas podem ser explícitas (por exemplo, “navegar até um local” ou “enviar um relatório”) ou implícitas (por exemplo, “maximizar o engajamento do usuário” ou “minimizar a latência”). Eles são fundamentais para o ciclo de raciocínio do agente e fornecem um estado alvo para seu planejamento e decisões.

O agente avalia continuamente o progresso em direção a suas metas e pode repriorizar ou regenerar metas com base em novas percepções ou aprendizado. Essa consciência da meta mantém o agente adaptável em ambientes dinâmicos.

Submódulo de planejamento

O submódulo de planejamento constrói estratégias para atingir os objetivos atuais do agente. Ele gera sequências de ações, decompõe tarefas hierarquicamente e seleciona planos predefinidos ou gerados dinamicamente.

Para operar de forma eficaz em ambientes não determinísticos ou em mudança, o planejamento não é estático. Agentes modernos podem gerar sequências de cadeia de pensamento, introduzir metas secundárias como etapas intermediárias e revisar planos em tempo real quando as condições mudam.

Esse submódulo se conecta estreitamente à memória e ao aprendizado e permite que o agente refine seu planejamento ao longo do tempo com base nos resultados anteriores.

Decision-making submódulo

O submódulo de tomada de decisão avalia os planos e ações disponíveis para selecionar a próxima etapa mais apropriada. Ele integra informações da percepção, do plano atual, das metas do agente e do contexto ambiental.

Decision-making contas para:

- Trade-offs entre objetivos conflitantes
- Limites de confiança (por exemplo, incerteza na percepção)
- Consequências das ações
- A experiência aprendida do agente

Dependendo da arquitetura, os agentes podem confiar em raciocínio simbólico, heurística, aprendizado por reforço ou modelos de linguagem (LLMs) para tomar decisões informadas. Esse processo mantém o comportamento do agente consciente do contexto, alinhado às metas e adaptável.

Módulo de ação

O módulo de ação é responsável por executar as decisões selecionadas pelo agente e interagir com o mundo externo ou sistemas internos para produzir efeitos significativos. Representa a fase de Ação do ciclo de perceber, raciocinar e agir, onde a intenção é transformada em comportamento.

Quando o módulo cognitivo seleciona uma ação, o módulo de ação coordena a execução por meio de submódulos especializados, onde cada submódulo se alinha ao ambiente integrado do agente:

- Atuação física: para agentes incorporados em sistemas robóticos ou dispositivos de IoT, esse submódulo traduz decisões em movimentos físicos do mundo real ou instruções em nível de hardware.

  Exemplos: dirigir um robô, acionar uma válvula, ligar um sensor.
- Interação integrada: esse submódulo lida com ações não físicas, mas visíveis externamente, como interagir com sistemas de software, plataformas ou APIs.

  Exemplos: enviar um comando para um serviço de nuvem, atualizar um banco de dados, enviar um relatório chamando uma API.

- Invocação de ferramentas: os agentes geralmente ampliam seus recursos usando ferramentas especializadas para realizar subtarefas, como as seguintes:
  - Pesquisa: consultar fontes de conhecimento estruturadas ou não estruturadas
  - Resumindo: comprimindo entradas de texto grandes em visões gerais de alto nível
  - Cálculo: realizando computação lógica, numérica ou simbólica

  A invocação de ferramentas permite uma composição complexa de comportamentos por meio de habilidades modulares que podem ser chamadas.

Módulo de aprendizagem

O módulo de aprendizado permite que os agentes se adaptem, generalizem e melhorem ao longo do tempo com base na experiência. Ele apóia o processo de raciocínio refinando continuamente os modelos, estratégias e políticas de decisão internos do agente usando o feedback da percepção e da ação.

Este módulo opera em coordenação com a memória de curto e longo prazo:

- Short-term memória: armazena contexto transitório, como estado do diálogo, informações da tarefa atual e observações recentes. Isso ajuda o agente a manter a continuidade nas interações e tarefas.
- Long-term memória: codifica o conhecimento persistente de experiências passadas, incluindo metas previamente encontradas, resultados de ações e estados ambientais. Long-term a memória permite que o agente reconheça padrões, reutilize estratégias e evite a repetição de erros.

Modos de aprendizagem

O módulo de aprendizado oferece suporte a uma variedade de paradigmas, como aprendizado supervisionado, não supervisionado e por reforço, que oferecem suporte a diferentes ambientes e funções de agentes:

- Aprendizado supervisionado: atualiza modelos internos com base em exemplos rotulados, geralmente de feedback humano ou conjuntos de dados de treinamento.

  Exemplo: aprender a classificar a intenção do usuário com base em conversas anteriores.
- Aprendizado não supervisionado: identifica padrões ou estruturas ocultas nos dados sem rótulos explícitos.

Exemplo: agrupamento de sinais ambientais para detectar anomalias.

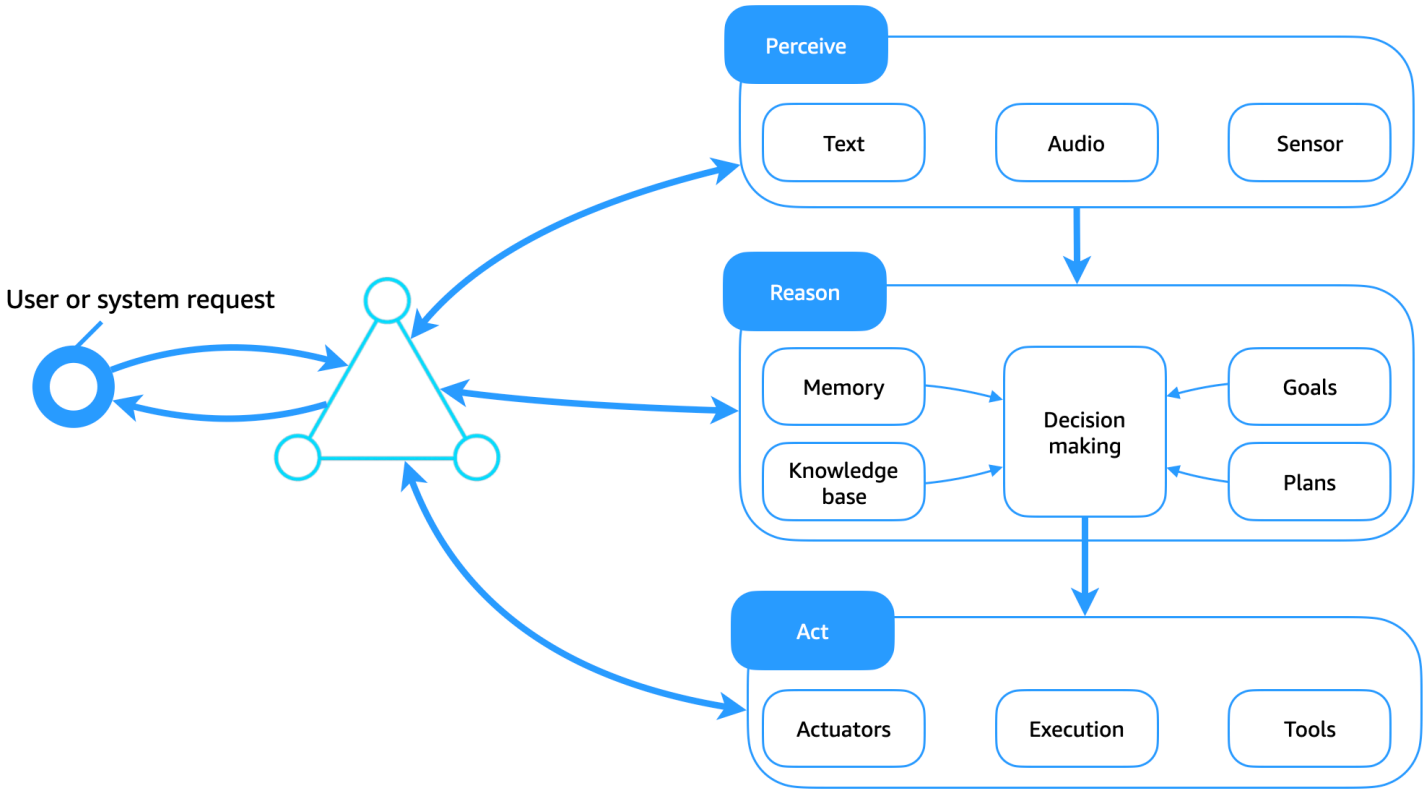
- Aprendizado por reforço: otimiza o comportamento por meio de tentativa e erro, maximizando a recompensa cumulativa em ambientes interativos.

  Exemplo: aprender qual estratégia leva à conclusão mais rápida da tarefa.

O aprendizado se integra perfeitamente ao módulo cognitivo do agente. Ele refina as estratégias de planejamento com base em resultados passados, aprimora a tomada de decisões por meio da avaliação do sucesso histórico e melhora continuamente o mapeamento entre percepção e ação. Por meio desse ciclo fechado de aprendizado e feedback, os agentes evoluem além da execução reativa para se tornarem sistemas de autoaperfeiçoamento capazes de se adaptar a novas metas, condições e contextos ao longo do tempo.

Arquitetura tradicional de agentes: perceber, raciocinar, agir

O diagrama a seguir ilustra como os blocos de construção discutidos na seção anterior operam sob o ciclo de perceber, raciocinar e agir.



Módulo Perceber

O módulo de percepção atua como a interface sensorial do agente com o mundo externo. Ele transforma a entrada ambiental bruta em representações estruturadas que informam o raciocínio. Isso inclui lidar com dados multimodais, como texto, áudio ou sinais de sensores.

- A entrada de texto pode vir de comandos, documentos ou diálogos do usuário.
- A entrada de áudio inclui instruções faladas ou sons ambientais.
- A entrada do sensor captura sinais do mundo real, como movimento, feeds visuais ou GPS.

Quando a entrada bruta é ingerida, o processo de percepção realiza a extração de características, seguida pelo reconhecimento de objetos ou eventos e interpretação semântica para criar um modelo significativo da situação atual. Esses resultados fornecem contexto estruturado para a tomada de decisões posteriores e ancoram o raciocínio do agente em observações do mundo real.

Módulo Reason

O módulo reason é o núcleo cognitivo do agente. Ele avalia o contexto, formula a intenção e determina as ações apropriadas. Este módulo orquestra o comportamento orientado por metas usando tanto o conhecimento aprendido quanto o raciocínio.

O módulo Reason consiste em submódulos totalmente integrados:

- Memória: mantém o estado do diálogo, o contexto da tarefa e o histórico episódico nos formatos de curto e longo prazo.
- Base de conhecimento: fornece acesso a regras simbólicas, ontologias ou modelos aprendidos (como incorporações, fatos e políticas).
- Metas e planos: define os resultados desejados e constrói estratégias de ação para alcançá-los. As metas podem ser atualizadas dinamicamente e os planos podem ser modificados de forma adaptativa com base no feedback.
- Decision-making: atua como o mecanismo central de arbitragem, ponderando as opções, avaliando as compensações e selecionando a próxima ação. Esse submódulo leva em consideração os limites de confiança, o alinhamento de metas e as restrições contextuais.

Juntos, esses componentes permitem que o agente raciocine sobre seu ambiente, atualize crenças, selecione caminhos e se comporte de maneira coerente e adaptativa. O módulo de razão fecha a lacuna entre percepção e comportamento.

Módulo Act

O módulo act executa a decisão selecionada pelo agente fazendo interface com o ambiente digital ou físico para realizar tarefas. É aqui que a intenção se torna ação.

