Padrões e fluxos de trabalho de IA da Agentic em AWS

AWS Orientação prescritiva

AWS Orientação prescritiva: Padrões e fluxos de trabalho de IA da Agentic em AWS

Table of Contents

Padrões e fluxos de trabalho de IA agentes em AWS

Aaron Sempf e Andrew Hooker, Amazon Web Services

Julho de 2025 (histórico do documento)

As organizações estão adotando grandes modelos de linguagem (LLMs) e agentes de software para resolver problemas dinâmicos de vários domínios usando uma nova disciplina arquitetônica chamada padrões agentes. Os padrões de agência são projetos fundamentais e construções modulares que são usados para projetar e orquestrar agentes de IA orientados a objetivos em muitos contextos.

Público-alvo

Este guia é destinado a arquitetos, desenvolvedores e líderes de produtos que desejam criar aplicativos inteligentes que vão além da lógica estática, lógica simbólica e automação determinística.

Objetivos

Este guia fornece uma estrutura de design e uma abordagem de implementação para sistemas de agentes de IA que operam de forma autônoma, permanecendo controláveis e alinhados com suas metas. Ele conecta padrões arquitetônicos orientados por eventos a várias alternativas de agentes, demonstrando como criar sistemas de agentes de nível de produção usando arquiteturas nativas em nuvem. Os seguintes assuntos são discutidos neste guia:

- Padrões do agente — Os padrões do agente são modelos de design reutilizáveis que descrevem a estrutura e o comportamento de agentes individuais. Isso inclui agentes de raciocínio, agentes de recuperação aumentada, agentes de codificação, interfaces de voz, orquestradores de fluxo de trabalho e sistemas multiagentes colaborativos. Cada padrão ilustra como os agentes percebem, raciocinam, agem e aprendem, mapeados. Serviços da AWS
- Fluxos de trabalho do LLM — Os fluxos de trabalho se concentram em como os agentes usam LLMs para raciocinar. Eles exploram estratégias de estímulo e mecanismos de planejamento e descrevem como LLMs são usados não apenas para gerar texto, mas também para impulsionar comportamentos estruturados, interpretáveis e confiáveis em um loop de agentes.
- Padrões de fluxo de trabalho agentes — Os padrões de fluxo de trabalho descrevem como vários agentes, ferramentas e ambientes interagem para formar sistemas autônomos. Isso inclui padrões para orquestração de tarefas, delegação de subagentes, coordenação baseada em eventos,

observabilidade e controle. Esses aspectos promovem arquiteturas de IA escaláveis, combináveis e auditáveis.

Sobre esta série de conteúdo

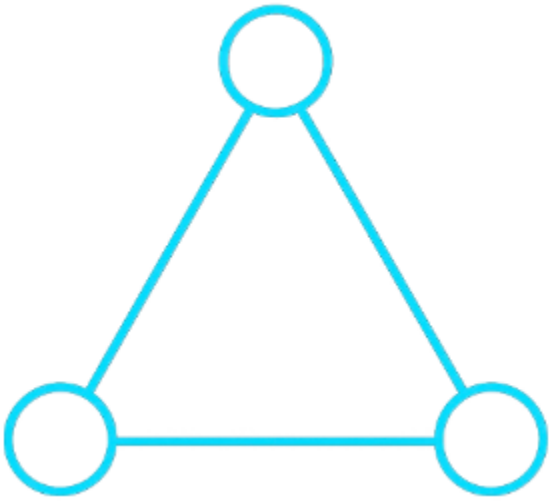
Este guia faz parte de uma série sobre IA agente em AWS. Para obter mais informações e ver os outros guias desta série, consulte Agentic AI no site de Orientação AWS Prescritiva.

Padrões de agentes

Os padrões de agentes são blocos de construção reutilizáveis e compostos que podem ser personalizados para domínios, casos de uso e níveis de complexidade específicos. Os sistemas Agentic diferem, no entanto, das aplicações tradicionais. No centro de todos os projetos de agentes de IA está um modelo conceitual ancorado nos três princípios fundamentais a seguir:

- Assíncrono — os agentes operam em ambientes pouco acoplados e ricos em eventos
- Autonomia — Os agentes agem de forma independente, sem controle humano ou externo
- Agência — Os agentes agem com um propósito, em nome de um usuário ou sistema, em direção a objetivos específicos

O triângulo no diagrama a seguir representa os principais blocos de construção de um agente de software: percepção, razão e ação. Isso permite que um sistema agente observe, tome decisões e aja dentro de seu ambiente.



Por design, os padrões agentes fornecem uma linguagem de design modular para a criação de sistemas de IA, o que significa que eles são acessíveis, operacionais, extensíveis e prontos para produção. O projeto desses sistemas requer atenção cuidadosa às três dimensões inter-relacionadas a seguir, que serão discutidas mais adiante neste guia.

Nesta seção

- Tool-based agentes para servidores
- Computer-use agentes
- Agentes de codificação
- Agentes de fala e voz
- Agentes de orquestração de fluxo de trabalho
- Memory-augmented agentes
- Agentes de simulação e de teste
- Agentes observadores e de monitoramento
- Multi-agent colaboração

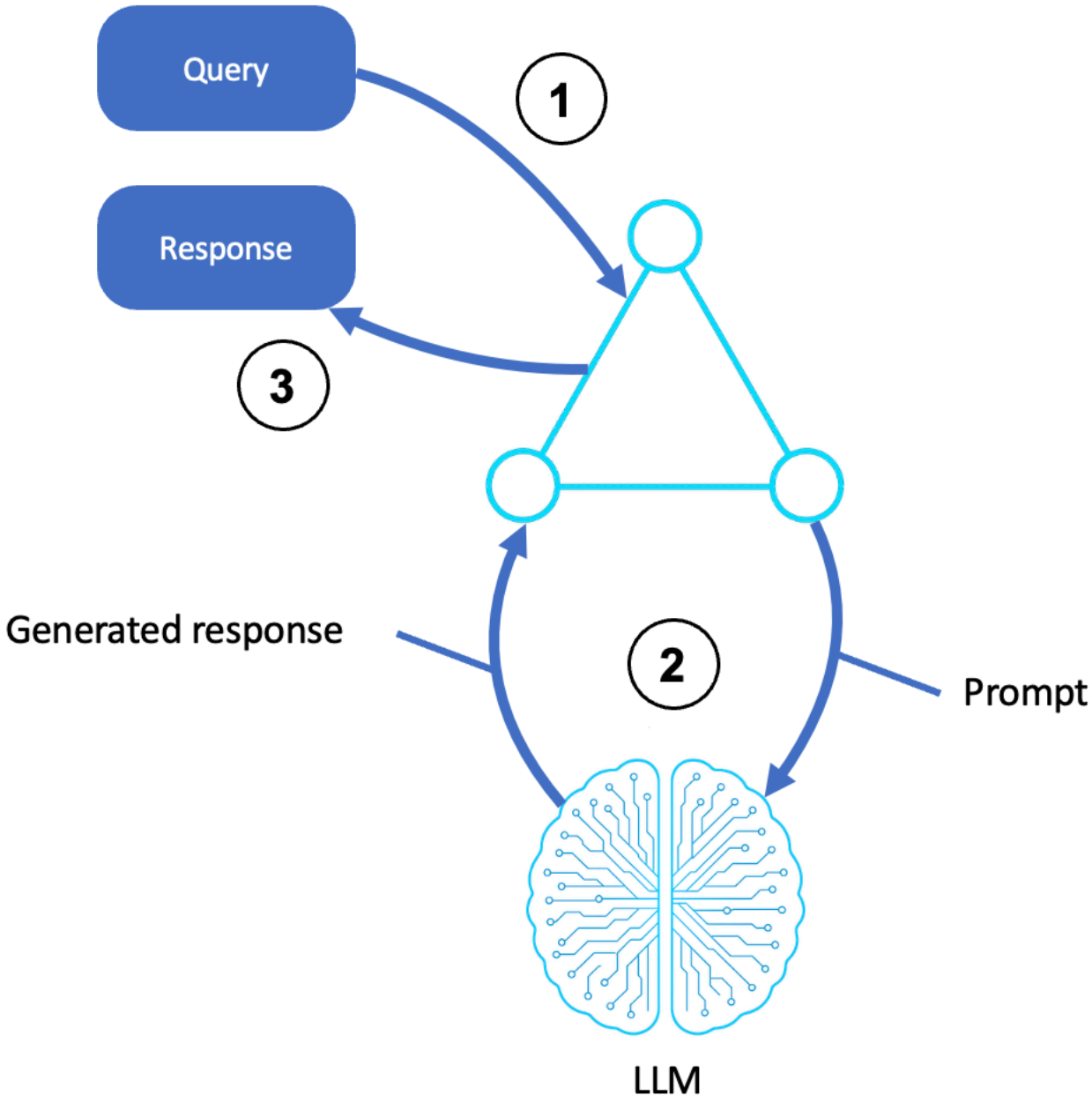
Agentes básicos de raciocínio

Um agente de raciocínio básico é a forma mais simples de IA agente que realiza inferência lógica ou tomada de decisão em resposta a uma consulta. Ele aceita a entrada de um usuário ou sistema e processa consultas e gera respostas usando prompts estruturados.

Esse padrão é útil para tarefas que exigem raciocínio, classificação ou resumo em uma única etapa com base em um determinado contexto. Ele não usa memória, ferramentas ou gerenciamento de estado, o que o torna sem estado, leve e altamente combinável em grandes fluxos de trabalho.

Arquitetura

O fluxo de um agente de raciocínio básico é mostrado no diagrama a seguir:



Description

1. Recebe uma entrada
   - Um usuário, sistema ou agente upstream envia uma consulta ou instrução.
   - A entrada é transferida para o shell do agente ou para a camada de orquestração.
   - Essa etapa inclui qualquer pré-processamento, modelagem imediata e identificação de metas.
2. Invoca o LLM

- O agente transforma a consulta em um prompt estruturado e a envia para um LLM (por exemplo, por meio do Amazon Bedrock).
- O LLM gera uma resposta com base no prompt usando conhecimento e contexto pré-treinados.
- A saída gerada pode incluir etapas de raciocínio (cadeia de pensamento), respostas finais ou opções classificadas.

3. Retorna uma resposta
   - A saída gerada é retransmitida para a interface do agente.
   - Isso pode incluir formatação, pós-processamento ou uma resposta de API.