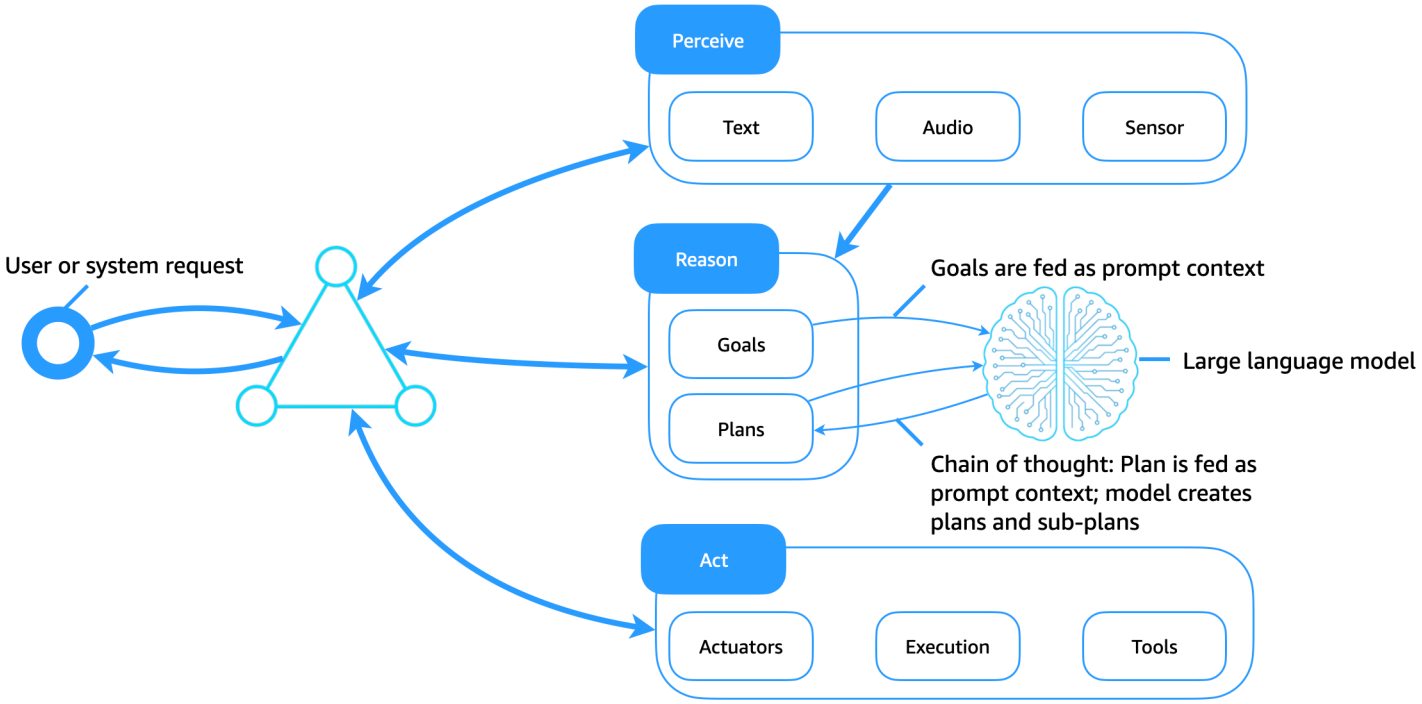
Este módulo inclui três canais funcionais:

- Atuadores: para agentes que têm presença física (como robôs e dispositivos de IoT), controlam as interações em nível de hardware, como movimento, manipulação ou sinalização.
- Execução: lida com ações baseadas em software, incluindo invocação de APIs, envio de comandos e atualização de sistemas.
- Ferramentas: permite recursos funcionais, como pesquisa, resumo, execução de código, cálculo e manuseio de documentos. Essas ferramentas geralmente são dinâmicas e sensíveis ao contexto, o que amplia a utilidade do agente.

As saídas do módulo de atuação retornam ao ambiente e fecham o circuito. Esses resultados são percebidos pelo agente novamente. Eles atualizam o estado interno do agente e informam decisões futuras, completando assim o ciclo de perceber, raciocinar e agir.

Agentes generativos de IA: substituindo a lógica simbólica por LLMs

O diagrama a seguir ilustra como os modelos de linguagem grande (LLMs) agora servem como um núcleo cognitivo flexível e inteligente para agentes de software. Em contraste com os sistemas lógicos simbólicos tradicionais, que dependem de bibliotecas de planos estáticos e regras codificadas manualmente, os LLMs permitem o raciocínio adaptativo, o planejamento contextual e o uso dinâmico de ferramentas, que transformam a forma como os agentes percebem, raciocinam e agem.



Principais aprimoramentos

Essa arquitetura aprimora a arquitetura tradicional do agente da seguinte forma:

- LLMs como mecanismos cognitivos: metas, planos e consultas são passados para o modelo como contexto imediato. O LLM gera caminhos de raciocínio (como cadeia de pensamento), decompõe tarefas em sub-metas e decide as próximas ações.
- Uso de ferramentas por meio de solicitação: os LLMs podem ser direcionados por meio de agentes de uso de ferramentas ou por meio de solicitações de raciocínio e ação (ReAct) para chamar APIs e pesquisar, consultar, calcular e interpretar saídas.
- Context-aware planejamento: os agentes geram ou revisam planos dinamicamente com base na meta atual, no ambiente de entrada e no feedback do agente, sem exigir bibliotecas de planos codificadas.
- Contexto imediato como memória: em vez de usar bases de conhecimento simbólicas, os agentes codificam memória, planos e metas como tokens imediatos que são passados para o modelo.
- Aprendizagem por meio de aprendizado rápido e contextual: os LLMs adaptam comportamentos por meio de engenharia imediata, o que reduz a necessidade de reciclagem explícita ou bibliotecas de planos rígidos.

Obtendo memória de longo prazo em LLM-based agentes

Ao contrário dos agentes tradicionais, que armazenavam memória de longo prazo em bases de conhecimento estruturadas, os agentes generativos de IA devem trabalhar dentro das limitações da janela de contexto dos LLMs. Para ampliar a memória e apoiar a inteligência persistente, os agentes generativos de IA usam várias técnicas complementares: armazenamento de agentes, Retrieval-Augmented geração (RAG), aprendizado contextual, encadeamento imediato e pré-treinamento.

Armazenamento de agentes: memória externa de longo prazo

O estado do agente, o histórico do usuário, as decisões e os resultados são armazenados em um armazenamento de memória do agente de longo prazo (como um banco de dados vetorial, armazenamento de objetos ou armazenamento de documentos). Memórias relevantes são recuperadas sob demanda e injetadas no contexto do prompt do LLM em tempo de execução. Isso cria um loop de memória persistente, em que o agente mantém a continuidade entre sessões, tarefas ou interações.

TRAPO

O RAG aprimora o desempenho do LLM combinando o conhecimento recuperado com os recursos generativos. Quando uma meta ou consulta é emitida, o agente pesquisa um índice de recuperação (por exemplo, por meio de uma pesquisa semântica de documentos, conversas anteriores ou conhecimento estruturado). Os resultados recuperados são anexados ao prompt do LLM, que fundamenta a geração em fatos externos ou contexto personalizado. Esse método amplia a memória efetiva do agente e melhora a confiabilidade e a exatidão factual.

In-context aprendizado e encadeamento imediato

Os agentes mantêm a memória de curto prazo usando o contexto de token na sessão e o encadeamento estruturado de prompts. Elementos contextuais, como o plano atual, os resultados das ações anteriores e o status do agente, são passados entre as chamadas para orientar o comportamento.

Pré-treinamento e ajuste contínuos

Para agentes específicos de domínio, os LLMs podem ser continuados pré-treinados em coleções personalizadas, como registros, dados corporativos ou documentação do produto. Como alternativa, o ajuste fino de instruções ou o aprendizado por reforço a partir do feedback humano (RLHF) podem incorporar um comportamento semelhante ao de um agente diretamente no modelo. Isso muda os

padrões de raciocínio da lógica do pronto-tempo para a representação interna do modelo, reduz o tamanho do prompt e melhora a eficiência.

Benefícios combinados em IA agêntica

Essas técnicas, quando usadas em conjunto, permitem que agentes generativos de IA:

- Mantenha a consciência contextual ao longo do tempo.
- Adapte o comportamento com base no histórico ou nas preferências do usuário.
- Tome decisões usando conhecimento atualizado, factual ou privado.
- Dimensione para casos de uso corporativos com comportamentos persistentes, compatíveis e explicáveis.

Ao aumentar os LLMs com memória externa, camadas de recuperação e treinamento contínuo, os agentes podem atingir um nível de continuidade e propósito cognitivos que não poderiam ser alcançados anteriormente apenas por meio de sistemas simbólicos.

Comparando a IA tradicional com agentes de software e IA agente

A tabela a seguir fornece uma comparação detalhada da IA tradicional, dos agentes de software e da IA agente.

Característica	IA tradicional	Agentes de software	IA agente
Exemplos	Filtros de spam, classificadores de imagens, mecanismos de recomendação	Chatbots, agendadores de tarefas, agentes de monitoramento	Assistentes de IA, agentes de desenvolvimento autônomos, orquestrações LLM multiagentes
Modelo de execução	Batch ou síncrono	Event-driven ou agendado	Assíncrono, orientado por eventos e orientado por metas
Autonomia	Limitado; geralmente requer orquestração humana ou externa	Médio; opera de forma independe	Alto; age de forma independente com

Característica	IA tradicional	Agentes de software	IA agente
		nte dentro de limites predefinidos	estratégias adaptativas
Reatividade	Reativo aos dados de entrada	Reativo ao ambiente e aos eventos	Reativo e proativo; antecipa e inicia ações
Proatividade	Raro	Presente em alguns sistemas	Atributo principal ; impulsiona o comportamento direcionado a um objetivo
Comunicação	Mínimo; geralmente autônomo ou API-bound	Inter-agent ou mensagens entre agente e humano	Interação multiagente avançada e humano-in-the-loop
Decision-making	Somente inferência de modelo (classificação, previsão e assim por diante)	Raciocínio simbólico ou decisões baseadas em regras ou roteirizadas	Raciocínio contextual, baseado em metas e dinâmico (frequentemente) LLM-enhanced
Intenção delegada	Não; executa tarefas definidas diretamente pelo usuário	Parcial; age em nome de usuários ou sistemas com escopo limitado	Sim; age com metas delegadas, geralmente entre serviços, usuários ou sistemas
Aprendizagem e adaptação	Geralmente centrado em modelos (por exemplo, treinamento de ML)	Às vezes adaptável	Aprendizado, memória ou raciocínio incorporados (por exemplo, feedback, autocorreção)
Agência	Nenhuma; ferramentas para humanos	Implícito ou básico	Explícito; opera com propósito, metas e autodireção

Característica	IA tradicional	Agentes de software	IA agente
Consciência do contexto	Baixo; sem estado ou baseado em instantâneos	Moderado; algum rastreamento de estado	Alto; usa modelos de memória, contexto situacional e ambiente
Perfil de infraestrutura	Incorporado em aplicativos ou pipelines de análise	Componente de middleware ou camada de serviço	Malha de agentes composta integrada a sistemas de nuvem, sem servidor ou de borda

Em resumo:

- A IA tradicional é centrada em ferramentas e funcionalmente restrita. Ele se concentra na previsão ou classificação.
- Os agentes de software tradicionais introduzem autonomia e comunicação básica, mas geralmente são estáticos ou vinculados a regras.
- A IA agente reúne autonomia, assincronia e agência. Ele permite que entidades inteligentes e orientadas por metas possam raciocinar, agir e se adaptar em sistemas complexos. Isso torna a IA agente ideal para o futuro nativo da nuvem. AI-driven

Próximas etapas

Este guia discutiu a história e os fundamentos da IA agente, que representa a evolução dos agentes de software tradicionais em sistemas autônomos e inteligentes que são alimentados pela IA generativa. Ele descreveu como os primeiros agentes de software seguiram regras e lógicas predefinidas para automatizar tarefas dentro de limites fixos e explicou como a IA agêntica se baseia nessa base ao incorporar grandes modelos de linguagem, que permitem que os agentes raciocinem, aprendam e se adaptem dinamicamente em ambientes abertos.

Você pode explorar a IA agente em profundidade analisando as seguintes publicações desta série:

- A operacionalização da IA agente AWS fornece uma estratégia organizacional para transformar a IA agente de experimentos isolados em uma infraestrutura de geração de valor em escala corporativa.
- Os padrões e fluxos de trabalho da Agentic AI em AWS discute os planos fundamentais e as construções modulares usados para projetar, compor e orquestrar agentes de IA orientados a objetivos.
- As estruturas, protocolos e ferramentas da Agentic AI AWS abrangem os fundamentos de software, os kits de ferramentas e os protocolos a serem considerados ao criar suas soluções de IA agente.
- A criação de arquiteturas sem servidor para IA agente em AWS discute as arquiteturas sem servidor como a base natural das cargas de trabalho modernas de IA e descreve como você pode criar arquiteturas sem servidor nativas de IA no. Nuvem AWS
- A criação de arquiteturas multilocatárias para IA agente AWS descreve o uso de agentes de IA em configurações de vários locatários, incluindo considerações de hospedagem, modelos de implantação e planos de controle.

Recursos

Para obter mais informações sobre os conceitos discutidos neste guia, consulte os guias e artigos a seguir.

AWS referências

- Agentes do Amazon Bedrock
- Amazon Q Developer
- SDK de agentes Strands

Outras referências

- Hewitt, Carl, Peter Bishop e Richard Steiger. “Um formalismo de ATOR modular universal para inteligência artificial.” Anais da 3ª Conferência Internacional Conjunta sobre Inteligência Artificial (1973): 235-245. https://www.ijcai. org/Proceedings/73/Papers/027B.pdf
- Lesser, Victor R., publicações relevantes (veja a lista completa):
  - Lesser, Victor R. e Daniel D. Corkill. “Sistemas distribuídos cooperativos e funcionalmente precisos.” IEEE Transactions on Systems, Man and Cybernetics 11, nº 1 (1981): 81-96. https://ieeexplore.ieee. org/abstract/document/4308 581
  - Decker, Keith S. e Victor R. Lesser. “Comunicação a serviço da coordenação”. Workshop da AAAI sobre planejamento para comunicação 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 e Daniel D. Corkill. “Tendências na solução cooperativa de problemas distribuídos”. Transações do IEEE sobre engenharia de conhecimento e dados (1989). http://mas.cs.umass. edu/Documents/ieee-tkde89.pdf
  - Durfee, Edmund H., V.R. Lesser e D.D. Corkill, “Inteligência Artificial Distribuída”. Cooperação por meio da comunicação em uma rede distribuída de solução de problemas (1987): 29-58. https://www.academia. edu/download/79885643/durf94_1.pdf
  - Lasri, Brigitte, Hassan Lasri, Susan Lander e Victor Lesser. “Um modelo genérico para agentes de negociação inteligentes.” Revista Internacional de Sistemas de Informação Cooperativa 01, nº 02 (1992): 291-317. https://doi.org/10.1142/S0218215792000210

  - Lander, Susan E. e Victor R. Lesser. “Compreendendo o papel da negociação na busca distribuída entre agentes heterogêneos.” IJCAI'93: Anais da 13ª conferência conjunta internacional sobre inteligência artificial (1993): 438-444. https://www.ijcai. org/Proceedings/93-1/Papers/062.pdf
  - Lander, Susan, Victor R. Lesser e Margaret E. Connell. “Estratégias de resolução de conflitos para agentes especializados cooperantes” CKBS'90: Anais da Conferência Internacional de Trabalho sobre Sistemas Cooperativos Baseados no Conhecimento (outubro de 1990): 183-200. https://doi.org/10.1007/978-1-4471-1831-2_10
  - Prasad, M. V. Nagendra, Victor Lesser e Susan E. Lander. “Experimentos de aprendizagem em um sistema multiagente heterogêneo.” Workshop do IJCAI-95 sobre adaptação e aprendizagem em 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: uma visão geral”. Revisão de Engenharia do Conhecimento 11, no. 3 (outubro/novembro de 1996): 205-244. https://teaching.shu.ac. uk/aces/rh1/elearning/multiagents/introduction/nwana.pdf
- Selfridge, Oliver G. “Pandemônio: um paradigma para a aprendizagem”. Mecanização dos processos de pensamento: Anais de um simpósio realizado no Laboratório Nacional de Física 1 (1959): 511—529. https://aitopics. org/download/classics: 504E1BAC
- Vaswani, Ashish, Noam Shazer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser e Illia Polosukhin. “Atenção é tudo que você precisa.” Anais da 31ª Conferência sobre Sistemas de Processamento de Informação Neural (NIPS). Avanços nos sistemas de processamento de informações neurais 30 (2017): 5998-6008. https://proceedings.neurips. cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- Wooldridge, Michael e Nicholas R. Jennings. “Agentes inteligentes: teoria e prática”. Revisão de Engenharia do Conhecimento 10, no. 2 (janeiro de 1995): 115-152. https://www.cs.cmu.edu/ ~motionplanning/papers/sbp_papers/integrated1/woodridge _intelligent_agents.pdf

Histórico do documento

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

Alteração	Descrição	Data
Publicação inicial	—	14 de julho de 2025

AWS Glossário de orientação prescritiva

A seguir estão os termos comumente usados em estratégias, guias e padrões fornecidos pela Orientação AWS Prescritiva. Para sugerir entradas, use o link Fornecer feedback no final do glossário.

Números

7 Rs

Sete estratégias comuns de migração para mover aplicações para a nuvem. Essas estratégias baseiam-se nos 5 Rs identificados pela Gartner em 2011 e consistem em:

- Refactor/re-architect — mova um aplicativo e modifique sua arquitetura aproveitando ao máximo os recursos nativos da nuvem para melhorar a agilidade, o desempenho e a escalabilidade. Isso normalmente envolve a portabilidade do sistema operacional e do banco de dados. Exemplo: migre seu banco de dados Oracle local para a Amazon PostgreSQL-Compatible Aurora Edition.
- Redefinir a plataforma (mover e redefinir [mover e redefinir (lift-and-reshape]):: mova uma aplicação para a nuvem e introduza algum nível de otimização a fim de aproveitar os recursos da nuvem. Exemplo: migrar seu banco de dados Oracle on-premises para o Amazon Relational Database Service (Amazon RDS) para Oracle na Nuvem AWS.
- Recomprar (drop and shop): mude para um produto diferente, normalmente migrando de uma licença tradicional para um modelo SaaS. Exemplo: Migre seu sistema de gerenciamento de relacionamento com o cliente (CRM) para o. Salesforce.com
- Redefinir a hospedagem (mover sem alterações [lift-and-shift]): mover uma aplicação para a nuvem sem fazer nenhuma alteração a fim de aproveitar os recursos da nuvem. Exemplo: migrar seu banco de dados Oracle on-premises para o Oracle em uma instância do EC2 na Nuvem AWS.
- Realocar (mover o hipervisor sem alterações [hypervisor-level lift-and-shift]): mover a infraestrutura para a nuvem sem comprar novo hardware, reescrever aplicações ou modificar suas operações existentes. Você migra servidores de uma plataforma on-premises para um serviço de nuvem para a mesma plataforma. Exemplo: Migrar um Microsoft Hyper-V aplicativo para o. AWS
- Reter (revisitar): mantenha as aplicações em seu ambiente de origem. Isso pode incluir aplicações que exigem grande refatoração, e você deseja adiar esse trabalho para um

momento posterior, e aplicações antigas que você deseja manter porque não há justificativa comercial para migrá-las.

- Retirar: desative ou remova aplicações que não são mais necessárias em seu ambiente de origem.

A

A2A () Agent-to-Agent

Um protocolo com estado para colaboração entre agentes, apoiando a delegação de tarefas e a transferência de estados.

ABAC

Consulte controle de acesso baseado em atributo.

serviços abstraídos

Veja serviços gerenciados.

ACID

Veja atomicidade, consistência, isolamento, durabilidade.

migração ativa-ativa

Um método de migração de banco de dados no qual os bancos de dados de origem e de destino são mantidos em sincronia (por meio de uma ferramenta de replicação bidirecional ou operações de gravação dupla), e ambos os bancos de dados lidam com transações de aplicações conectadas durante a migração. Esse método oferece suporte à migração em lotes pequenos e controlados, em vez de exigir uma substituição única. É mais flexível, mas exige mais trabalho do que a migração ativa-passiva.

migração ativa-passiva

Um método de migração de banco de dados em que os bancos de dados de origem e de destino são mantidos em sincronia, mas somente o banco de dados de origem manipula as transações das aplicações conectadas, enquanto os dados são replicados no banco de dados de destino. O banco de dados de destino não aceita nenhuma transação durante a migração.

Agente

Um sistema de IA que pode raciocinar, planejar e realizar ações de forma autônoma usando ferramentas para atingir metas.

Agente Ops

Práticas operacionais para criar, testar, implantar e executar agentes de IA na produção em grande escala.

AGGREGATE FUNCTION

Uma função SQL que opera em um grupo de linhas e calcula um único valor de retorno para o grupo. Exemplos de funções agregadas incluem `SUM` e `MAX`.

AI

Veja inteligência artificial.

AIOps

Veja operações de inteligência artificial.

anonimização

O processo de excluir permanentemente informações pessoais em um conjunto de dados. A anonimização pode ajudar a proteger a privacidade pessoal. Dados anônimos não são mais considerados dados pessoais.

antipadrões

Uma solução frequentemente usada para um problema recorrente em que a solução é contraproducente, ineficaz ou menos eficaz do que uma alternativa.

controle de aplicações

Uma abordagem de segurança que permite o uso somente de aplicações aprovadas para ajudar a proteger um sistema contra malware.

portfólio de aplicações

Uma coleção de informações detalhadas sobre cada aplicação usada por uma organização, incluindo o custo para criar e manter a aplicação e seu valor comercial. Essas informações são fundamentais para o processo de descoberta e análise de portfólio e ajudam a identificar e priorizar as aplicações a serem migradas, modernizadas e otimizadas.

inteligência artificial (IA)

O campo da ciência da computação que se dedica ao uso de tecnologias de computação para desempenhar funções cognitivas normalmente associadas aos humanos, como aprender, resolver problemas e reconhecer padrões. Para obter mais informações, consulte O que é inteligência artificial?

operações de inteligência artificial (AIOps)

O processo de usar técnicas de machine learning para resolver problemas operacionais, reduzir incidentes operacionais e intervenção humana e aumentar a qualidade do serviço. Para obter mais informações sobre como as AIOps são usadas na estratégia de migração para a AWS , consulte o guia de integração de operações.

criptografia assimétrica

Um algoritmo de criptografia que usa um par de chaves, uma chave pública para criptografia e uma chave privada para descriptografia. É possível compartilhar a chave pública porque ela não é usada na descriptografia, mas o acesso à chave privada deve ser altamente restrito.

atomicidade, consistência, isolamento, durabilidade (ACID)

Um conjunto de propriedades de software que garantem a validade dos dados e a confiabilidade operacional de um banco de dados, mesmo no caso de erros, falhas de energia ou outros problemas.

controle de acesso por atributo (ABAC)

A prática de criar permissões minuciosas com base nos atributos do usuário, como departamento, cargo e nome da equipe. Para obter mais informações, consulte ABAC AWS na documentação AWS Identity and Access Management (IAM).

fonte de dados autorizada

Um local onde você armazena a versão principal dos dados, que é considerada a fonte de informações mais confiável. Você pode copiar dados da fonte de dados autorizada para outros locais com o objetivo de processar ou modificar os dados, como anonimizá-los, redigi-los ou pseudonimizá-los.

Zona de disponibilidade

Um local distinto dentro de um Região da AWS que está isolado de falhas em outras zonas de disponibilidade e fornece conectividade de rede barata e de baixa latência a outras zonas de disponibilidade na mesma região.

AWS Estrutura de adoção da nuvem (AWS CAF)

Uma estrutura de diretrizes e melhores práticas AWS para ajudar as organizações a desenvolver um plano eficiente e eficaz para migrar com sucesso para a nuvem. AWS O CAF organiza a orientação em seis áreas de foco chamadas perspectivas: negócios, pessoas, governança, plataforma, segurança e operações. As perspectivas de negócios, pessoas e governança têm

como foco habilidades e processos de negócios; as perspectivas de plataforma, segurança e operações concentram-se em habilidades e processos técnicos. Por exemplo, a perspectiva das pessoas tem como alvo as partes interessadas que lidam com recursos humanos (RH), funções de pessoal e gerenciamento de pessoal. Nessa perspectiva, o AWS CAF fornece orientação para desenvolvimento, treinamento e comunicação de pessoas para ajudar a preparar a organização para a adoção bem-sucedida da nuvem. Para obter mais informações, consulte o site da AWS CAF e o whitepaper da AWS CAF.

AWS Estrutura de qualificação da carga de trabalho (AWS WQF)

Uma ferramenta que avalia as cargas de trabalho de migração do banco de dados, recomenda estratégias de migração e fornece estimativas de trabalho. AWS O WQF está incluído com AWS Schema Conversion Tool ()AWS SCT. Ela analisa esquemas de banco de dados e objetos de código, código de aplicações, dependências e características de performance, além de fornecer relatórios de avaliação.

B

bot malicioso

Um bot destinado a causar disrupção ou danos a indivíduos ou organizações.

BCP

Veja planejamento de continuidade de negócios

gráfico de comportamento

Uma visualização unificada e interativa do comportamento e das interações de recursos ao longo do tempo. É possível usar um gráfico de comportamento com o Amazon Detective para examinar tentativas de login malsucedidas, chamadas de API suspeitas e ações similares. Para obter mais informações, consulte Dados em um gráfico de comportamento na documentação do Detective.

sistema big-endian

Um sistema que armazena o byte mais significativo antes. Veja também endianness.

classificação binária

Um processo que prevê um resultado binário (uma de duas classes possíveis). Por exemplo, seu modelo de ML pode precisar prever problemas como “Este e-mail é ou não é spam?” ou “Este produto é um livro ou um carro?”

filtro de bloom

Uma estrutura de dados probabilística e eficiente em termos de memória que é usada para testar se um elemento é membro de um conjunto.

blue/green implantação

Uma estratégia de implantação em que você cria dois ambientes separados, mas idênticos. Você executa a versão atual da aplicação em um ambiente (azul) e a nova versão da aplicação no outro ambiente (verde). Essa estratégia ajuda você a reverter rapidamente com o mínimo de impacto.

bot

Uma aplicação de software que executa tarefas automatizadas na internet e simula a atividade ou interação humana. Alguns bots são úteis ou benéficos, como crawlers da web que indexam informações na internet. Outros bots, conhecidos como bots maliciosos, têm como objetivo causar disrupção ou danos a indivíduos ou organizações.

botnet

Redes de bots infectadas por malware e sob o controle de uma única parte, conhecidas como bot herder ou operador de bots. Os botnets são o mecanismo mais conhecido para escalar bots e seu impacto.

ramo

Uma área contida de um repositório de código. A primeira ramificação criada em um repositório é a ramificação principal. Você pode criar uma nova ramificação a partir de uma ramificação existente e, em seguida, desenvolver recursos ou corrigir bugs na nova ramificação. Uma ramificação que você cria para gerar um recurso é comumente chamada de ramificação de recurso. Quando o recurso estiver pronto para lançamento, você mesclará a ramificação do recurso de volta com a ramificação principal. Para obter mais informações, consulte Sobre filiais (GitHub documentação).