Capacidades

- Suporta linguagem natural ou entrada estruturada
- Usa engenharia rápida para orientar o comportamento
- Sem estado e escalável
- Pode ser incorporado à interface do usuário, CLI, APIs e pipelines

Limitações

- Sem memória ou consciência histórica
- Sem interação com ferramentas externas ou fontes de dados
- Limitado ao que o LLM sabe no momento da inferência

Casos de uso comuns

- Perguntas e respostas conversacionais
- Explicações e resumos de políticas
- Orientação para a tomada de decisões
- Fluxos de chatbot leves e automatizados
- Classificação, rotulagem e pontuação

Orientação para implementação

Você pode usar as seguintes ferramentas e serviços para criar um agente de raciocínio básico:

- Amazon Bedrock para invocação de LLM (Anthropic, AI21, Meta)
- Amazon API Gateway ou AWS Lambda para expô-lo como um microsserviço sem estado
- Modelos de prompt armazenados no Parameter Store ou como código AWS Secrets Manager

Resumo

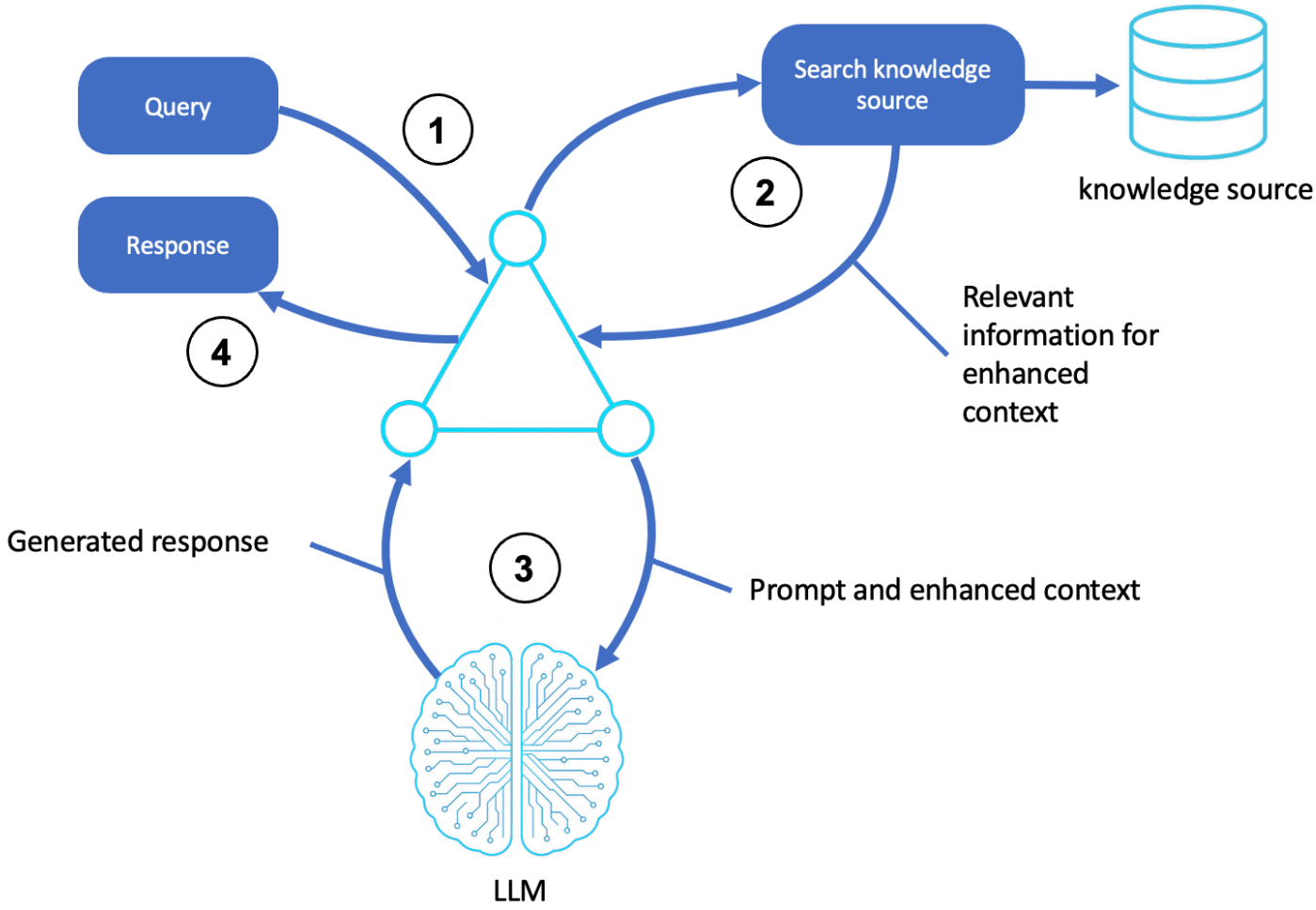
O agente de raciocínio básico é fundamental por causa de sua estrutura simples. Ele tem recursos essenciais que transformam metas em caminhos de raciocínio que levam a resultados inteligentes. Esse padrão geralmente é um ponto de partida para padrões avançados, como agentes baseados em ferramentas e agentes que usam geração aumentada de recuperação (RAG). Também é um componente confiável e modular de grandes fluxos de trabalho.

Agente RAG

Retrieval-augmented a geração (RAG) é uma técnica que combina recuperação de informações com geração de texto para criar respostas precisas e contextuais. O RAG permite que os agentes recuperem informações externas relevantes antes de contratar o LLM. Ele amplia a memória efetiva e a precisão do raciocínio de um agente ao basear suas decisões em informações atualizadas, factuais ou específicas do domínio. Em contraste com os LLMs sem estado que dependem exclusivamente de pesos pré-treinados, o RAG tem uma camada externa de pesquisa de conhecimento que aprimora dinamicamente as solicitações com o contexto.

Arquitetura

A lógica do padrão RAG é ilustrada no diagrama a seguir:



Description

1. Recebe uma consulta
   - Um usuário ou sistema upstream envia uma consulta ou meta ao agente.
   - O shell do agente aceita a solicitação e a formata como uma solicitação de raciocínio.
2. Pesquisa uma fonte externa
   - O agente identifica os conceitos e a intenção da consulta.
   - Ele consulta uma fonte de conhecimento, como um repositório vetorial, banco de dados ou índice de documentos usando pesquisa semântica ou correspondência de palavras-chave.
   - As passagens, documentos ou entidades mais relevantes são recuperados para uso na próxima etapa.
3. Gera uma resposta contextual

- O agente aumenta o prompt com as informações recuperadas, formando uma entrada contextualizada para o LLM.
- O LLM processa qualquer entrada usando raciocínio generativo (por exemplo, cadeia de pensamento ou reflexão) para produzir uma resposta precisa.

4. Retorna a saída final
   - O agente prepara a saída envolvendo-a em qualquer cabeçalho de comunicação ou na formatação necessária e, em seguida, a retorna ao usuário ou ao sistema de chamada.
   - (Opcional) Os documentos recuperados e a saída do LLM podem ser registrados, pontuados e armazenados na memória para futuras consultas.

Capacidades

- Fact-grounded produção mesmo em domínios de longa duração ou específicos da empresa
- Extensão de memória sem ajustar o modelo
- Contexto dinâmico baseado em cada consulta e estado do usuário
- Totalmente compatível com bancos de dados vetoriais, índices semânticos e filtragem de metadados

Casos de uso comuns

- Assistentes de conhecimento corporativo
- Bots de conformidade regulatória
- Co-pilotos de suporte ao cliente
- Search-enhanced chatbots
- Agentes de documentação para desenvolvedores

Orientação para implementação

Use as seguintes ferramentas e serviços para criar um agente que usa o RAG:

- Amazon Bedrock para invocação de LLM
- Amazon Kendra OpenSearch ou Amazon Aurora para documentação ou pesquisa estruturada de dados

- Amazon Simple Storage Service (Amazon S3) (Amazon S3) para armazenamento de documentos
- AWS Lambda para orquestrar a pesquisa, o prompt e a inferência do LLM
- Knowledge-based integrações com agentes (usando plug-ins de memória, recuperadores semânticos ou Amazon Bedrock)

Resumo

O agente RAG conecta o raciocínio do modelo estático à inteligência dinâmica do mundo real. Ele capacita os agentes com a capacidade de pesquisar o que não sabem, sintetizar respostas a partir do conhecimento recuperado e produzir respostas auditáveis e de alta confiança.

Os padrões RAG são a base para a criação de agentes inteligentes que escalam o acesso ao conhecimento sem reciclagem. Geralmente, é um precursor de padrões de orquestração mais complexos que envolvem o uso de ferramentas, planejamento e memória de longo prazo.