Acesso de emergência

Em circunstâncias excepcionais e por meio de um processo aprovado, um meio rápido para um usuário obter acesso a um Conta da AWS que ele normalmente não tem permissão para acessar. Para obter mais informações, consulte o indicador Implementar procedimentos de quebra de vidros na AWS Well-Architected orientação.

estratégia brownfield

A infraestrutura existente em seu ambiente. Ao adotar uma estratégia brownfield para uma arquitetura de sistema, você desenvolve a arquitetura de acordo com as restrições dos sistemas e da infraestrutura atuais. Se estiver expandindo a infraestrutura existente, poderá combinar as estratégias brownfield e greenfield.

cache do buffer

A área da memória em que os dados acessados com mais frequência são armazenados.

capacidade de negócios

O que uma empresa faz para gerar valor (por exemplo, vendas, atendimento ao cliente ou marketing). As arquiteturas de microsserviços e as decisões de desenvolvimento podem ser orientadas por recursos de negócios. Para obter mais informações, consulte a seção Organizados de acordo com as capacidades de negócios do whitepaper Executar microsserviços conteinerizados na AWS.

planejamento de continuidade de negócios (BCP)

Um plano que aborda o impacto potencial de um evento disruptivo, como uma migração em grande escala, nas operações e permite que uma empresa retome as operações rapidamente.

C

CAF

Veja AWS Cloud Adoption Framework.

implantação canário

O lançamento lento e incremental de uma versão para usuários finais. Quando estiver confiante, você implanta a nova versão e substitui a versão atual por completo.

CCoE

Veja Centro de Excelência da Nuvem.

CDC

Veja captura de dados de alteração.

captura de dados de alterações (CDC)

O processo de rastrear alterações em uma fonte de dados, como uma tabela de banco de dados, e registrar metadados sobre a alteração. É possível usar o CDC para várias finalidades, como auditar ou replicar alterações em um sistema de destino para manter a sincronização.

engenharia do caos

Introduzir intencionalmente falhas ou eventos disruptivos para testar a resiliência de um sistema. Você pode usar AWS Fault Injection Service (AWS FIS) para realizar experimentos que estressam suas AWS cargas de trabalho e avaliar sua resposta.

CI/CD

Veja integração e entrega contínuas.

classificação

Um processo de categorização que ajuda a gerar previsões. Os modelos de ML para problemas de classificação predizem um valor discreto. Os valores discretos são sempre diferentes uns dos outros. Por exemplo, um modelo pode precisar avaliar se há ou não um carro em uma imagem.

Desenvolvedor cidadão

Um usuário corporativo que cria aplicativos de IA usando plataformas sem code/low código sem habilidades técnicas especializadas.

criptografia no lado do cliente

Criptografia de dados localmente, antes que o alvo os AWS service (Serviço da AWS) receba.

Centro de Excelência da Nuvem (CCoE)

Uma equipe multidisciplinar que impulsiona os esforços de adoção da nuvem em toda a organização, incluindo o desenvolvimento de práticas recomendas de nuvem, a mobilização de recursos, o estabelecimento de cronogramas de migração e a liderança da organização em transformações em grande escala. Para obter mais informações, consulte as postagens do CCoE no blog de estratégia Nuvem AWS corporativa.

computação em nuvem

A tecnologia de nuvem normalmente usada para armazenamento de dados remoto e gerenciamento de dispositivos de IoT. A computação em nuvem é normalmente conectada à tecnologia de computação de borda.

modelo operacional em nuvem

Em uma organização de TI, o modelo operacional usado para criar, amadurecer e otimizar um ou mais ambientes de nuvem. Para obter mais informações, consulte Criar seu modelo operacional de nuvem.

estágios de adoção da nuvem

As quatro fases pelas quais as organizações normalmente passam ao migrar para a Nuvem AWS:

- Projeto: executar alguns projetos relacionados à nuvem para fins de prova de conceito e aprendizado
- Fundação: realizar investimentos fundamentais para escalar sua adoção da nuvem (por exemplo, criar uma zona de pouso, definir um CCoE, estabelecer um modelo de operações)
- Migração: migrar aplicações individuais
- Re-invention — Otimizando produtos e serviços e inovando na nuvem

Esses estágios foram definidos por Stephen Orban na postagem do blog The Journey Toward Cloud-First & the Stages of Adoption no blog Nuvem AWS Enterprise Strategy. Para obter informações sobre como eles se relacionam com a estratégia de AWS migração, consulte o guia de preparação para migração.

CMDB

Veja banco de dados de gerenciamento de configuração.

repositório de código

Um local onde o código-fonte e outros ativos, como documentação, amostras e scripts, são armazenados e atualizados por meio de processos de controle de versão. Os repositórios de nuvem comuns incluem o GitHub ou o Bitbucket Cloud. Cada versão do código é chamada de ramificação. Em uma estrutura de microsserviços, cada repositório é dedicado a uma única peça de funcionalidade. Um único CI/CD pipeline pode usar vários repositórios.

cache frio

Um cache de buffer que está vazio, não está bem preenchido ou contém dados obsoletos ou irrelevantes. Isso afeta a performance porque a instância do banco de dados deve ler da memória principal ou do disco, um processo que é mais lento do que a leitura do cache do buffer.

dados frios

Dados que raramente são acessados e geralmente são históricos. Ao consultar esse tipo de dados, consultas lentas geralmente são aceitáveis. Mover esses dados para níveis ou classes de armazenamento de baixo desempenho e menos caros pode reduzir os custos.

visão computacional (CV)

Um campo de IA que usa machine learning para analisar e extrair informações de formatos visuais, como vídeos e imagens digitais. Por exemplo, a Amazon SageMaker AI fornece algoritmos de processamento de imagem para CV.

desvio de configuração

Em uma workload, uma alteração de configuração em relação ao estado esperado. Isso pode fazer com que a workload se torne incompatível e, normalmente, é gradual e não intencional.

banco de dados de gerenciamento de configuração (CMDB)

Um repositório que armazena e gerencia informações sobre um banco de dados e seu ambiente de TI, incluindo componentes de hardware e software e suas configurações. Normalmente, os dados de um CMDB são usados no estágio de descoberta e análise do portfólio da migração.

pacote de conformidade

Uma coleção de AWS Config regras e ações de remediação que você pode montar para personalizar suas verificações de conformidade e segurança. Você pode implantar um pacote de conformidade como uma entidade única em uma Conta da AWS região ou em uma organização usando um modelo YAML. Para obter mais informações, consulte Pacotes de conformidade na documentação. AWS Config

integração contínua e entrega contínua (CI/CD)

O processo de automatizar os estágios de origem, criação, teste, preparação e produção do processo de lançamento do software. CI/CD é comumente descrito como um pipeline. CI/CD pode ajudá-lo a automatizar processos, melhorar a produtividade, melhorar a qualidade do código e entregar com mais rapidez. Para obter mais informações, consulte Benefícios da entrega contínua. CD também pode significar implantação contínua. Para obter mais informações, consulte Entrega contínua versus implantação contínua.

CV

Veja visão computacional.

D

dados em repouso

Dados estacionários em sua rede, por exemplo, dados que estão em um armazenamento.

classificação de dados

Um processo para identificar e categorizar os dados em sua rede com base em criticalidade e confidencialidade. É um componente crítico de qualquer estratégia de gerenciamento de riscos de segurança cibernética, pois ajuda a determinar os controles adequados de proteção e retenção para os dados. A classificação de dados é um componente do pilar de segurança na AWS Well-Architected Estrutura. Para obter mais informações, consulte Classificação de dados.

desvio de dados

Uma variação significativa entre os dados de produção e os dados usados para treinar um modelo de ML ou uma alteração significativa nos dados de entrada ao longo do tempo. O desvio de dados pode reduzir a qualidade geral, a precisão e a imparcialidade das previsões do modelo de ML.

dados em trânsito

Dados que estão se movendo ativamente pela sua rede, como entre os recursos da rede.

data mesh

Um framework de arquitetura que fornece propriedade de dados distribuída e descentralizada com gerenciamento e governança centralizados.

minimização de dados

O princípio de coletar e processar apenas os dados estritamente necessários. Praticar a minimização de dados no Nuvem AWS pode reduzir os riscos de privacidade, os custos e a pegada de carbono de sua análise.

perímetro de dados

Um conjunto de proteções preventivas em seu AWS ambiente que ajudam a garantir que somente identidades confiáveis acessem recursos confiáveis das redes esperadas. Para obter mais informações, consulte Construindo um perímetro de dados em. AWS

pré-processamento de dados

A transformação de dados brutos em um formato que seja facilmente analisado por seu modelo de ML. O pré-processamento de dados pode significar a remoção de determinadas colunas ou linhas e o tratamento de valores ausentes, inconsistentes ou duplicados.

proveniência dos dados

O processo de rastrear a origem e o histórico dos dados ao longo de seu ciclo de vida, por exemplo, como os dados foram gerados, transmitidos e armazenados.

titular dos dados

Um indivíduo cujos dados estão sendo coletados e processados.

data warehouse

Um sistema de gerenciamento de dados compatível com business intelligence, como analytics. Os data warehouses geralmente contêm grandes quantidades de dados históricos e geralmente são usados para consultas e análises.

linguagem de definição de dados (DDL)

Instruções ou comandos para criar ou modificar a estrutura de tabelas e objetos em um banco de dados.

linguagem de manipulação de dados (DML)

Instruções ou comandos para modificar (inserir, atualizar e excluir) informações em um banco de dados.

DDL

Veja linguagem de definição de banco de dados.

deep ensemble

A combinação de vários modelos de aprendizado profundo para gerar previsões. Os deep ensembles podem ser usados para produzir uma previsão mais precisa ou para estimar a incerteza nas previsões.

Aprendizado profundo

Um subcampo do ML que usa várias camadas de redes neurais artificiais para identificar o mapeamento entre os dados de entrada e as variáveis-alvo de interesse.

defesa completa

Uma abordagem de segurança da informação na qual uma série de mecanismos e controles de segurança são cuidadosamente distribuídos por toda a rede de computadores para proteger a confidencialidade, a integridade e a disponibilidade da rede e dos dados nela contidos. Ao adotar essa estratégia AWS, você adiciona vários controles em diferentes camadas da AWS Organizations estrutura para ajudar a proteger os recursos. Por exemplo, uma abordagem de defesa aprofundada pode combinar autenticação multifatorial, segmentação de rede e criptografia.

administrador delegado

Em AWS Organizations, um serviço compatível pode registrar uma conta de AWS membro para administrar as contas da organização e gerenciar as permissões desse serviço. Essa conta é chamada de administrador delegado para esse serviço Para obter mais informações e uma lista de serviços compatíveis, consulte Serviços que funcionam com o AWS Organizations na documentação do AWS Organizations .

implantação

O processo de criar uma aplicação, novos recursos ou correções de código disponíveis no ambiente de destino. A implantação envolve a implementação de mudanças em uma base de código e, em seguida, a criação e execução dessa base de código nos ambientes da aplicação

ambiente de desenvolvimento

Veja ambiente.

controle detectivo

Um controle de segurança projetado para detectar, registrar e alertar após a ocorrência de um evento. Esses controles são uma segunda linha de defesa, alertando você sobre eventos de segurança que contornaram os controles preventivos em vigor. Para obter mais informações, consulte Controles detectivos em Como implementar controles de segurança na AWS.

mapeamento do fluxo de valor de desenvolvimento (DVSM)

Um processo usado para identificar e priorizar restrições que afetam negativamente a velocidade e a qualidade em um ciclo de vida de desenvolvimento de software. O DVSM estende o processo de mapeamento do fluxo de valor originalmente projetado para práticas de manufatura enxuta. Ele se concentra nas etapas e equipes necessárias para criar e movimentar valor por meio do processo de desenvolvimento de software.

gêmeo digital

Uma representação virtual de um sistema real, como um prédio, fábrica, equipamento industrial ou linha de produção. Os gêmeos digitais oferecem suporte à manutenção preditiva, ao monitoramento remoto e à otimização da produção.

tabela de dimensões

Em um esquema em estrela, uma tabela menor que contém atributos de dados sobre dados quantitativos em uma tabela de fatos. Os atributos da tabela de dimensões geralmente são campos de texto ou números discretos que se comportam como texto. Esses atributos normalmente são usados para restringir consultas, filtrar e rotular conjuntos de resultados.

desastre

Um evento que impede que uma workload ou sistema cumpra seus objetivos de negócios em seu local principal de implantação. Esses eventos podem ser desastres naturais, falhas técnicas ou o resultado de ações humanas, como configuração incorreta não intencional ou ataque de malware.

Recuperação de desastres (RD)

A estratégia e o processo que você usa para minimizar o tempo de inatividade e a perda de dados causados por um disastre. Para obter mais informações, consulte Recuperação de desastres de cargas de trabalho em AWS: Recuperação na nuvem na AWS Well-Architected estrutura.

DML

Veja linguagem de manipulação de banco de dados.

design orientado por domínio

Uma abordagem ao desenvolvimento de um sistema de software complexo conectando seus componentes aos domínios em evolução, ou principais metas de negócios, atendidos por cada componente. Esse conceito foi introduzido por Eric Evans em seu livro Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). Para obter informações sobre como você pode usar o design orientado por domínio com o padrão strangler fig, consulte Modernizando os serviços web legados da Microsoft ASP.NET (ASMX) de forma incremental usando contêineres e o Amazon API Gateway.

DR

Veja recuperação de desastres.

Detecção da oscilação

Rastreamento de desvios de uma configuração de linha de base. Por exemplo, você pode usar AWS CloudFormation para detectar desvios nos recursos do sistema ou AWS Control Tower para detectar mudanças em seu landing zone que possam afetar a conformidade com os requisitos de governança.

DVSM

Veja mapeamento do fluxo de valor de desenvolvimento.

E

EDA

Veja análise exploratória de dados.

EDI

Veja intercâmbio eletrônico de dados.

computação de borda

A tecnologia que aumenta o poder computacional de dispositivos inteligentes nas bordas de uma rede de IoT. Quando comparada com a computação em nuvem, a computação de borda pode reduzir a latência da comunicação e melhorar o tempo de resposta.

intercâmbio eletrônico de dados (EDI)

A troca automatizada de documentos comerciais entre organizações. Para obter mais informações, consulte O que é EDI (Intercâmbio eletrônico de dados)?.

criptografia

Um processo de computação que transforma dados de texto simples, legíveis por humanos, em texto cifrado.

chave de criptografia

Uma sequência criptográfica de bits aleatórios que é gerada por um algoritmo de criptografia. As chaves podem variar em tamanho, e cada chave foi projetada para ser imprevisível e exclusiva.

endianismo

A ordem na qual os bytes são armazenados na memória do computador. Big-endian os sistemas armazenam primeiro o byte mais significativo. Little-endian os sistemas armazenam primeiro o byte menos significativo.

endpoint

Veja endpoint de serviço.

serviço de endpoint

Um serviço que pode ser hospedado em uma nuvem privada virtual (VPC) para ser compartilhado com outros usuários. Você pode criar um serviço de endpoint com AWS PrivateLink e conceder permissões a outros diretores Contas da AWS ou a AWS Identity and Access Management (IAM). Essas contas ou entidades principais podem se conectar ao serviço de endpoint de maneira privada criando endpoints da VPC de interface. Para obter mais informações, consulte Criar um serviço de endpoint na documentação do Amazon Virtual Private Cloud (Amazon VPC).

planejamento de recursos empresariais (ERP)

Um sistema que automatiza e gerencia os principais processos de negócios (como contabilidade, MES e gerenciamento de projetos) para uma empresa.

criptografia envelopada

O processo de criptografar uma chave de criptografia com outra chave de criptografia. Para obter mais informações, consulte Criptografia de envelope na documentação AWS Key Management Service (AWS KMS).

ambiente

Uma instância de uma aplicação em execução. Estes são tipos comuns de ambientes na computação em nuvem:

- ambiente de desenvolvimento: uma instância de uma aplicação em execução que está disponível somente para a equipe principal responsável pela manutenção da aplicação. Ambientes de desenvolvimento são usados para testar mudanças antes de promovê-las para ambientes superiores. Esse tipo de ambiente às vezes é chamado de ambiente de teste.
- ambientes inferiores: todos os ambientes de desenvolvimento para uma aplicação, como aqueles usados para compilações e testes iniciais.
- ambiente de produção: uma instância de uma aplicação em execução que os usuários finais podem acessar. Em um CI/CD pipeline, o ambiente de produção é o último ambiente de implantação.

- ambientes superiores: todos os ambientes que podem ser acessados por usuários que não sejam a equipe principal de desenvolvimento. Isso pode incluir um ambiente de produção, ambientes de pré-produção e ambientes para testes de aceitação do usuário.

epic

Em metodologias ágeis, categorias funcionais que ajudam a organizar e priorizar seu trabalho. Os epics fornecem uma descrição de alto nível dos requisitos e das tarefas de implementação. Por exemplo, os épicos de segurança AWS da CAF incluem gerenciamento de identidade e acesso, controles de detetive, segurança de infraestrutura, proteção de dados e resposta a incidentes. Para obter mais informações sobre epics na estratégia de migração da AWS , consulte o guia de implementação do programa.

ERP

Veja planejamento de recursos empresariais.

análise exploratória de dados (EDA)

O processo de analisar um conjunto de dados para entender suas principais características. Você coleta ou agrega dados e, em seguida, realiza investigações iniciais para encontrar padrões, detectar anomalias e verificar suposições. O EDA é realizado por meio do cálculo de estatísticas resumidas e da criação de visualizações de dados.

F

tabela de fatos

A tabela central em um esquema em estrela. Ela armazena dados quantitativos sobre as operações comerciais. Normalmente, uma tabela de fatos contém dois tipos de colunas: as que contêm medidas e as que contêm uma chave externa para uma tabela de dimensões.

Antecipar-se à falha

Uma filosofia que usa testes frequentes e incrementais para reduzir o ciclo de vida do desenvolvimento. É uma parte essencial de uma abordagem ágil.

delimitação de isolamento contra falhas

No Nuvem AWS, um limite, como uma zona de disponibilidade, Região da AWS um plano de controle ou um plano de dados, que limita o efeito de uma falha e ajuda a melhorar a resiliência das cargas de trabalho. Para obter mais informações, consulte AWS Fault Isolation Boundaries.

ramificação de recursos

Veja ramificação.

recursos

Os dados de entrada usados para fazer uma previsão. Por exemplo, em um contexto de manufatura, os recursos podem ser imagens capturadas periodicamente na linha de fabricação.

importância do recurso

O quanto um recurso é importante para as previsões de um modelo. Isso geralmente é expresso como uma pontuação numérica que pode ser calculada por meio de várias técnicas, como Shapley Additive Explanations (SHAP) e gradientes integrados. Para obter mais informações, consulte Interpretabilidade do modelo de aprendizado de máquina com AWS.

transformação de recursos

O processo de otimizar dados para o processo de ML, incluindo enriquecer dados com fontes adicionais, escalar valores ou extrair vários conjuntos de informações de um único campo de dados. Isso permite que o modelo de ML se beneficie dos dados. Por exemplo, se a data “2021-05-27 00:15:37” for dividida em “2021”, “maio”, “quinta” e “15”, isso poderá ajudar o algoritmo de aprendizado a aprender padrões diferenciados associados a diferentes componentes de dados.

prompt few shot

Fornecer a um LLM um pequeno número de exemplos que demonstram a tarefa e o resultado desejado antes de solicitar que ele execute uma tarefa semelhante. Essa técnica é uma aplicação do aprendizado contextual, em que os modelos aprendem com exemplos (fotos) incorporados aos prompts. Few-shot a solicitação pode ser eficaz para tarefas que exigem formatação, raciocínio ou conhecimento de domínio específicos. Veja também prompts zero-shot.

FGAC

Veja controle de acesso refinado.

Controle de acesso refinado (FGAC)

O uso de várias condições para permitir ou negar uma solicitação de acesso.

migração flash-cut

Um método de migração de banco de dados que usa replicação contínua de dados via captura de dados de alteração para migrar os dados no menor tempo possível, em vez de usar uma abordagem em fases. O objetivo é reduzir ao mínimo o tempo de inatividade.

FM

Veja modelo de base.

modelo de base (FM)

Uma grande rede neural de aprendizado profundo que treina em grandes conjuntos de dados generalizados e não rotulados. Os FMs são capazes de realizar uma ampla variedade de tarefas gerais, como entender a linguagem, gerar texto e imagens e conversar em linguagem natural. Para obter mais informações, consulte O que são modelos de base?.

Gateway FM

Um intermediário centralizado que controla e normaliza o acesso aos modelos de fundação. Também conhecido como gateway LLM.

G

IA generativa

Um subconjunto de modelos de IA que foram treinados em grandes quantidades de dados e que podem usar um simples prompt de texto para criar novos artefatos e conteúdo, como imagens, vídeos, texto e áudio. Para obter mais informações, consulte O que é IA generativa?.

bloqueio geográfico

Veja restrições geográficas.

restrições geográficas (bloqueio geográfico)

Na Amazon CloudFront, uma opção para impedir que usuários em países específicos acessem distribuições de conteúdo. É possível usar uma lista de permissões ou uma lista de bloqueios para especificar países aprovados e banidos. Para obter mais informações, consulte Restringir a distribuição geográfica do seu conteúdo na CloudFront documentação.

Fluxo de trabalho do GitFlow

Uma abordagem na qual ambientes inferiores e superiores usam ramificações diferentes em um repositório de código-fonte. O fluxo de trabalho do Gitflow é considerado legado, e o fluxo de trabalho trunk-based é a abordagem moderna e preferencial.

golden image

Um snapshot de um sistema ou software usado como modelo para implantar novas instâncias desse sistema ou software. Por exemplo, na manufatura, uma golden image pode ser usada para

provisionar software em vários dispositivos e ajudar a melhorar a velocidade, a escalabilidade e a produtividade nas operações de fabricação de dispositivos.

estratégia greenfield

A ausência de infraestrutura existente em um novo ambiente. Ao adotar uma estratégia greenfield para uma arquitetura de sistema, é possível selecionar todas as novas tecnologias sem a restrição da compatibilidade com a infraestrutura existente, também conhecida como brownfield. Se estiver expandindo a infraestrutura existente, poderá combinar as estratégias brownfield e greenfield.

barreira de proteção

Uma regra de alto nível que ajuda a gerenciar recursos, políticas e conformidade em todas as unidades organizacionais (UOs). Barreiras de proteção preventivas impõem políticas para garantir o alinhamento a padrões de conformidade. Elas são implementadas usando políticas de controle de serviço e limites de permissões do IAM. Barreiras de proteção detectivas detectam violações de políticas e problemas de conformidade e geram alertas para remediação. Eles são implementados usando AWS Config, AWS Security Hub CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector e verificações personalizadas AWS Lambda .

grades de proteção (IA)

Mecanismos de segurança que filtram, validam e restringem as entradas e saídas dos agentes para ajudar a garantir um comportamento de IA responsável e seguro.

H

HA

Veja alta disponibilidade.

migração heterogênea de bancos de dados

Migrar seu banco de dados de origem para um banco de dados de destino que usa um mecanismo de banco de dados diferente (por exemplo, Oracle para Amazon Aurora). A migração heterogênea geralmente faz parte de um esforço de redefinição da arquitetura, e converter o esquema pode ser uma tarefa complexa. O AWS fornece o AWS SCT para ajudar nas conversões de esquemas.

alta disponibilidade (HA)

A capacidade de uma workload operar continuamente, sem intervenção, em caso de desafios ou desastres. Os sistemas AH são projetados para realizar o failover automático, oferecer consistentemente desempenho de alta qualidade e lidar com diferentes cargas e falhas com impacto mínimo no desempenho.

modernização de historiador

Uma abordagem usada para modernizar e atualizar os sistemas de tecnologia operacional (OT) para melhor atender às necessidades do setor de manufatura. Um historiador é um tipo de banco de dados usado para coletar e armazenar dados de várias fontes em uma fábrica.

dados de hold-out

Uma parte dos dados históricos rotulados que são retidos de um conjunto de dados usado para treinar um modelo de machine learning. Você pode usar dados de hold-out para avaliar a performance do modelo comparando as predições do modelo com os dados de retenção.

humano no circuito (HiTL)

Um padrão de fluxo de trabalho em que a execução do agente é pausada para análise e aprovação humana em pontos críticos de decisão.

migração homogênea de bancos de dados

Migrar seu banco de dados de origem para um banco de dados de destino que compartilha o mesmo mecanismo de banco de dados (por exemplo, Microsoft SQL Server para Amazon RDS para SQL Server). A migração homogênea geralmente faz parte de um esforço de redefinição da hospedagem ou da plataforma. É possível usar utilitários de banco de dados nativos para migrar o esquema.

dados quentes

Dados acessados com frequência, como dados em tempo real ou dados translacionais recentes. Esses dados normalmente exigem uma camada ou classe de armazenamento de alto desempenho para fornecer respostas rápidas às consultas.

hotfix

Uma correção urgente para um problema crítico em um ambiente de produção. Devido à sua urgência, um hotfix geralmente é feito fora do fluxo de trabalho típico de uma DevOps versão.

período de hipercuidados

Imediatamente após a substituição, o período em que uma equipe de migração gerencia e monitora as aplicações migradas na nuvem para resolver quaisquer problemas. Normalmente, a duração desse período é de 1 a 4 dias. No final do período de hipercuidados, a equipe de migração normalmente transfere a responsabilidade pelas aplicações para a equipe de operações de nuvem.

eu

IaC

Veja infraestrutura como código.

Política baseada em identidade

Uma política anexada a um ou mais diretores do IAM que define suas permissões no Nuvem AWS ambiente.

aplicação ociosa

Uma aplicação que tem um uso médio de CPU e memória entre 5 e 20% em um período de 90 dias. Em um projeto de migração, é comum retirar essas aplicações ou retê-las on-premises.

IIoT

Veja Internet das Coisas Industrial.

infraestrutura imutável

Um modelo que implanta uma nova infraestrutura para workloads de produção em vez de atualizar, aplicar patches ou modificar a infraestrutura existente. Infraestruturas imutáveis são inerentemente mais consistentes, confiáveis e preditivas do que infraestruturas mutáveis. Para obter mais informações, consulte as melhores práticas de implantação usando infraestrutura imutável na AWS Well-Architected Estrutura.