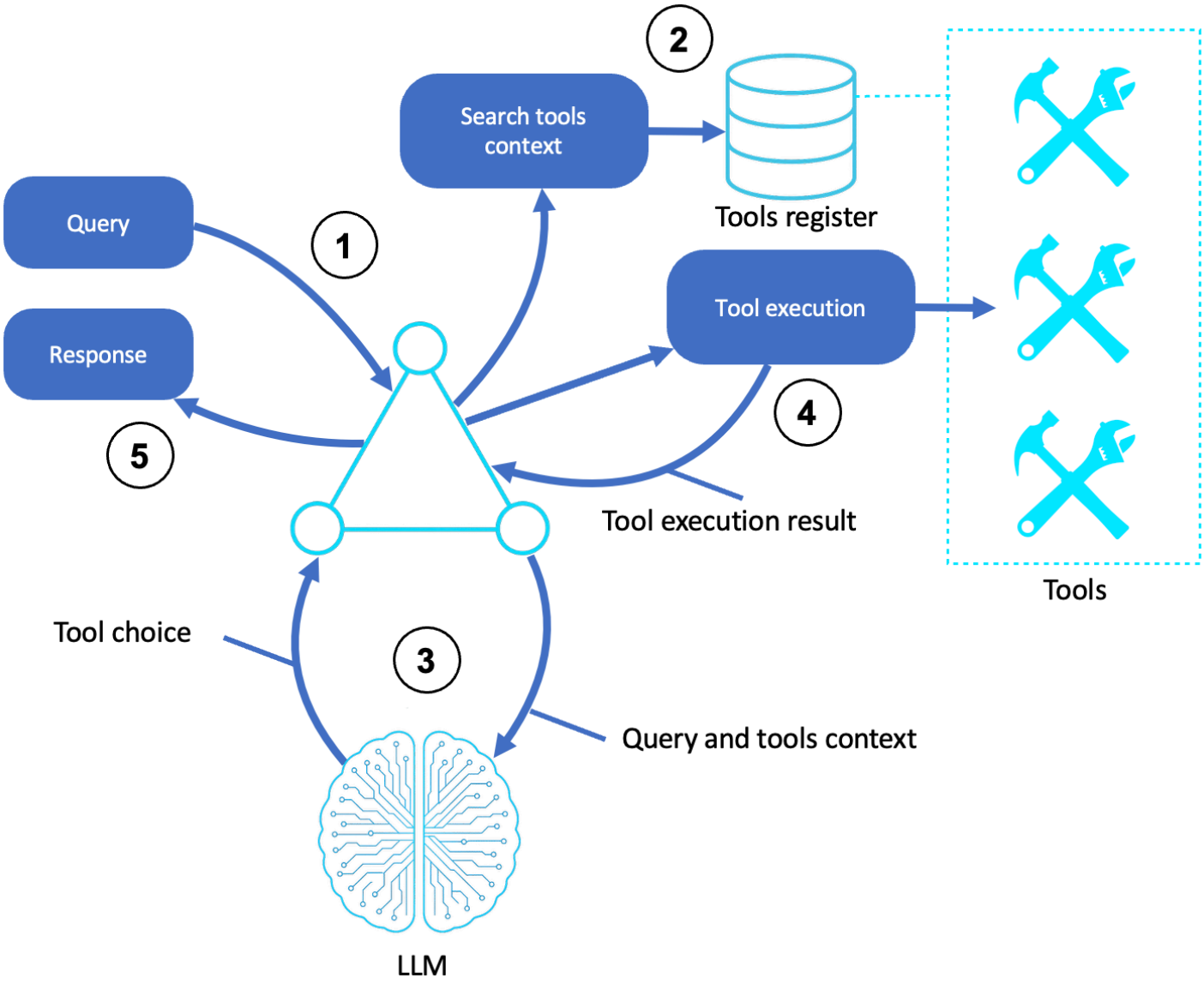
Tool-based agentes para chamar funções

Tool-based os agentes ampliam as capacidades dos agentes de raciocínio invocando funções externas ou APIs para concluir tarefas que vão além do raciocínio somente de linguagem. Esse padrão usa um LLM para decidir qual ferramenta usar e, em seguida, gera argumentos de chamada e incorpora a saída de uma ferramenta em seu ciclo de raciocínio.

Esse padrão permite que os agentes ajam em vez de apenas fornecer respostas. A interface da ferramenta representa qualquer recurso que possa ser chamado, desde cálculos aritméticos e pesquisas de banco de dados até APIs externas e serviços em nuvem.

Arquitetura

Um agente baseado em ferramentas para chamar funções é mostrado no diagrama a seguir:



Description

1. Recebe consulta
   - O agente recebe uma consulta ou tarefa em linguagem natural do usuário ou do sistema de chamada.
2. Pesquisas por ferramentas
   - O agente usa metadados internos ou um registro de ferramentas para pesquisar ferramentas, esquemas e recursos relevantes disponíveis.
3. Seleciona e invoca ferramentas

- O LLM recebe os metadados da consulta e da ferramenta (por exemplo, nomes de funções, tipos de entrada e descrições) em seu prompt.
- Ele escolhe a ferramenta mais relevante, constrói argumentos de entrada e retorna uma chamada de função estruturada.

4. Executa a ferramenta escolhida
   - O shell do agente ou o executor de ferramentas executa a função selecionada e retorna o resultado (por exemplo, uma saída de API, valor de banco de dados ou computação).
5. Retorna uma resposta
   - O LLM passa os resultados para o agente, diretamente ou como parte de um prompt atualizado. Em seguida, ele retorna um resultado em linguagem natural.

Capacidades

- Seleção dinâmica de ferramentas com base no contexto da tarefa
- Schema-based solicitação (OpenAPI, esquema AWS JSON, interface de função)
- Interpretação dos resultados e encadeamento dos resultados no raciocínio
- Operações sem estado ou com reconhecimento de sessão

Casos de uso comuns

- Assistentes virtuais com acesso externo a dados
- Calculadoras e estimadores financeiros
- API-based trabalhadores do conhecimento
- LLMs que invocam, SageMaker endpoints AWS Lambda da Amazon e serviços SaaS

Orientação para implementação

Use o seguinte para criar agentes baseados em ferramentas para chamar funções:

- Amazon Bedrock com suporte para chamadas funcionais (Anthropic Claude)
- AWS Lambda como um back-end de execução de ferramentas
- Amazon API Gateway ou AWS Step Functions para orquestração de ferramentas

- Amazon DynamoDB ou Amazon Relational Database Service (Amazon RDS) para metadados de ferramentas sensíveis ao contexto
- EventBridge Pipelines da Amazon ou AWS Step Functions que mapeiam estados para rotear saídas

Resumo

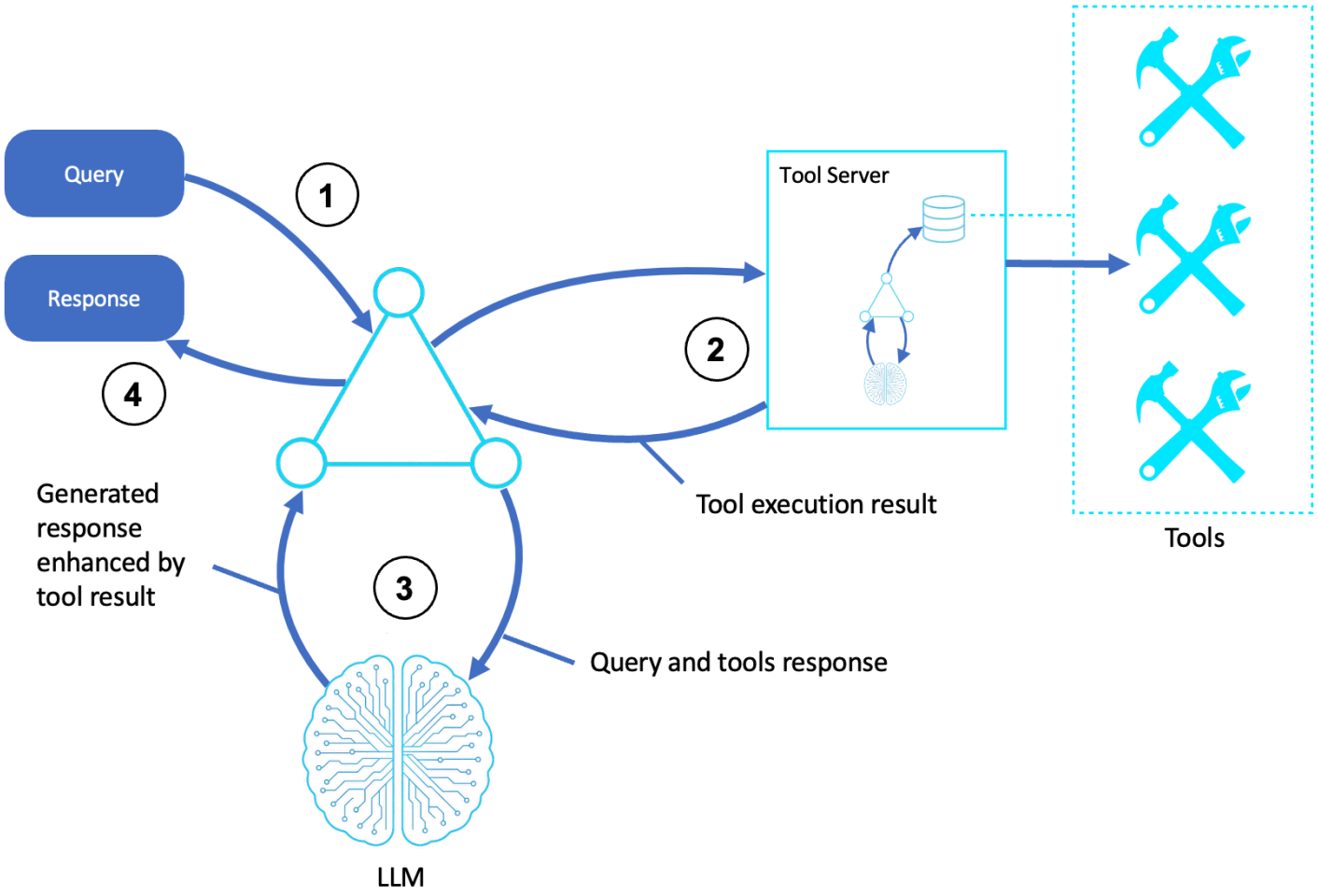
Tool-based agentes de chamada de funções representam uma mudança da compreensão da linguagem para a execução de ações. Esses agentes invocam ferramentas dinâmicas e sensíveis ao contexto, mantendo o raciocínio do LLM, transformando assistentes passivos em sistemas que concluem tarefas, acessam serviços e integram operações comerciais. Esse padrão é um componente importante da IA agente em ambientes corporativos, especialmente quando combinado com esquemas declarativos, estruturas de autorização e sistemas multiagentes.

Tool-based agentes para servidores

Tool-based agentes para servidores aprimoram os agentes de chamada de funções delegando a execução da ferramenta a um servidor externo que tem um ambiente de tempo de execução dedicado para ferramentas, scripts e agentes compostos. Diferentemente das chamadas de função em linha que o loop do agente seleciona e invoca, os agentes baseados em servidor terceirizam a lógica e o pipeline de execução para outros agentes ou sistemas. Isso fornece recursos avançados, como encadeamento de várias ferramentas, execução isolada e raciocínio especializado. Os servidores de ferramentas são ideais para ações complexas, com estado ou que consomem muitos recursos, nas quais as próprias ferramentas podem envolver modelos de IA, regras de negócios ou ambientes separados.

Arquitetura

O seguinte é um padrão para agentes baseados em ferramentas para servidores:



Description

1. Recebe consulta
   - Um usuário ou sistema envia uma solicitação ao shell do agente.
   - O agente interpreta a consulta e se prepara para enviá-la para um servidor de ferramentas.
2. Executa processos do servidor de ferramentas
   - O agente envia a tarefa, junto com os parâmetros estruturados, para um servidor de ferramentas.
   - O servidor de ferramentas pode então:
     - Execute scripts ou lógica em sistemas computacionais dedicados (por exemplo AWS Lambda, contêineres ou Amazon SageMaker)
     - Use seu próprio subagente com raciocínio LLM para selecionar e executar ferramentas
     - Gerencie dependências, novas tentativas ou fluxos de execução em várias etapas

    - Resultados de saída para o agente primário quando a tarefa estiver concluída

3. Usa o raciocínio LLM com a saída da ferramenta
    - O agente invoca um LLM, transmitindo a consulta original e o resultado do servidor de ferramentas como parte do prompt.
    - O LLM sintetiza uma resposta que incorpora as informações recém-adquiridas.
4. Retorna uma resposta
    - O agente retorna uma resposta estruturada ou em linguagem natural ao usuário ou ao sistema de chamada.
    - (Opcional) Os resultados podem ser armazenados na memória ou nos registros de auditoria.

Capacidades

- As ferramentas são invocadas fora do ciclo de execução do agente primário
- A execução da ferramenta pode envolver chamadas LLM, cadeias lógicas ou subagentes
- O agente atua como um controlador ou despachante, não apenas como um invólucro de ferramentas
- Permite composição, escalabilidade e isolamento da lógica

Casos de uso comuns

- Orquestrando cadeias de modelos (por exemplo, combinando LLM, visão e código)
- AI-driven tubulações de automação
- DevOps agentes assistentes com roteiristas
- Agentes complexos de computação financeira, simulação ou otimização
- Ferramentas multimodais (por exemplo, combinando áudio, documentação e ação)

Orientação para implementação

Você pode criar esse padrão usando o seguinte Serviços da AWS:

- Amazon Bedrock (agente, host e inferência de LLM)
- AWS Lambda, Amazon ECS ou SageMaker endpoints da Amazon como tempo de execução do servidor de ferramentas AWS Fargate

- Amazon API Gateway ou AWS App Runner para expor as APIs do servidor de ferramentas
- Amazon EventBridge para mensagens dissociadas de agente para ferramenta
- AWS Step Functions ou AWS AppFabric para compor lógica multiagente no servidor de ferramentas

Resumo

Tool-based os agentes que usam servidores são altamente modulares e escaláveis. Eles separam a lógica de decisão da execução, o que permite que o agente primário permaneça leve enquanto transfere ações complexas ou confidenciais para outros sistemas. Isso é importante para a IA agente de nível corporativo, especialmente em ambientes que exigem governança, observabilidade, isolamento, composição dinâmica ou qualquer combinação dos dois.

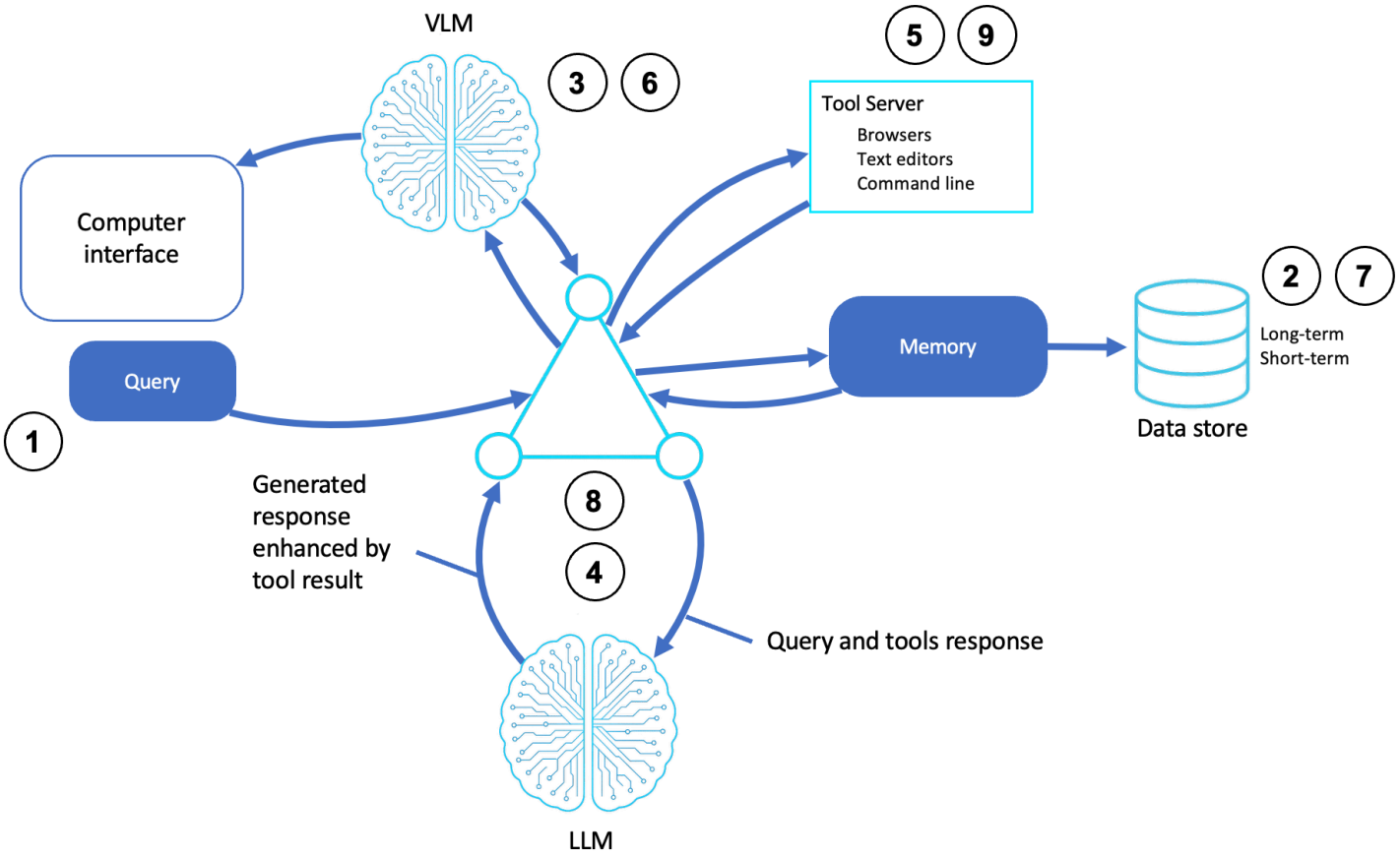
Computer-use agentes

Computer-use os agentes podem simular ou controlar ambientes digitais, como navegadores, terminais, sistemas de arquivos e aplicativos. Esses agentes interpretam a intenção do usuário, interagem com interfaces visuais e de texto e realizam ações direcionadas a objetivos combinando raciocínio LLM, modelos de linguagem visual (VLMs) e servidores de ferramentas que executam comandos ou simulam eventos de entrada.

Esse padrão é importante para automações práticas de IA, em que o agente funciona não apenas como um assistente, mas também como um proxy que executa ações como um humano faria, geralmente usando as mesmas ferramentas e ambientes.

Arquitetura

Um padrão de agente de uso do computador é mostrado no diagrama a seguir:



Description

1. Recebe uma consulta
   - Uma tarefa ou solicitação é fornecida por meio de uma interface de usuário, API ou interface de linguagem natural.
2. Acessa a memória
   - O agente recupera a memória de curto e longo prazo para relembrar comandos, metas e estados do sistema anteriores.
3. Analisa o contexto visual
   - Um VLM observa a tela do computador, o estado do sistema ou os elementos da interface do usuário para entender um determinado contexto e identificar itens acionáveis.
4. Razões por meio de um LLM
   - O LLM combina a consulta, o estado da memória, a ferramenta e a resposta do servidor para determinar a próxima ação.
5. Interage com o servidor de ferramentas

- O agente invoca ferramentas hospedadas em um servidor, que podem incluir o seguinte:
  - Navegadores (por exemplo, Chrome sem cabeçalho) e ambientes de shell
  - Editores de texto e código
  - Interfaces de script personalizadas

6. Atualiza as entradas visuais
   - Se a interface do usuário do sistema mudar ou for necessária uma observação adicional, o VLM poderá reanalisar o estado da tela ou os buffers de texto.
7. Atualiza a memória
   - Novos insights, estados do sistema ou feedback do usuário são gravados na memória de curto e longo prazo.
8. Formula decisões e explicações finais
   - O LLM sintetiza resultados ou recomenda ações com base na consulta e na saída da ferramenta.
9. Retorna uma resposta
   - O agente retorna os resultados para a interface (por exemplo, uma tarefa concluída, confirmação ou conteúdo gerado).

Capacidades

- Raciocínio multimodal com entradas visuais e textuais
- Controle sobre aplicativos por meio de simulações ou entradas API-driven
- Gerenciamento de memória para estado persistente
- Autonomia na execução de sequências (fluxos de várias etapas)

Casos de uso comuns

- Desenvolvedores de IA que escrevem e executam código em IDEs
- Computer-use agentes para fluxos de trabalho digitais repetitivos
- Usuários simulados para testes de software e garantia de qualidade
- Agentes de acessibilidade para navegar pelas interfaces de usuário por meio de instruções de voz ou de alto nível
- Automação inteligente de processos robóticos (RPA) aprimorada com o raciocínio

Orientação para implementação

- Você pode criar esse padrão usando o seguinte Serviços da AWS:
- Amazon Bedrock para LLM-based planejamento e raciocínio
- Amazon Elastic Compute Cloud (Amazon EC2) ou notebooks SageMaker Amazon para executar AWS Lambda servidores de ferramentas com ambientes de interface de usuário simulados
- Amazon Simple Storage Service (Amazon S3) ou Amazon DynamoDB para persistência de memória
- Amazon Rekognition (ou modelos personalizados) para análise de imagens de UI em cenários híbridos
- Amazon CloudWatch Logs ou AWS X-Ray para trilhas de observabilidade e auditoria

Resumo

Computer-use os agentes atuam como operadores digitais autônomos, preenchendo a lacuna entre as interações e ações humano-computador. AI-driven Ao incorporar memória, orquestração de ferramentas e VLMs, esses agentes podem interagir de forma adaptativa com sistemas projetados para humanos, executar ações, atualizar arquivos, navegar por menus e gerar respostas.