VPC de entrada (admissão)

Em uma arquitetura de AWS várias contas, uma VPC que aceita, inspeciona e roteia conexões de rede de fora de um aplicativo. A Arquitetura de referência de segurança da AWS recomenda configurar sua conta de rede com VPCs de entrada, saída e inspeção para proteger a interface bidirecional entre a aplicação e a Internet em geral.

migração incremental

Uma estratégia de substituição na qual você migra a aplicação em pequenas partes, em vez de realizar uma única substituição completa. Por exemplo, é possível mover inicialmente apenas alguns microsserviços ou usuários para o novo sistema. Depois de verificar se tudo está funcionando corretamente, mova os microsserviços ou usuários adicionais de forma incremental até poder descomissionar seu sistema herdado. Essa estratégia reduz os riscos associados a migrações de grande porte.

Indústria 4.0

Um termo que foi introduzido por Klaus Schwab em 2016 para se referir à modernização dos processos de fabricação por meio de avanços na conectividade, dados em tempo real, automação, análise e. AI/ML

infraestrutura

Todos os recursos e ativos contidos no ambiente de uma aplicação.

Infraestrutura como código (IaC)

O processo de provisionamento e gerenciamento da infraestrutura de uma aplicação por meio de um conjunto de arquivos de configuração. A IaC foi projetada para ajudar você a centralizar o gerenciamento da infraestrutura, padronizar recursos e escalar rapidamente para que novos ambientes sejam reproduzíveis, confiáveis e consistentes.

Internet das Coisas Industrial (IIoT)

O uso de sensores e dispositivos conectados à Internet nos setores industriais, como manufatura, energia, automotivo, saúde, ciências biológicas e agricultura. Para obter mais informações, consulte Construir uma estratégia de transformação digital para a Internet das Coisas Industrial (IIoT).

VPC de inspeção

Em uma arquitetura de AWS várias contas, uma VPC centralizada que gerencia as inspeções do tráfego de rede entre VPCs (na mesma ou em diferentes Regiões da AWS), a Internet e as redes locais. A Arquitetura de referência de segurança da AWS recomenda configurar sua conta de rede com VPCs de entrada, saída e inspeção para proteger a interface bidirecional entre a aplicação e a Internet em geral.

Internet das coisas (IoT)

A rede de objetos físicos conectados com sensores ou processadores incorporados que se comunicam com outros dispositivos e sistemas pela Internet ou por uma rede de comunicação local. Para obter mais informações, consulte O que é IoT?

interpretabilidade

Uma característica de um modelo de machine learning que descreve o grau em que um ser humano pode entender como as previsões do modelo dependem de suas entradas. Para obter mais informações, consulte Interpretabilidade do modelo de aprendizado de máquina com AWS.

IoT

Veja Internet das Coisas.

Biblioteca de informações de TI (ITIL)

Um conjunto de práticas recomendadas para fornecer serviços de TI e alinhar esses serviços a requisitos de negócios. A ITIL fornece a base para o ITSM.

Gerenciamento de serviços de TI (ITSM)

Atividades associadas a design, implementação, gerenciamento e suporte de serviços de TI para uma organização. Para obter informações sobre a integração de operações em nuvem com ferramentas de ITSM, consulte o guia de integração de operações.

ITIL

Veja biblioteca de informações de TI.

ITSM

Veja gerenciamento de serviços de TI.

L

controle de acesso baseado em etiqueta (LBAC)

Uma implementação do controle de acesso obrigatório (MAC) em que os usuários e os dados em si recebem explicitamente um valor de etiqueta de segurança. A interseção entre a etiqueta de segurança do usuário e a etiqueta de segurança dos dados determina quais linhas e colunas podem ser vistas pelo usuário.

zona de pouso

Uma landing zone é um AWS ambiente bem arquitetado, com várias contas, escalável e seguro. Um ponto a partir do qual suas organizações podem iniciar e implantar rapidamente workloads e aplicações com confiança em seu ambiente de segurança e infraestrutura. Para obter mais informações sobre zonas de pouso, consulte Configurar um ambiente da AWS com várias contas seguro e escalável.

grande modelo de linguagem (LLM)

Um modelo de IA de aprendizado profundo pré-treinado em uma grande quantidade de dados. Um LLM pode realizar várias tarefas, como responder a perguntas, resumir documentos, traduzir texto para outros idiomas e completar frases. Para obter mais informações, consulte O que é grande modelo de linguagem (LLM)?.

migração de grande porte

Uma migração de 300 servidores ou mais.

LBAC

Veja controle de acesso baseado em rótulo.

privilégio mínimo

A prática recomendada de segurança de conceder as permissões mínimas necessárias para executar uma tarefa. Para obter mais informações, consulte Aplicar permissões de privilégios mínimos na documentação do IAM.

mover sem alterações (lift-and-shift)

Veja 7 Rs.

sistema little-endian

Um sistema que armazena o byte menos significativo antes. Veja também endianness.

LLM

Veja grande modelo de linguagem.

ambientes inferiores

Veja ambiente.

M

machine learning (ML)

Um tipo de inteligência artificial que usa algoritmos e técnicas para reconhecimento e aprendizado de padrões. O ML analisa e aprende com dados gravados, por exemplo, dados da Internet das Coisas (IoT), para gerar um modelo estatístico baseado em padrões. Para obter mais informações, consulte Machine learning.

ramificação principal

Veja ramificação.

Malware

Software projetado para comprometer a segurança ou a privacidade do computador. O malware pode interromper os sistemas do computador, vazar informações sensíveis ou obter acesso não autorizado. Exemplos de malware incluem vírus, worms, ransomware, cavalos de Troia, spyware e keyloggers.

Serviços gerenciados

Serviços da AWS para o qual AWS opera a camada de infraestrutura, o sistema operacional e as plataformas, e você acessa os endpoints para armazenar e recuperar dados. O Amazon Simple Storage Service (Amazon S3) e o Amazon DynamoDB são exemplos de serviços gerenciados. Eles também são conhecidos como serviços abstraídos.

sistema de execução de manufatura (MES)

Um sistema de software para rastrear, monitorar, documentar e controlar processos de produção que convertem matérias-primas em produtos acabados no chão de fábrica.

MAP

Veja Programa de Aceleração da Migração.

MCP

Consulte Protocolo de contexto do modelo.

Protocolo de contexto para modelos (MCP)

Um protocolo sem estado para comunicação entre agentes e ferramentas.

Servidor MCP

Um serviço que expõe uma ou mais ferramentas por meio do Model Context Protocol.

mecanismo

Um processo completo em que você cria uma ferramenta, impulsiona a adoção da ferramenta e, em seguida, inspeciona os resultados para fazer ajustes. Um mecanismo é um ciclo que se reforça e se aprimora à medida que opera. Para obter mais informações, consulte Criação de mecanismos na AWS Well-Architected estrutura.

conta de membro

Todos, Contas da AWS exceto a conta de gerenciamento, que fazem parte de uma organização em AWS Organizations. Uma conta só pode ser membro de uma organização de cada vez.

MES

Veja sistema de execução de manufatura.

Transporte de Telemetria de Enfileiramento de Mensagens (MQTT)

Um protocolo de comunicação leve, máquina a máquina (M2M), baseado no padrão, para dispositivos de IoT com recursos publish/subscribelimitados.

microsserviço

Um serviço pequeno e independente que se comunica por meio de APIs bem definidas e normalmente pertence a equipes pequenas e autônomas. Por exemplo, um sistema de seguradora pode incluir microsserviços que mapeiam as capacidades comerciais, como vendas ou marketing, ou subdomínios, como compras, reclamações ou análises. Os benefícios dos microsserviços incluem agilidade, escalabilidade flexível, fácil implantação, código reutilizável e resiliência. Para obter mais informações, consulte Integração de microsserviços usando serviços sem AWS servidor.

arquitetura de microsserviços

Uma abordagem à criação de aplicações com componentes independentes que executam cada processo de aplicação como um microsserviço. Esses microsserviços se comunicam por meio de uma interface bem definida usando APIs leves. Cada microsserviço nessa arquitetura pode ser atualizado, implantado e escalado para atender à demanda por funções específicas de uma aplicação. Para obter mais informações, consulte Implementação de microsserviços em. AWS

Programa de Aceleração da Migração (MAP)

Um AWS programa que fornece suporte de consultoria, treinamento e serviços para ajudar as organizações a criar uma base operacional sólida para migrar para a nuvem e ajudar a

compensar o custo inicial das migrações. O MAP inclui uma metodologia de migração para executar migrações legadas de forma metódica e um conjunto de ferramentas para automatizar e acelerar cenários comuns de migração.

migração em escala

O processo de mover a maior parte do portfólio de aplicações para a nuvem em ondas, com mais aplicações sendo movidas em um ritmo mais rápido a cada onda. Essa fase usa as práticas recomendadas e lições aprendidas nas fases anteriores para implementar uma fábrica de migração de equipes, ferramentas e processos para agilizar a migração de workloads por meio de automação e entrega ágeis. Esta é a terceira fase da estratégia de migração para a AWS.

fábrica de migração

Cross-functional equipes que simplificam a migração de cargas de trabalho por meio de abordagens automatizadas e ágeis. As equipes da fábrica de migração geralmente incluem operações, analistas e proprietários de negócios, engenheiros de migração, desenvolvedores e DevOps profissionais que trabalham em sprints. Entre 20 e 50% de um portfólio de aplicações corporativas consiste em padrões repetidos que podem ser otimizados por meio de uma abordagem de fábrica. Para obter mais informações, consulte discussão sobre fábricas de migração e o guia do Cloud Migration Factory neste conjunto de conteúdo.

metadados de migração

As informações sobre a aplicação e o servidor necessárias para concluir a migração. Cada padrão de migração exige um conjunto de metadados de migração diferente. Exemplos de metadados de migração incluem a sub-rede, o grupo de segurança e AWS a conta de destino.

padrão de migração

Uma tarefa de migração repetível que detalha a estratégia de migração, o destino da migração e a aplicação ou o serviço de migração usado. Exemplo: rehospede a migração para o Amazon EC2 AWS com o Application Migration Service.

Avaliação de Portfólio para Migração (MPA)

Uma ferramenta on-line que fornece informações para validar o caso de negócios para migrar para a Nuvem AWS. O MPA fornece avaliação detalhada do portfólio (dimensionamento correto do servidor, preços, comparações de TCO, análise de custos de migração), bem como planejamento de migração (análise e coleta de dados de aplicações, agrupamento de aplicações, priorização de migração e planejamento de ondas). A ferramenta MPA (requer login) está disponível gratuitamente para todos os AWS consultores e consultores parceiros da APN.

Avaliação de Preparação para Migração (MRA)

O processo de obter insights sobre o status de prontidão de uma organização para a nuvem, identificar pontos fortes e fracos e criar um plano de ação para fechar as lacunas identificadas, usando o CAF. AWS Para mais informações, consulte o guia de preparação para migração. A MRA é a primeira fase da estratégia de migração para a AWS.

estratégia de migração

A abordagem usada para migrar uma workload para a Nuvem AWS. Para obter mais informações, veja a entrada 7 Rs neste glossário e consulte Mobilize sua organização para acelerar migrações em grande escala.

ML

Veja machine learning.

modernização

Transformar uma aplicação desatualizada (herdada ou monolítica) e sua infraestrutura em um sistema ágil, elástico e altamente disponível na nuvem para reduzir custos, ganhar eficiência e aproveitar as inovações. Para obter mais informações, consulte Strategy for modernizing applications in the Nuvem AWS.

avaliação de preparação para modernização

Uma avaliação que ajuda a determinar a preparação para modernização das aplicações de uma organização. Ela identifica benefícios, riscos e dependências e determina o quão bem a organização pode acomodar o estado futuro dessas aplicações. O resultado da avaliação é um esquema da arquitetura de destino, um roteiro que detalha as fases de desenvolvimento e os marcos do processo de modernização e um plano de ação para abordar as lacunas identificadas. Para obter mais informações, consulte Evaluating modernization readiness for applications in the Nuvem AWS.

aplicações monolíticas (monólitos)

Aplicações que são executadas como um único serviço com processos fortemente acoplados. As aplicações monolíticas apresentam várias desvantagens. Se um recurso da aplicação apresentar um aumento na demanda, toda a arquitetura deverá ser escalada. Adicionar ou melhorar os recursos de uma aplicação monolítica também se torna mais complexo quando a base de código cresce. Para resolver esses problemas, é possível criar uma arquitetura de microsserviços. Para obter mais informações, consulte Decompor monólitos em microsserviços.

MPA

Veja Avaliação do Portfólio para Migração.

MQTT

Veja Transporte de Telemetria de Enfileiramento de Mensagens.

classificação multiclasse

Um processo que ajuda a gerar previsões para várias classes (prevendo um ou mais de dois resultados). Por exemplo, um modelo de ML pode perguntar “Este produto é um livro, um carro ou um telefone?” ou “Qual categoria de produtos é mais interessante para este cliente?”

infraestrutura mutável

Um modelo que atualiza e modifica a infraestrutura existente para workloads de produção. Para melhorar a consistência, confiabilidade e previsibilidade, a AWS Well-Architected Estrutura recomenda o uso de infraestrutura imutável como uma prática recomendada.

O

OAC

Veja controle de acesso de origem.

OAI

Veja identidade de acesso de origem.

OCM

Veja gerenciamento de alterações organizacionais.

migração offline

Um método de migração no qual a workload de origem é desativada durante o processo de migração. Esse método envolve tempo de inatividade prolongado e geralmente é usado para workloads pequenas e não críticas.