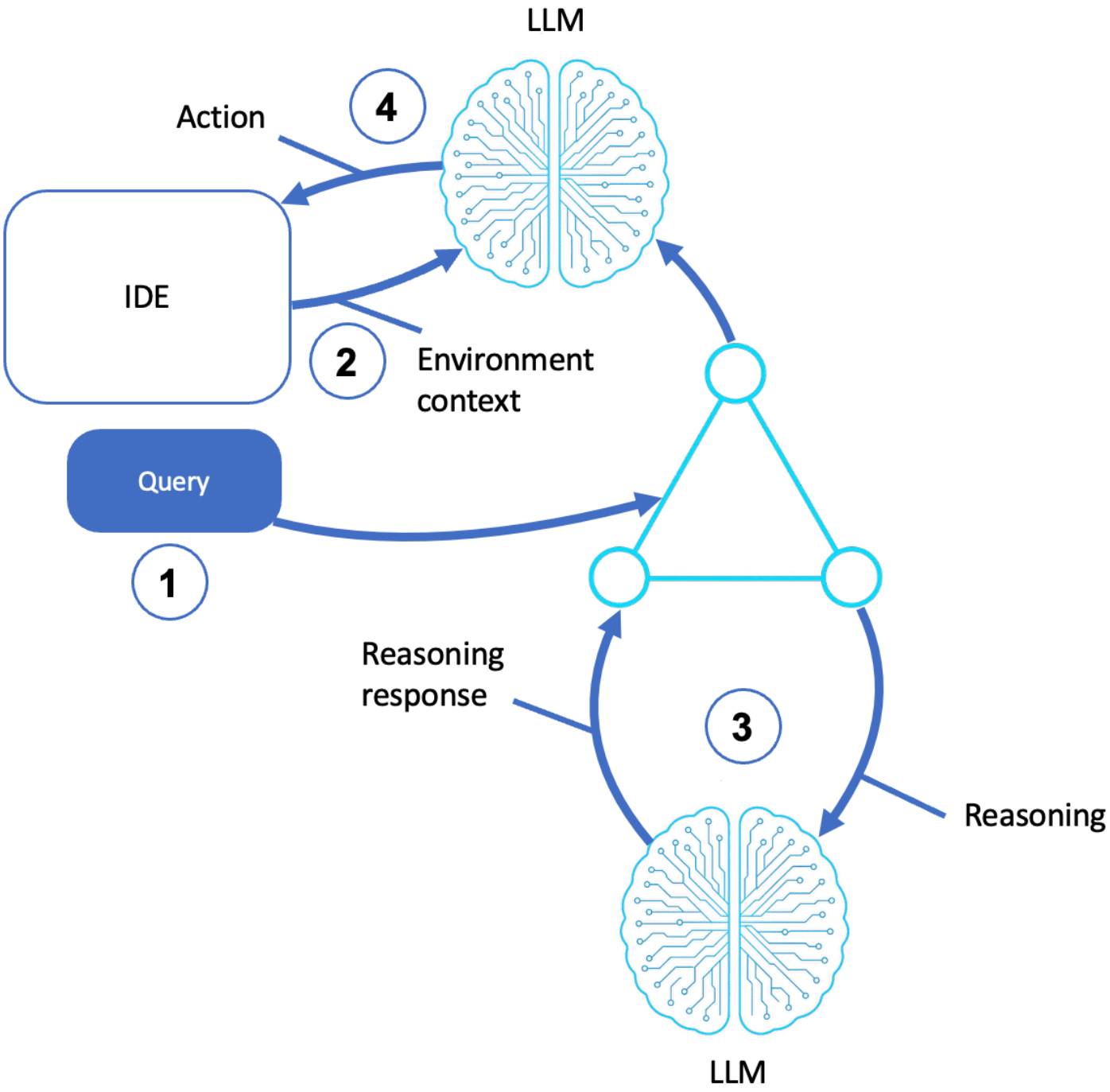
Agentes de codificação

Os agentes de codificação podem raciocinar sobre tarefas de programação, gerar ou modificar código e interagir com ambientes de desenvolvedores, como IDEs e CLIs. Esses agentes combinam a compreensão da linguagem natural com o raciocínio estruturado para auxiliar, aumentar e automatizar o desenvolvimento de software, desde a geração de funções até a correção de bugs e a criação de testes.

Diferentemente das ferramentas de preenchimento automático, os agentes de codificação interpretam ativamente as metas do usuário, consultam o ambiente de desenvolvimento em busca de contexto (por exemplo, ele abre arquivos e rastreia erros), identificam os requisitos e, em seguida, propõem e executam ações.

Arquitetura

Um padrão de agente codificador é mostrado no diagrama a seguir:



Description

1. Recebe consulta
   - O usuário fornece instruções em linguagem natural por meio de uma paleta de comandos, janela de bate-papo ou CLI (por exemplo, “Adicionar registro a esta função” ou “Refatorar para facilitar a leitura”).

2. Extrai o contexto do ambiente
   - O agente coleta o contexto do IDE, incluindo arquivos ativos, posição do cursor, trechos de código e tabelas de símbolos.
   - Ele emite mensagens de erro, resultados de testes e saídas de outros agentes.
3. Raciocínio LLM
   - O agente envia um prompt, incluindo a consulta e o contexto ambiental, para um LLM.
     - O LLM executa um raciocínio para determinar o seguinte:
     - O que precisa mudar
     - Como gerar uma solução
     - Qualquer etapa de refatoração, reescrita ou codificação
4. Executa ações
   - O LLM retorna a saída para o agente e a importa para o ambiente IDE ou runtime.
   - Isso pode incluir inserir ou modificar código, gerar comentários ou documentação e acionar tarefas posteriores de compilação, teste e linting.

Capacidades

- High-contextual reconhecimento (por exemplo, estado do IDE, cursor e árvore de sintaxe)
- Raciocínio iterativo de metas e feedback
- Planejamento de código opcional e separação de ações (por exemplo, primeiro motivo e depois ação)
- Funciona em fluxos de trabalho de desenvolvedores síncronos ou assíncronos

Casos de uso comuns

- Geração de código a partir de descrições de tarefas
- Refatoração e otimização de código
- Test-case geração e validação
- Explicações de erros e depuração
- Assistentes de documentação
- Co-pilotos de programação emparelhados

Orientação para implementação

- Você pode criar esse padrão usando as seguintes ferramentas e Serviços da AWS:
- Amazon Bedrock para LLM-driven geração e raciocínio
- Amazon Q Developer para sugestões e conclusões de codificação
- AWS Lambda ou Amazon Elastic Container Service (Amazon ECS) para executar e testar ambientes sandbox
- AWS Cloud9, extensões do VS Code ou integrações personalizadas de IDE para hospedar e avaliar o contexto
- Amazon Simple Storage Service (Amazon S3) (Amazon S3) para armazenar solicitações intermediárias, respostas e histórico de revisões

Resumo

Agentes de codificação são novas ferramentas de AI-powered desenvolvimento capazes de interpretar a linguagem natural, analisar o contexto, gerar alterações de código em várias etapas e integrar-se ao ciclo de vida do desenvolvimento de software.

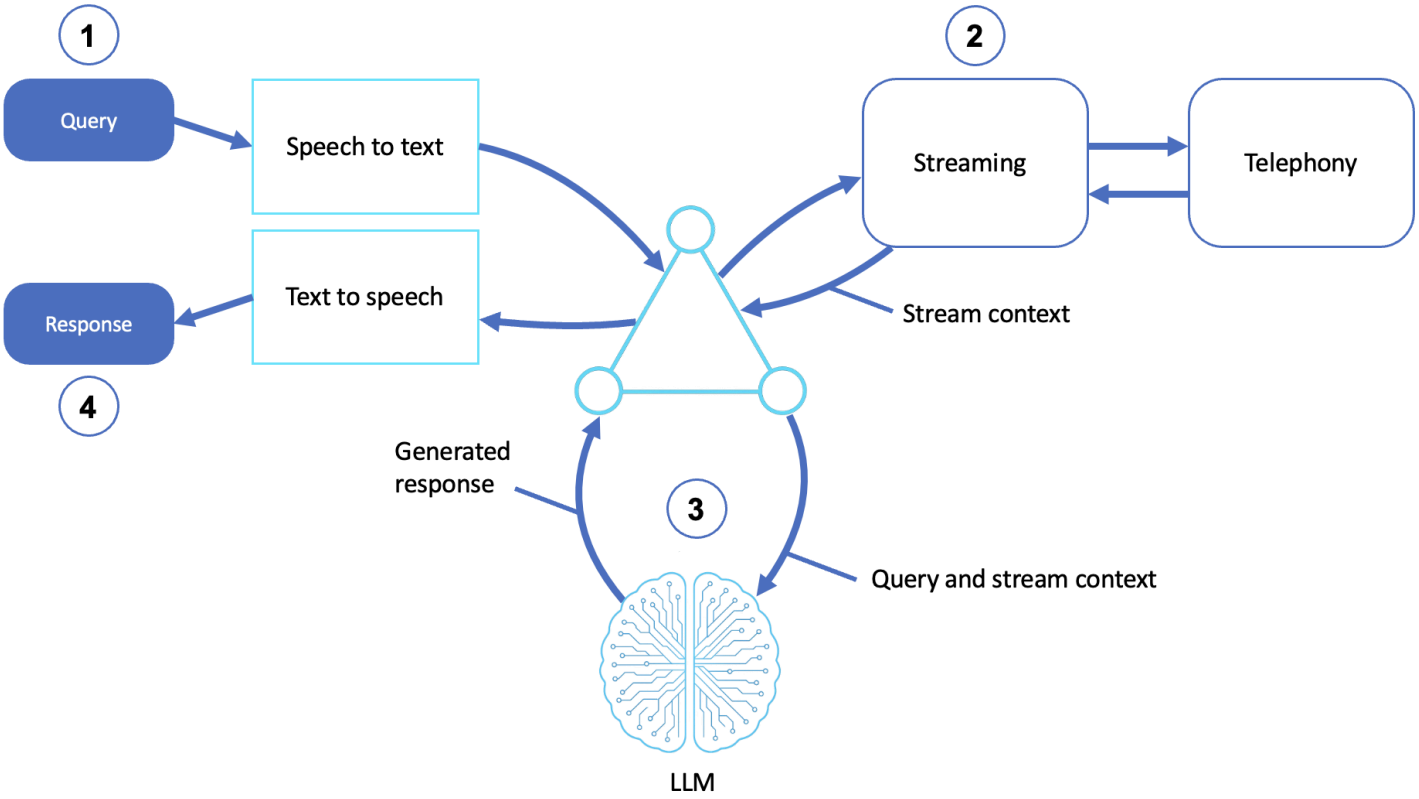
Agentes de fala e voz

Agentes de voz e voz interagem com os usuários por meio do diálogo falado. Esses agentes integram reconhecimento de fala, compreensão de linguagem natural e síntese de fala para permitir a IA conversacional em plataformas telefônicas, móveis, web e incorporadas.

Os agentes de voz são particularmente eficazes em ambientes sem usar as mãos, em tempo real ou orientados pela acessibilidade. Ao combinar interfaces de streaming com LLM-powered raciocínio, elas facilitam interações ricas e dinâmicas que parecem naturais para os usuários.

Arquitetura

Um agente de fala e voz é mostrado no diagrama a seguir:



Description

1. Recebe uma consulta de voz
   - O usuário envia uma solicitação para um telefone, microfone ou sistema embarcado.
   - Um módulo de fala para texto (STT) converte o áudio em texto.
2. Integra o contexto de streaming e telefonia
   - O agente usa uma interface de streaming para gerenciar o áudio I/O em tempo real.
   - Se for implantada em um contexto de contact center ou de telecomunicações, a integração telefônica gerencia o roteamento de sessões, a entrada multifrequência de dois tons (DTMF) e o transporte de mídia.

Nota: DTMF se refere aos tons gerados quando você pressiona os botões no teclado do telefone. No contexto da integração do contexto de streaming e telefonia em agentes de voz, o DTMF é usado como um mecanismo de entrada de sinal durante uma chamada telefônica, especialmente em sistemas de resposta de voz interativa (IVR). As entradas DTMF permitem que o agente:

- Reconheça as seleções do menu (por exemplo, “Pressione 1 para faturar”. Pressione 2 para obter suporte.”)
- Colete entradas numéricas (por exemplo, números de conta, PINs e números de confirmação)
- Acione fluxos de trabalho ou transições de estado em fluxos de chamadas
- Reverta da fala para o tom de toque quando necessário

1. Razões por meio do contexto de fluxo do LLM
   - A consulta é enviada ao agente, que a transmite, junto com qualquer metadado da sessão (por exemplo, ID do chamador, contexto anterior), para um LLM.
   - O LLM gera uma resposta, possivelmente usando uma estratégia de cadeia de pensamento ou memória de múltiplas voltas se a interação for contínua.
2. Retorna uma resposta de voz
   - O agente converte sua resposta em fala usando a conversão de texto em fala (TTS).
   - Ele retorna o áudio para o usuário por meio de um canal de voz.

Capacidades

- Real-time compreensão e geração de fala
- Multilíngue I/O com suporte a STT e TTS
- Integração com APIs de telefonia ou streaming
- Reconhecimento da sessão e transferência de memória entre turnos

Casos de uso comuns

- Sistemas IVR conversacionais
- Recepcionistas virtuais e agendadores de consultas
- Voice-driven agentes de helpdesk
- Assistentes de voz vestíveis
- Interfaces de voz para casas inteligentes e ferramentas de acessibilidade

Orientação para implementação

Você pode criar esse padrão usando as seguintes ferramentas e Serviços da AWS:

- Amazon Lex V2 ou Amazon Transcribe para STT
- Amazon Polly para TTS
- Amazon Chime SDK, Amazon Connect Customer ou Amazon Interactive Video Service (Amazon IVS) para streaming e telefonia
- Amazon Bedrock para raciocinar com Anthropic, AI21 ou outros modelos de fundação
- AWS Lambda para conectar STT, LLM, TTS e contexto de sessão

(Opcional) Aprimoramentos adicionais podem incluir o seguinte:

- Amazon Kendra OpenSearch ou para RAG com reconhecimento de contexto
- Amazon DynamoDB para memória de sessão
- Amazon CloudWatch Logs e AWS X-Ray para rastreabilidade

Resumo

Os agentes de fala e voz são sistemas inteligentes que interagem por meio de conversas naturais. Ao integrar interfaces de fala com o raciocínio LLM e a infraestrutura de streaming em tempo real, os agentes de voz permitem interações perfeitas, acessíveis e escaláveis.

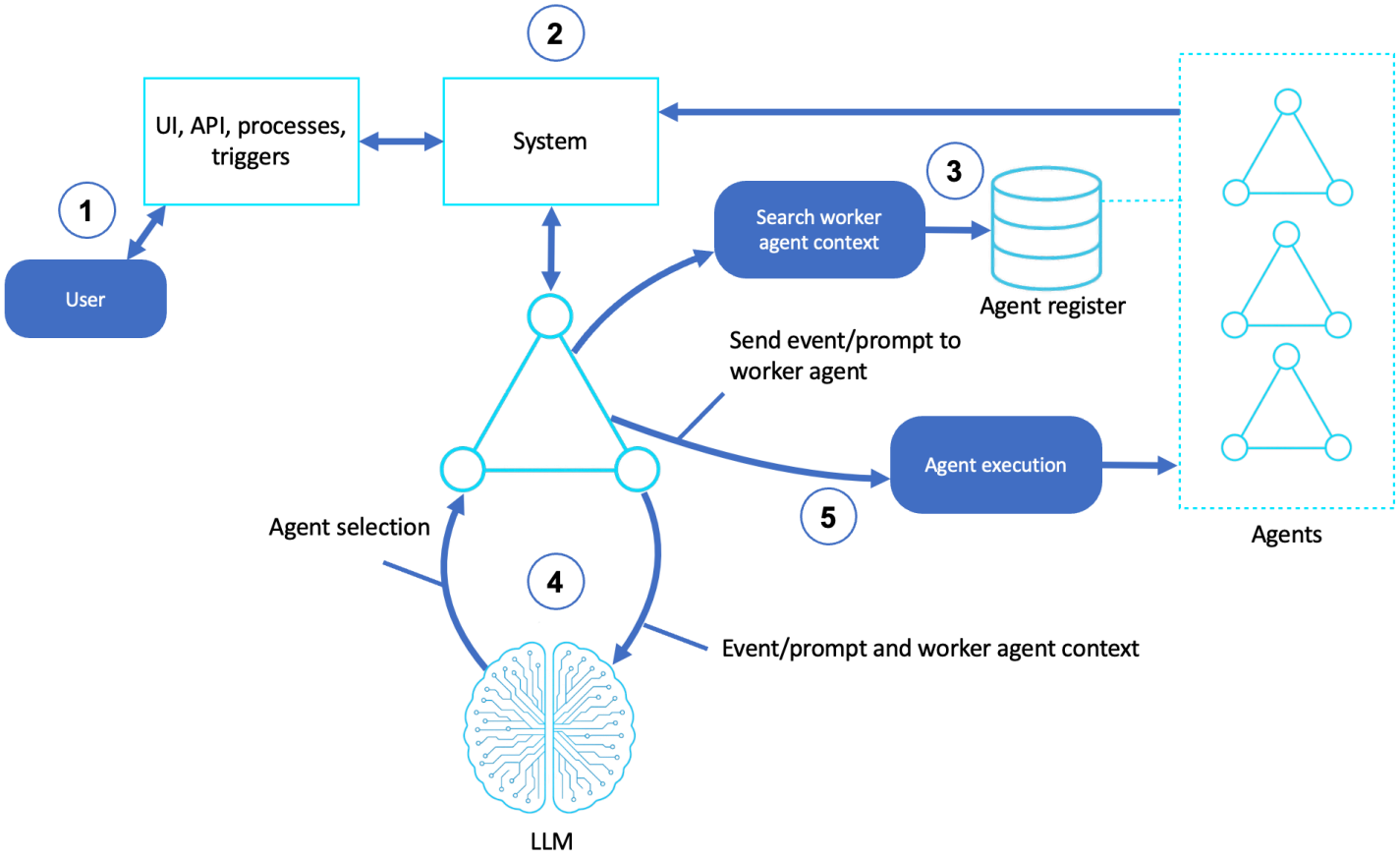
Agentes de orquestração de fluxo de trabalho

Agentes de orquestração de fluxo de trabalho gerenciam e coordenam tarefas, processos e serviços de várias etapas em sistemas distribuídos. Em vez de raciocinar e agir isoladamente, esses agentes delegam trabalho a subagentes ou outros sistemas, mantêm o contexto de execução e se adaptam com base em resultados intermediários.

Esses agentes são uma parte fundamental dos fluxos de automação. Eles são particularmente úteis ao lidar com tarefas de longa execução, composições de vários agentes e integrações entre domínios em que vários agentes e ferramentas devem ser chamados em sequência ou condicionalmente.

Arquitetura

Um agente de orquestração de fluxo de trabalho é mostrado no diagrama a seguir:



Description

1. Recebe a entrada do usuário
   - Um usuário (ou acionador externo) inicia uma tarefa por meio de uma interface de usuário, API ou evento do sistema.
2. Lida com eventos do sistema
   - Um componente do sistema recebe a solicitação e emite um evento ou comando que requer orquestração.
3. Recupera o contexto
   - O agente do fluxo de trabalho consulta as bases de conhecimento e os registros de agentes para encontrar o agente de trabalho certo para a tarefa com base nos metadados, no domínio e na taxa de sucesso anterior.
4. Seleciona um agente LLM

- Um LLM ajuda a selecionar o melhor agente ou plano de fluxo de trabalho analisando a descrição da tarefa e as opções disponíveis.
- Também pode formular solicitações específicas de tarefas para enviar a um agente selecionado.

5. Delega e executa
   - O agente de trabalho escolhido recebe o evento ou o prompt e começa a executar os comandos.
   - Ele pode rastrear o estado de execução, tentar novamente em caso de falha e passar resultados intermediários para o próximo agente na sequência.

Capacidades

- Composição do agente (por exemplo, supervisores, agentes colaboradores e ferramentas)
- Event-driven ou execução programada
- Rastreamento de memória e estado ao longo do tempo
- Orquestração hierárquica ou paralela de tarefas (síncrona em comparação com fluxos de trabalho assíncronos)
- Seleção e encadeamento dinâmicos de agentes

Casos de uso comuns

- Automação em várias etapas (por exemplo, ingestão de dados e geração de relatórios)
- Roteamento e escalonamento do atendimento ao cliente (por exemplo, agente como coordenador)
- Agentes de IA coordenam com humanos e bots dentro do mesmo circuito
- Automatiza os processos corporativos usando LLM-powered a lógica
- Os sistemas híbridos combinam agentes de IA e ferramentas tradicionais de orquestração

Orientação para implementação

Você pode criar esse padrão usando as seguintes ferramentas e Serviços da AWS:

- Amazon Bedrock para raciocínio e seleção de agentes
- AWS Step Functions ou Amazon EventBridge para composição do fluxo de trabalho

- AWS Lambda como unidades de execução ou executores de tarefas
- Amazon DynamoDB, Amazon Simple Storage Service (Amazon S3) ou Amazon RDS para monitorar estados e resultados
- AWS AppFabric ou Amazon AppFlow para coordenação entre sistemas
- (Opcional) Use o agente de SageMaker execução da Amazon para hospedar agentes de trabalho específicos do domínio

Resumo

Agentes de fluxo de trabalho coordenam, adaptam e alinham metas em ambientes multiagentes. Isso significa que os agentes de IA podem colaborar, se adaptar às condições de tempo de execução e fornecer resultados complexos por meio de fluxos de trabalho modulares e explicáveis.