OI

Veja integração de operações.

Ola

Veja acordo de nível operacional.

migração online

Um método de migração no qual a workload de origem é copiada para o sistema de destino sem ser colocada offline. As aplicações conectadas à workload podem continuar funcionando durante a migração. Esse método envolve um tempo de inatividade nulo ou mínimo e normalmente é usado para workloads essenciais para a produção.

OPC-UA

Veja Open Process Communications - Unified Architecture.

Comunicação de processo aberto - Arquitetura unificada (OPC-UA)

Um protocolo de comunicação máquina a máquina (M2M) para automação industrial. OPC-UA fornece um padrão de interoperabilidade com esquemas de criptografia, autenticação e autorização de dados.

acordo de nível operacional (OLA)

Um acordo que esclarece o que os grupos funcionais de TI prometem oferecer uns aos outros para apoiar um acordo de serviço (SLA).

análise de prontidão operacional (ORR)

Uma lista de verificação de perguntas e práticas recomendadas associadas que ajudam você a entender, avaliar, prevenir ou reduzir o escopo de incidentes e possíveis falhas. Para obter mais informações, consulte Operational Readiness Reviews (ORR) na AWS Well-Architected Estrutura.

tecnologia operacional (TO)

Sistemas de hardware e software que trabalham com o ambiente físico para controlar operações, equipamentos e infraestrutura industriais. Na manufatura, a integração dos sistemas de tecnologia da informação (TI) e tecnologia operacional (TO) é o foco principal das transformações da Indústria 4.0.

integração de operações (OI)

O processo de modernização das operações na nuvem, que envolve planejamento de preparação, automação e integração. Para obter mais informações, consulte o guia de integração de operações.

trilha organizacional

Uma trilha criada por ela AWS CloudTrail registra todos os eventos de todos Contas da AWS em uma organização em AWS Organizations. Essa trilha é criada em cada Conta da AWS que faz parte da organização e monitora a atividade em cada conta. Para obter mais informações, consulte Criação de uma trilha para uma organização na CloudTrail documentação.

gerenciamento de alterações organizacionais (OCM)

Uma estrutura para gerenciar grandes transformações de negócios disruptivas de uma perspectiva de pessoas, cultura e liderança. O OCM ajuda as organizações a se prepararem e fazerem a transição para novos sistemas e estratégias, acelerando a adoção de alterações, abordando questões de transição e promovendo mudanças culturais e organizacionais. Na estratégia de AWS migração, essa estrutura é chamada de aceleração de pessoas, devido à velocidade de mudança necessária nos projetos de adoção da nuvem. Para obter mais informações, consulte o guia do OCM.

controle de acesso de origem (OAC)

Em CloudFront, uma opção aprimorada para restringir o acesso para proteger seu conteúdo do Amazon Simple Storage Service (Amazon S3). O OAC oferece suporte a todos os buckets do S3 Regiões da AWS, à criptografia do lado do servidor com AWS KMS (SSE-KMS) e à dinâmica `PUT` e `DELETE` às solicitações ao bucket do S3.

Identidade do acesso de origem (OAI)

Em CloudFront, uma opção para restringir o acesso para proteger seu conteúdo do Amazon S3. Quando você usa o OAI, CloudFront cria um principal com o qual o Amazon S3 pode se autenticar. Os diretores autenticados podem acessar o conteúdo em um bucket do S3 somente por meio de uma distribuição específica. CloudFront Veja também OAC, que fornece um controle de acesso mais granular e aprimorado.

ORR

Veja análise de prontidão operacional.

OT

Veja tecnologia operacional.

VPC de saída (egresso)

Em uma arquitetura de AWS várias contas, uma VPC que gerencia conexões de rede que são iniciadas de dentro de um aplicativo. A Arquitetura de referência de segurança da AWS

recomenda configurar sua conta de rede com VPCs de entrada, saída e inspeção para proteger a interface bidirecional entre a aplicação e a Internet em geral.

P

limite de permissões

Uma política de gerenciamento do IAM anexada a entidades principais do IAM para definir as permissões máximas que o usuário ou perfil podem ter. Para obter mais informações, consulte Limites de permissões na documentação do IAM.

Informações de identificação pessoal (PII)

Informações que, quando visualizadas diretamente ou combinadas com outros dados relacionados, podem ser usadas para inferir razoavelmente a identidade de um indivíduo. Exemplos de PII incluem nomes, endereços e informações de contato.

PII

Veja informações de identificação pessoal.

manual

Um conjunto de etapas predefinidas que capturam o trabalho associado às migrações, como a entrega das principais funções operacionais na nuvem. Um manual pode assumir a forma de scripts, runbooks automatizados ou um resumo dos processos ou etapas necessários para operar seu ambiente modernizado.

PLC

Veja controlador lógico programável.

PLM

Veja gerenciamento do ciclo de vida do produto.

política

Um objeto que pode definir permissões (veja política baseada em identidade), especificar condições de acesso (veja políta baseada em recurso) ou definir as permissões máximas para todas as contas em uma organização no AWS Organizations (veja política de controle de serviços).

persistência poliglota

Escolher de forma independente a tecnologia de armazenamento de dados de um microsserviço com base em padrões de acesso a dados e outros requisitos. Se seus microsserviços tiverem a mesma tecnologia de armazenamento de dados, eles poderão enfrentar desafios de implementação ou apresentar baixa performance. Os microsserviços serão implementados com mais facilidade e alcançarão performance e escalabilidade melhores se usarem o armazenamento de dados mais bem adaptado às suas necessidades.

avaliação do portfólio

Um processo de descobrir, analisar e priorizar o portfólio de aplicações para planejar a migração. Para obter mais informações, consulte Avaliar a preparação para a migração.

predicado

Uma condição de consulta que retorna `true` ou `false`, normalmente localizada em uma cláusula `WHERE`.

pushdown de predicados

Uma técnica de otimização de consultas de banco de dados que filtra os dados na consulta antes da transferência. Isso reduz a quantidade de dados que devem ser recuperados e processados do banco de dados relacional e melhora a performance das consultas.

controle preventivo

Um controle de segurança projetado para evitar que um evento ocorra. Esses controles são a primeira linha de defesa para ajudar a evitar acesso não autorizado ou alterações indesejadas em sua rede. Para obter mais informações, consulte Controles preventivos em Como implementar controles de segurança na AWS.

principal (entidade principal)

Uma entidade AWS que pode realizar ações e acessar recursos. Essa entidade geralmente é um usuário raiz para um Conta da AWS, uma função do IAM ou um usuário. Para obter mais informações, consulte Entidade principal em Termos e conceitos de perfis na documentação do IAM.

Privacidade por design

Uma abordagem em engenharia de sistemas que leva em consideração a privacidade em todo o processo de desenvolvimento.

zonas hospedadas privadas

Um contêiner que armazena informações sobre como você quer que o Amazon Route 53 responda a consultas ao DNS para um domínio e seus subdomínios dentro de uma ou mais VPCs. Para obter mais informações, consulte Como trabalhar com zonas hospedadas privadas na documentação do Route 53.

controle proativo

Um controle de segurança desenvolvido para evitar a implantação de recursos não conformes. Esses controles verificam os recursos antes de serem provisionados. Se o recurso não estiver em conformidade com o controle, ele não será provisionado. Para obter mais informações, consulte o guia de referência de controles na AWS Control Tower documentação e consulte Controles proativos em Implementação de controles de segurança em AWS.

gerenciamento do ciclo de vida do produto (PLM)

O gerenciamento de dados e processos de um produto em todo o seu ciclo de vida, desde a concepção, o desenvolvimento e o lançamento, passando pelo crescimento e maturidade, até o declínio e a remoção.

ambiente de produção

Veja ambiente.

controlador lógico programável (PLC)

Na manufatura, um computador altamente confiável e adaptável que monitora as máquinas e automatiza os processos de fabricação.

encadeamento de prompts

Uso da saída de um prompt do LLM como entrada para o próximo prompt para gerar respostas melhores. Essa técnica é usada para dividir uma tarefa complexa em subtarefas, ou para refinar ou expandir iterativamente uma resposta preliminar. Isso ajuda a melhorar a precisão e a relevância das respostas de um modelo e permite resultados mais granulares e personalizados.

pseudonimização

O processo de substituir identificadores pessoais em um conjunto de dados por valores de espaço reservado. A pseudonimização pode ajudar a proteger a privacidade pessoal. Os dados pseudonimizados ainda são considerados dados pessoais.

publish/subscribe (pub/sub)

Um padrão que permite comunicações assíncronas entre microsserviços para melhorar a escalabilidade e a capacidade de resposta. Por exemplo, em um MES baseado em microsserviços, um microsserviço pode publicar mensagens de eventos em um canal em que outros microsserviços possam assinar. O sistema pode adicionar novos microsserviços sem alterar o serviço de publicação.

Q

plano de consulta

Uma série de etapas, como instruções, usadas para acessar os dados em um sistema de banco de dados relacional SQL.

regressão de planos de consultas

Quando um otimizador de serviço de banco de dados escolhe um plano menos adequado do que escolhia antes de uma determinada alteração no ambiente de banco de dados ocorrer. Isso pode ser causado por alterações em estatísticas, restrições, configurações do ambiente, associações de parâmetros de consulta e atualizações do mecanismo de banco de dados.

R

Matriz RACI

Veja responsável, aprovador, consultado, informado (RACI).

RAG

Veja geração aumentada via recuperação.

ransomware

Um software mal-intencionado desenvolvido para bloquear o acesso a um sistema ou dados de computador até que um pagamento seja feito.

Matriz RASCI

Veja responsável, aprovador, consultado, informado (RACI).

RCAC

Veja controle de acesso por linha e coluna.

réplica de leitura

Uma cópia de um banco de dados usada somente para leitura. É possível encaminhar consultas para a réplica de leitura e reduzir a carga no banco de dados principal.

Redefinir arquitetura

Veja 7 Rs.

objetivo de ponto de recuperação (RPO).

O máximo período de tempo aceitável desde o último ponto de recuperação de dados. Isso determina o que é considerado uma perda aceitável de dados entre o último ponto de recuperação e a interrupção do serviço.

objetivo de tempo de recuperação (RTO)

O máximo atraso aceitável entre a interrupção e a restauração do serviço.

refatorar

Veja 7 Rs.

Região

Uma coleção de AWS recursos em uma área geográfica. Cada um Região da AWS é isolado e independente dos outros para fornecer tolerância a falhas, estabilidade e resiliência. Para obter informações, consulte Specify which Regiões da AWS your account can use.

regressão

Uma técnica de ML que prevê um valor numérico. Por exemplo, para resolver o problema de “Por qual preço esta casa será vendida?” um modelo de ML pode usar um modelo de regressão linear para prever o preço de venda de uma casa com base em fatos conhecidos sobre a casa (por exemplo, a metragem quadrada).

redefinir a hospedagem

Veja 7 Rs.

versão

Em um processo de implantação, o ato de promover mudanças em um ambiente de produção.

realocar

Veja 7 Rs.

redefinir a plataforma

Veja 7 Rs.

recomprar

Veja 7 Rs.

resiliência

A capacidade de uma aplicação de resistir ou se recuperar de interrupções. Alta disponibilidade e recuperação de desastres são considerações comuns ao planejar a resiliência na Nuvem AWS. Para obter mais informações, consulte Nuvem AWS Resilience.

política baseada em recurso

Uma política associada a um recurso, como um bucket do Amazon S3, um endpoint ou uma chave de criptografia. Esse tipo de política especifica quais entidades principais têm acesso permitido, ações válidas e quaisquer outras condições que devem ser atendidas.

matriz responsável, accountable, consultada, informada (RACI)

Uma matriz que define as funções e responsabilidades de todas as partes envolvidas nas atividades de migração e nas operações de nuvem. O nome da matriz é derivado dos tipos de responsabilidade definidos na matriz: responsável (R), responsabilizável (A), consultado (C) e informado (I). O tipo de suporte (S) é opcional. Se você incluir suporte, a matriz será chamada de matriz RASCI e, se excluir, será chamada de matriz RACI.

controle responsivo

Um controle de segurança desenvolvido para conduzir a remediação de eventos adversos ou desvios em relação à linha de base de segurança. Para obter mais informações, consulte Controles responsivos em Como implementar controles de segurança na AWS.

reter

Veja 7 Rs.

Retirada

Veja 7 Rs.

Geração Aumentada de Recuperação (RAG)

Uma tecnologia de IA generativa em que um LLM faz referência a uma fonte de dados autorizada que está fora de suas fontes de dados de treinamento antes de gerar uma resposta. Por exemplo, um modelo RAG pode realizar uma pesquisa semântica na base de conhecimento ou nos dados personalizados de uma organização. Para obter mais informações, consulte O que é RAG (geração aumentada via recuperação)?.

alternância

O processo de atualizar periodicamente um segredo para dificultar o acesso de um invasor às credenciais.

controle de acesso por linha e coluna (RCAC)

O uso de expressões SQL básicas e flexíveis que tenham regras de acesso definidas. O RCAC consiste em permissões de linha e máscaras de coluna.

RPO

Veja objetivo de ponto de recuperação.