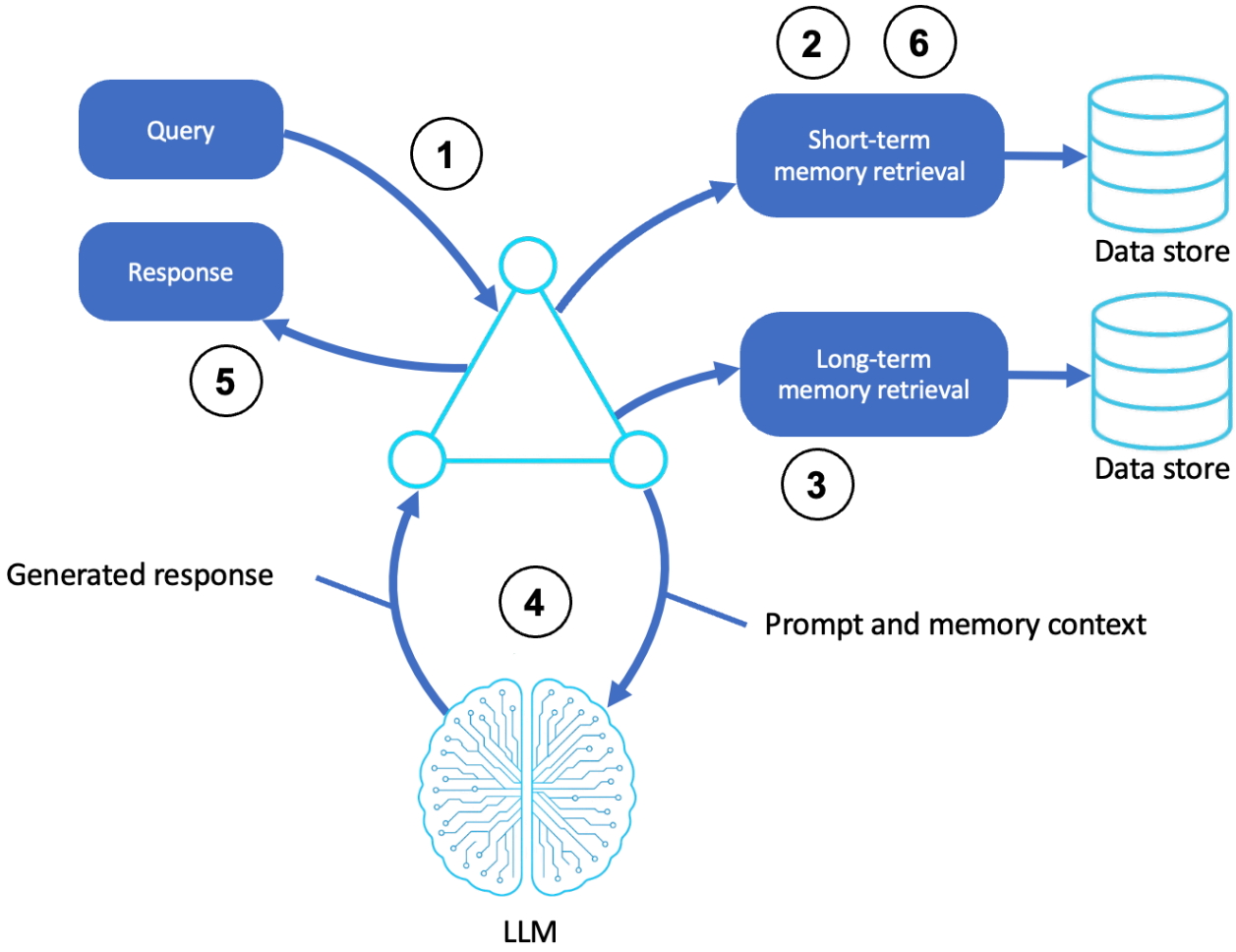
Memory-augmented agentes

Memory-augmented os agentes são aprimorados com a capacidade de armazenar, recuperar e raciocinar usando memória de curto e longo prazo. Isso permite que eles mantenham o contexto em várias tarefas, sessões e interações, o que produz respostas mais coerentes, personalizadas e estratégicas.

Diferentemente dos agentes apátridas, os agentes com memória aumentada se adaptam referenciando dados históricos, aprendem com resultados anteriores e tomam decisões alinhadas às metas, preferências e ambiente do usuário.

Arquitetura

Um agente com memória aumentada é mostrado no diagrama a seguir:



Description

1. Recebe informações ou eventos
   - O agente recebe uma consulta do usuário ou um evento do sistema. Isso pode ser um texto, um gatilho de API ou uma mudança ambiental.
2. Recupera a memória de curto prazo
   - O agente recupera o histórico recente da conversa, o contexto da tarefa ou o estado do sistema que é relevante para a sessão ou o fluxo de trabalho.
3. Recupera a memória de longo prazo
   - O agente consulta a memória de longo prazo (por exemplo, bancos de dados vetoriais e armazenamentos de valores-chave) para obter informações históricas, como as seguintes:
     - Preferências do usuário
     - Decisões e resultados anteriores

    - Conceitos, resumos ou experiências aprendidas

4. Razões por meio do LLM
    - O contexto da memória é incorporado ao prompt do LLM, permitindo que o agente raciocine com base nas entradas atuais e no conhecimento prévio.
5. Gera saídas
    - O agente produz uma resposta, um plano ou uma ação contextualmente consciente que é personalizada de acordo com o histórico da tarefa e as entradas do usuário.
6. Atualiza a memória
    - Novas informações, como metas atualizadas, sinais de sucesso e fracasso e respostas estruturadas, são armazenadas para tarefas futuras.

Capacidades

- Continuidade da sessão em conversas ou eventos
- Persistência de metas ao longo do tempo
- Consciência contextual baseada em um estado em evolução
- Adaptabilidade informada por sucessos e fracassos anteriores
- Personalização alinhada às preferências e ao histórico do usuário

Casos de uso comuns

- Co-pilotos conversacionais que lembram as preferências do usuário
- Agentes de codificação que rastreiam alterações na base de código
- Agentes de fluxo de trabalho que se adaptam de acordo com o histórico de tarefas
- Gêmeos digitais que evoluem a partir do conhecimento do sistema
- Agentes de pesquisa que evitam recuperações redundantes

Implementação de agentes com memória aumentada

Use as seguintes ferramentas e Serviços da AWS para agentes com memória aumentada:

Camada de memória	AWS service (Serviço da AWS)	Finalidade
Short-term	Contexto do Amazon DynamoDB, Redis, Amazon Bedrock	Recuperação rápida de estados de interação recentes
Long-term (estruturado)	Amazon Aurora, Amazon DynamoDB, Amazon Neptune	Fatos, relacionamentos e registros
Long-term (semântico)	OpenSearch, PostgreSQL, Pinha	Embedding-based recuperaç ão (ou seja, RAG)
Armazenamento	Amazon S3	Armazenamento de transcriç ões, memórias estruturadas e arquivos
Orquestração	AWS Lambda or AWS Step Functions	Gerenciando o ciclo de vida de injeção e atualização de memória
Reasoning	Amazon Bedrock	Claude antrópico ou Mistral com alertas de memória

Implementando a solicitação com injeção de memória

Para integrar a memória ao raciocínio do agente, use uma combinação de estado estruturado e injeção de contexto com recuperação aumentada:

- Inclua o estado mais recente do agente e o histórico recente de diálogos como entrada estruturada ao criar o prompt para o modelo de linguagem, para que ele possa raciocinar com todo o contexto.
- Use a geração aumentada de recuperação (RAG) para extrair documentos ou fatos relevantes da memória de longo prazo.
- Resuma os planos, o contexto e as interações anteriores para fins de compactação e relevância.
- Injete módulos de memória externa, como armazenamentos vetoriais ou registros estruturados, durante a inferência para orientar a tomada de decisões.

Resumo

Memory-augmented os agentes mantêm a continuidade do pensamento aprendendo com a experiência e lembrando o contexto do usuário. Esses agentes superam a inteligência reativa usando colaboração, personalização e raciocínio estratégico de longo prazo. Em termos de IA agente, a memória permite que os agentes se comportem mais como contrapartes digitais adaptáveis e menos como ferramentas sem estado.