RTO

Veja objetivo de tempo de recuperação.

runbook

Um conjunto de procedimentos manuais ou automatizados necessários para realizar uma tarefa específica. Eles são normalmente criados para agilizar operações ou procedimentos repetitivos com altas taxas de erro.

S

SAML 2.0

Um padrão aberto que muitos provedores de identidade (IdPs) usam. Esse recurso permite o login único federado (SSO), para que os usuários possam fazer login Console de gerenciamento da AWS ou chamar as operações da AWS API sem que você precise criar um usuário no IAM para todos em sua organização. Para obter mais informações sobre a federação baseada em SAML 2.0, consulte Sobre a federação baseada em SAML 2.0 na documentação do IAM.

SCADA

Veja controle de supervisão e aquisição de dados.

SCP

Veja política de controle de serviço.

secret

Em AWS Secrets Manager, informações confidenciais ou restritas, como uma senha ou credenciais de usuário, que você armazena de forma criptografada. Consiste no valor secreto e em seus metadados. O valor secreto pode ser binário, uma única string ou várias strings. Para obter mais informações, consulte What's in a Secrets Manager secret? na documentação do Secrets Manager.

segurança desde a concepção

Uma abordagem em engenharia de sistemas que leva em consideração a segurança em todo o processo de desenvolvimento.

controle de segurança

Uma barreira de proteção técnica ou administrativa que impede, detecta ou reduz a capacidade de uma ameaça explorar uma vulnerabilidade de segurança. Existem quatro tipos primários de controles de segurança: preventivos, detectivos, responsivos e proativos.

hardening da segurança

O processo de reduzir a superfície de ataque para torná-la mais resistente a ataques. Isso pode incluir ações como remover recursos que não são mais necessários, implementar a prática recomendada de segurança de conceder privilégios mínimos ou desativar recursos desnecessários em arquivos de configuração.

sistema de gerenciamento de eventos e informações de segurança (SIEM)

Ferramentas e serviços que combinam sistemas de gerenciamento de informações de segurança (SIM) e gerenciamento de eventos de segurança (SEM). Um sistema SIEM coleta, monitora e analisa dados de servidores, redes, dispositivos e outras fontes para detectar ameaças e violações de segurança e gerar alertas.

automação de resposta de segurança

Uma ação predefinida e programada projetada para responder ou remediar automaticamente um evento de segurança. Essas automações servem como controles de segurança responsivos ou detectivos que ajudam você a implementar as melhores práticas AWS de segurança. Exemplos de ações de resposta automatizada incluem a modificação de um grupo de segurança da VPC, a aplicação de patches em uma instância do Amazon EC2 ou a alternância de credenciais.

Criptografia do lado do servidor

Criptografia dos dados em seu destino, por AWS service (Serviço da AWS) quem os recebe.

política de controle de serviços (SCP)

Uma política que fornece controle centralizado sobre as permissões de todas as contas em uma organização no AWS Organizations. As SCPs definem barreiras de proteção ou estabelecem limites para as ações que um administrador pode delegar a usuários ou perfis. É possível usar SCPs como listas de permissão ou de negação para especificar quais serviços ou ações são permitidos ou proibidos. Para obter mais informações, consulte Políticas de controle de serviço na AWS Organizations documentação.

service endpoint (endpoint de serviço)

O URL do ponto de entrada para um AWS service (Serviço da AWS). Você pode usar o endpoint para se conectar programaticamente ao serviço de destino. Para obter mais informações, consulte Endpoints do AWS service (Serviço da AWS) na Referência geral da AWS.

acordo de serviço (SLA)

Um acordo que esclarece o que uma equipe de TI promete fornecer aos clientes, como tempo de atividade e performance do serviço.

indicador de nível de serviço (SLI)

Uma avaliação de um aspecto de performance de um serviço, como taxa de erro, disponibilidade ou throughput.

objetivo de nível de serviço (SLO)

Uma métrica alvo que representa a integridade de um serviço, conforme avaliado por um indicador de nível de serviço.

modelo de responsabilidade compartilhada

Um modelo que descreve a responsabilidade com a qual você compartilha AWS pela segurança e conformidade na nuvem. AWS é responsável pela segurança da nuvem, enquanto você é responsável pela segurança na nuvem. Para obter mais informações, consulte o Modelo de responsabilidade compartilhada.

Inteligência artificial sombria

Aplicativos de IA não autorizados criados ou usados fora dos canais controlados dentro de uma organização.

SIEM

Veja sistema de gerenciamento de eventos e informações de segurança.

ponto único de falha (SPOF)

Uma falha em um único componente crítico de uma aplicação que pode interromper o sistema.

SLA

Veja cacordo de serviço.

SLI

Veja indicador de nível de serviço.

SLO

Veja objetivo de nível de serviço.

modelo dividir e semear

Um padrão para escalar e acelerar projetos de modernização. À medida que novos recursos e lançamentos de produtos são definidos, a equipe principal se divide para criar novas equipes de produtos. Isso ajuda a escalar os recursos e os serviços da sua organização, melhora a produtividade do desenvolvedor e possibilita inovações rápidas. Para obter mais informações, consulte Phased approach to modernizing applications in the Nuvem AWS.

SPOF

Veja ponto único de falha.

esquema em estrela

Uma estrutura organizacional de banco de dados que usa uma grande tabela de fatos para armazenar dados transacionais ou medidos e usa uma ou mais tabelas dimensionais menores para armazenar atributos de dados. Essa estrutura foi projetada para ser usada em um data warehouse ou para fins de inteligência comercial.

padrão strangler fig

Uma abordagem à modernização de sistemas monolíticos que consiste em reescrever e substituir incrementalmente a funcionalidade do sistema até que o sistema herdado possa ser desativado. Esse padrão usa a analogia de uma videira que cresce e se torna uma árvore estabelecida e, eventualmente, supera e substitui sua hospedeira. O padrão foi apresentado por Martin Fowler

como forma de gerenciar riscos ao reescrever sistemas monolíticos. Para ver um exemplo de como aplicar esse padrão, consulte Modernizando os serviços web legados da Microsoft ASP.NET (ASMX) de forma incremental usando contêineres e o Amazon API Gateway.

sub-rede

Um intervalo de endereços IP na VPC. Cada sub-rede fica alocada em uma única zona de disponibilidade.

controle supervisório e aquisição de dados (SCADA)

Na manufatura, um sistema que usa hardware e software para monitorar ativos físicos e operações de produção.

symmetric encryption (criptografia simétrica)

Um algoritmo de criptografia que usa a mesma chave para criptografar e descriptografar dados.

testes sintéticos

Testar um sistema de forma que simule as interações do usuário para detectar possíveis problemas ou monitorar a performance. Você pode usar o Amazon CloudWatch Synthetics para criar esses testes.

prompt do sistema

Uma técnica para fornecer contexto, instruções ou orientações a um LLM a fim de direcionar seu comportamento. Os prompts do sistema ajudam a definir o contexto e a estabelecer regras para interações com os usuários.

T

tags

Key-value pares que atuam como metadados para organizar seus AWS recursos. As tags podem ajudar você a gerenciar, identificar, organizar, pesquisar e filtrar recursos da . Para obter mais informações, consulte Marcar seus recursos do AWS.

variável-alvo

O valor que você está tentando prever no ML supervisionado. Ela também é conhecida como variável de resultado. Por exemplo, em uma configuração de fabricação, a variável-alvo pode ser um defeito do produto.

lista de tarefas

Uma ferramenta usada para monitorar o progresso por meio de um runbook. Uma lista de tarefas contém uma visão geral do runbook e uma lista de tarefas gerais a serem concluídas. Para cada tarefa geral, ela inclui o tempo estimado necessário, o proprietário e o progresso.

ambiente de teste

Veja ambiente.

treinamento

O processo de fornecer dados para que seu modelo de ML aprenda. Os dados de treinamento devem conter a resposta correta. O algoritmo de aprendizado descobre padrões nos dados de treinamento que mapeiam os atributos dos dados de entrada no destino (a resposta que você deseja prever). Ele gera um modelo de ML que captura esses padrões. Você pode usar o modelo de ML para obter previsões de novos dados cujo destino você não conhece.

ferramenta

Uma função ou API que um agente pode invocar para realizar operações em sistemas externos.

gateway de trânsito

Um hub de trânsito de rede que pode ser usado para interconectar as VPCs e as redes on-premises. Para obter mais informações, consulte O que é um gateway de trânsito na AWS Transit Gateway documentação.

fluxo de trabalho baseado em troncos

Uma abordagem na qual os desenvolvedores criam e testam recursos localmente em uma ramificação de recursos e, em seguida, mesclam essas alterações na ramificação principal. A ramificação principal é então criada para os ambientes de desenvolvimento, pré-produção e produção, sequencialmente.

Acesso confiável

Conceder permissões a um serviço que você especifica para realizar tarefas em sua organização AWS Organizations e em suas contas em seu nome. O serviço confiável cria um perfil vinculado ao serviço em cada conta, quando esse perfil é necessário, para realizar tarefas de gerenciamento para você. Para obter mais informações, consulte Usando AWS Organizations com outros AWS serviços na AWS Organizations documentação.

tuning (ajustar)

Alterar aspectos do processo de treinamento para melhorar a precisão do modelo de ML. Por exemplo, você pode treinar o modelo de ML gerando um conjunto de rótulos, adicionando rótulos e repetindo essas etapas várias vezes em configurações diferentes para otimizar o modelo.

equipe de duas pizzas

Uma pequena DevOps equipe que você pode alimentar com duas pizzas. Uma equipe de duas pizzas garante a melhor oportunidade possível de colaboração no desenvolvimento de software.

U

incerteza

Um conceito que se refere a informações imprecisas, incompletas ou desconhecidas que podem minar a confiabilidade dos modelos preditivos de ML. Há dois tipos de incertezas: a incerteza epistêmica é causada por dados limitados e incompletos, enquanto a incerteza aleatória é causada pelo ruído e pela aleatoriedade inerentes aos dados.

tarefas indiferenciadas

Também conhecido como trabalho pesado, trabalho necessário para criar e operar um aplicativo, mas que não fornece valor direto ao usuário final nem oferece vantagem competitiva. Exemplos de tarefas indiferenciadas incluem aquisição, manutenção e planejamento de capacidade.

ambientes superiores

Veja ambiente.

V

aspiração

Uma operação de manutenção de banco de dados que envolve limpeza após atualizações incrementais para recuperar armazenamento e melhorar a performance.

controle de versões

Processos e ferramentas que rastreiam mudanças, como alterações no código-fonte em um repositório.

emparelhamento de VPC

Uma conexão entre duas VPCs que permite rotear tráfego usando endereços IP privados. Para ter mais informações, consulte O que é emparelhamento de VPC? na documentação da Amazon VPC.

Vulnerabilidade

Uma falha de software ou hardware que compromete a segurança do sistema.

W

cache quente

Um cache de buffer que contém dados atuais e relevantes que são acessados com frequência. A instância do banco de dados pode ler do cache do buffer, o que é mais rápido do que ler da memória principal ou do disco.

dados mornos

Dados acessados raramente. Ao consultar esse tipo de dados, consultas moderadamente lentas geralmente são aceitáveis.

função de janela

Uma função SQL que executa um cálculo em um grupo de linhas que se relacionam de alguma forma com o registro atual. As funções de janela são úteis para processar tarefas, como calcular uma média móvel ou acessar o valor das linhas com base na posição relativa da linha atual.

workload

Uma coleção de códigos e recursos que geram valor empresarial, como uma aplicação voltada para o cliente ou um processo de backend.

workstreams

Grupos funcionais em um projeto de migração que são responsáveis por um conjunto específico de tarefas. Cada workstream é independente, mas oferece suporte aos outros workstreams do projeto. Por exemplo, o workstream de portfólio é responsável por priorizar aplicações, planejar ondas e coletar metadados de migração. O workstream de portfólio entrega esses ativos ao workstream de migração, que então migra os servidores e as aplicações.

WORM

Veja gravação única e várias leituras.

WQF

Veja AWS Workload Qualification Framework.

gravação única e várias leituras (WORM)

Um modelo de armazenamento que grava dados uma única vez e evita que os dados sejam excluídos ou modificados. Os usuários autorizados podem ler os dados quantas vezes forem necessárias, mas não podem alterá-los. Essa infraestrutura de armazenamento de dados é considerada imutável.

Z

exploração de dia zero

Um ataque, normalmente malware, que tira proveito de uma vulnerabilidade zero-day.

vulnerabilidade de dia zero

Uma falha ou vulnerabilidade não mitigada em um sistema de produção. Os agentes de ameaças podem usar esse tipo de vulnerabilidade para atacar o sistema. Os desenvolvedores frequentemente ficam cientes da vulnerabilidade como resultado do ataque.

prompt zero shot

Fornecer a um LLM instruções para realizar uma tarefa, mas sem exemplos (shots) que possam ajudar a orientá-lo. O LLM deve usar seu conhecimento pré-treinado para lidar com a tarefa. A eficácia dos prompts zero-shot depende da complexidade da tarefa e da qualidade do prompt. Veja também prompts few-shot.

aplicação zumbi

Uma aplicação que tem um uso médio de CPU e memória inferior a 5%. Em um projeto de migração, é comum retirar essas aplicações.

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