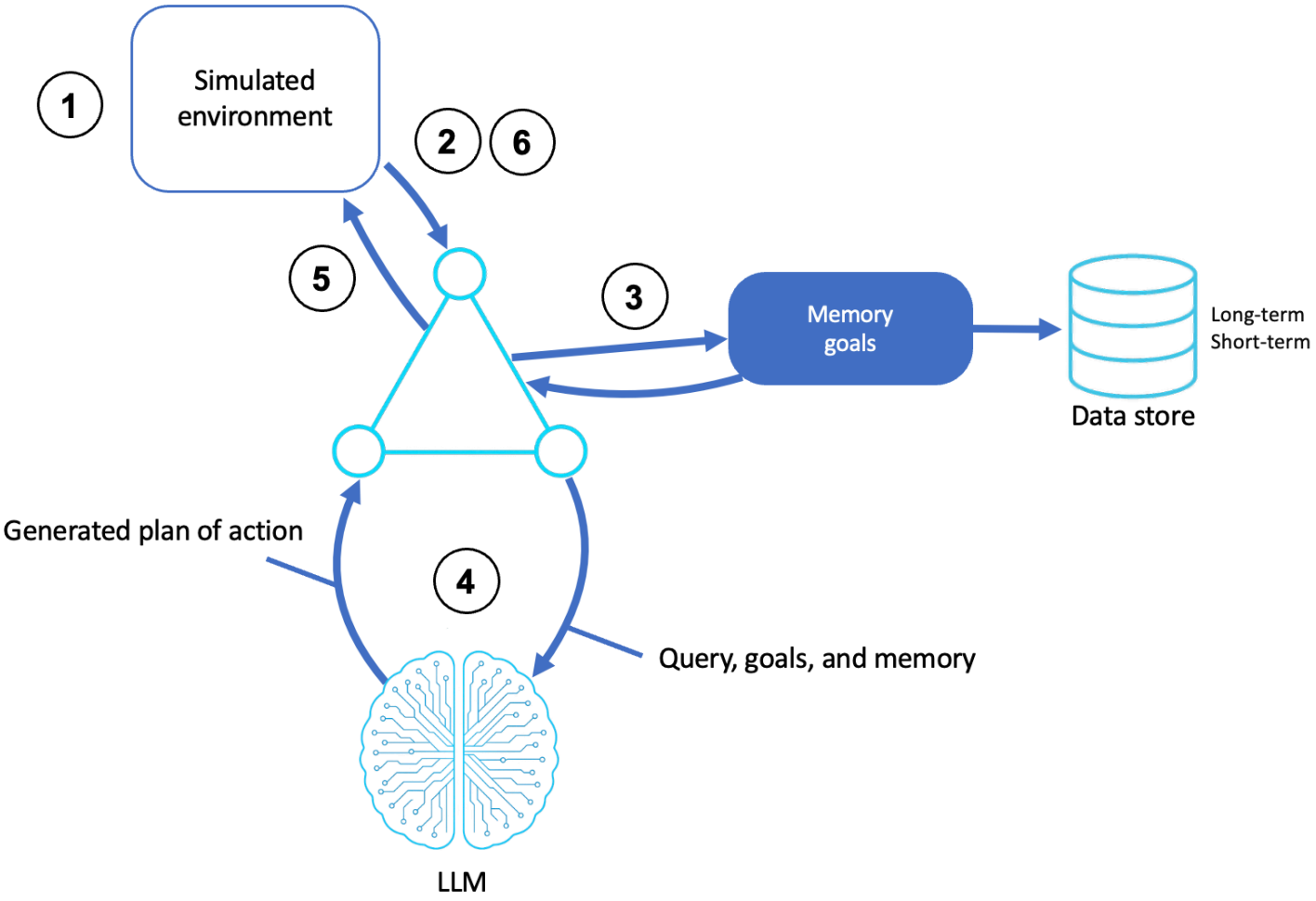
Agentes de simulação e de teste

Agentes de simulação e de teste operam em ambientes virtualizados ou controlados, onde raciocinam, agem e aprendem. Esses agentes simulam comportamento, modelam resultados e treinam estratégias em ambientes repetíveis antes de aplicá-las a ambientes do mundo real.

Esse padrão é útil para desenvolvimento iterativo, aprendizado por reforço (RL), avaliação autônoma de tomada de decisão e testes de comportamento emergentes. Os agentes de simulação geralmente operam em circuitos fechados, recebendo feedback de seu ambiente e ajustando seu comportamento de acordo, tornando-os essenciais para tarefas que envolvem raciocínio espacial, controle em tempo real ou dinâmica complexa do sistema.

Arquitetura

O diagrama a seguir mostra uma simulação ou um agente de teste:



Description

1. Inicia um ambiente
   - O agente inicia um ambiente simulado (por exemplo, um mundo 3D, mecanismo de física, sandbox de CLI ou fluxo de dados sintético).
   - O agente é carregado no ambiente com uma tarefa, meta ou política inicial.
2. Percebe o agente
   - O agente percebe o estado atual por meio da telemetria de simulação (por exemplo, emulação de sensor, câmera virtual e registros estruturados).
3. Recupera objetivo e memória
   - O agente recupera o objetivo atribuído, as instruções do cenário ou a meta contextual.
   - Ele também pode recuperar a memória anterior, incluindo o seguinte:
     - Long-term estratégias ou políticas

    - Mapas ambientais ou restrições conhecidas
    - Sucessos ou fracassos anteriores de simulações semelhantes
4. Razões e planos
    - Um LLM interpreta o estado simulado, os objetivos da tarefa e o conhecimento aprendido.
    - Ele gera um plano de ação ou comando de controle.
5. Executa ações simuladas
    - O agente executa o plano, modifica o estado, navega pelo espaço ou interage com entidades virtuais.
6. Aprende
    - O agente avalia os resultados da ação
    - Dependendo da configuração do agente, ele pode fazer o seguinte:
        - Execute RL
        - Registre os resultados para futuros ajustes
        - Adapte estratégias em tempo real

Capacidades

- Opera em ambientes sintéticos ou virtuais
- Oferece suporte ao aprendizado por tentativa e erro, ao refinamento de políticas e à modelagem do sistema
- Low-risk testes de comportamento, tratamento de falhas e casos extremos
- Permite a análise do comportamento de agentes emergentes em configurações de vários agentes
- Suporta controle de circuito fechado e exploração humana em circuito

Casos de uso comuns

- Aprendizado por reforço para robótica, drones e jogos
- Treinamento de veículos autônomos em estradas virtuais
- UIs ou CLIs simuladas para cenários DevOps de teste
- Experimentos de comportamento emergente em simulações sociais
- Validação de segurança da lógica de decisão antes da produção

Orientação para implementação

Você pode criar um agente de simulação e teste usando as seguintes ferramentas e: Serviços da AWS

Componente	AWS service (Serviço da AWS)	Finalidade
Environment	Amazon ECS, Amazon EC2 ou um simulador personalizado no SageMaker Amazon Studio Lab	Execute mundos virtuais (Gazebo, Unity, Unreal) ou CLIs de sandbox
Lógica do agente	Amazon Bedrock SageMaker, Amazon ou AWS Lambda	LLM-based planejadores ou agentes de RL
Ciclo de feedback	SageMaker Aprendizado por reforço da Amazon CloudWatch, Amazon ou registros personalizados	Rastreamento de recompensas, pontuação de resultados e registro de comportamento
Memória e repetição	Amazon S3, Amazon DynamoDB ou Amazon RDS	Estado persistente, histórico de episódios ou dados do cenário
Visualização	CloudWatch Painéis da Amazon ou cadernos da Amazon SageMaker	Observe as mudanças nas políticas, os resultados e as métricas de treinamento

A seguir estão os aplicativos adicionais:

- AWS SimSpace Weaverpara simulações espaciais em grande escala
- AWS IoT Corepara testar dispositivos de sombra
- Amazon SageMaker Experiments para avaliação e benchmarking de agentes

Resumo

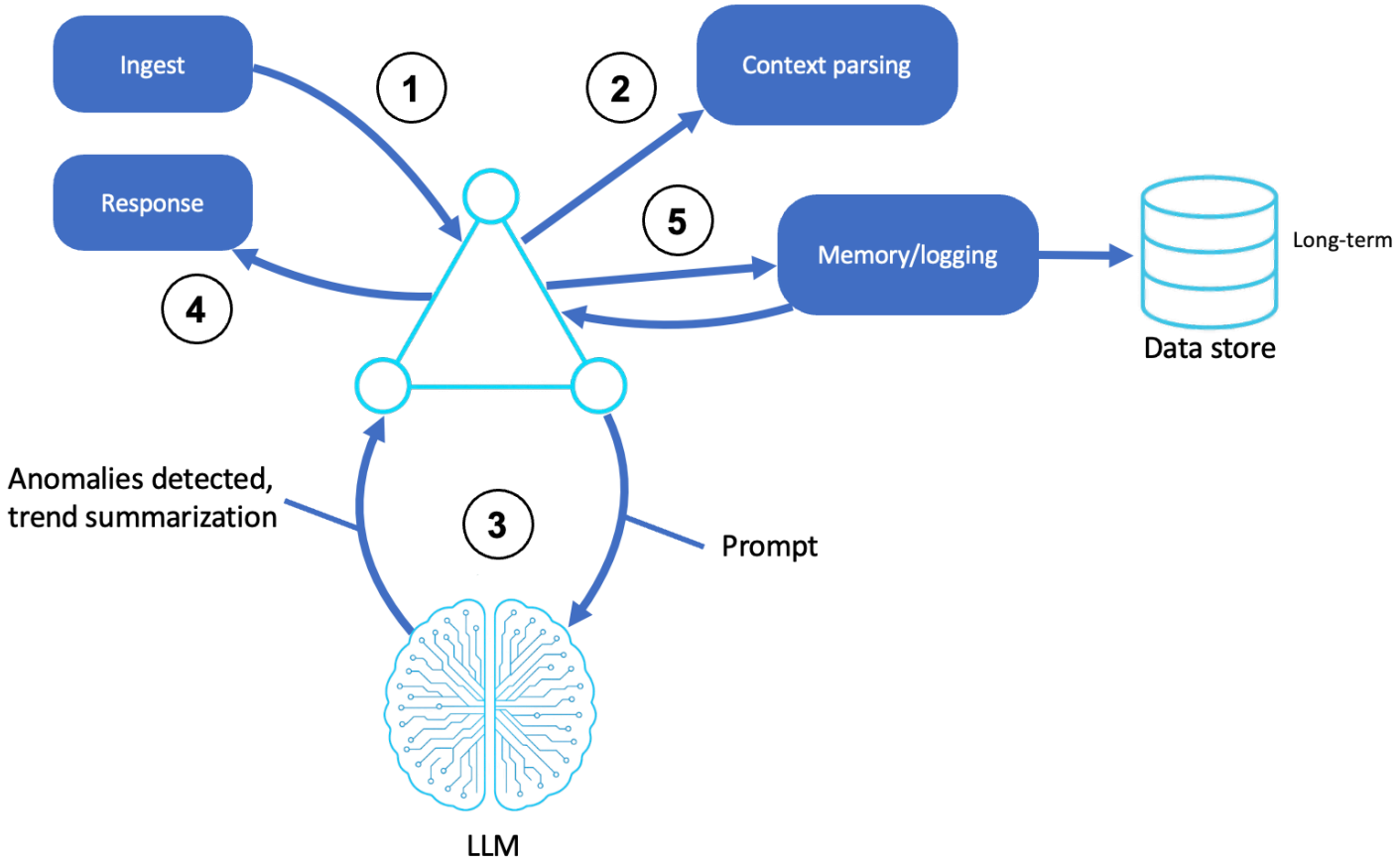
Os agentes de simulação e de teste são para exploração estruturada antes de serem implantados em sistemas de produção. Use esses agentes para treinar políticas de navegação autônoma, testar processos de negócios em ambientes sintéticos e avaliar padrões de coordenação em enxames.

Agentes observadores e de monitoramento

Agentes observadores e de monitoramento observam passivamente sistemas, ambientes e interações para detectar padrões, gerar insights e acionar ações. Como observadores inteligentes, eles aprimoram alertas, diagnósticos e auditorias sem iniciar diretamente o comportamento.

Esses agentes se destacam onde o monitoramento tradicional carece de adaptabilidade ou raciocínio, especialmente para AI-in-the-loop monitoramento, detecção de anomalias, supervisão de conformidade e inteligência de segurança. Agentes observadores são ouvintes de eventos que monitoram continuamente a telemetria do sistema e as interações do usuário. O agente depende da percepção, interpretação e escalonamento ou relatório condicional.

Arquitetura



Description

1. Telemetria de ingestão
   - O agente recebe informações de uma ou mais fontes do sistema, como as seguintes:
     - Registros (aplicativo, infraestrutura, segurança)
     - Métricas (desempenho, latência, uso)
     - Eventos (chamadas de API, ações do usuário, dados do sensor)
2. Analise o contexto
   - A entrada bruta é analisada, estruturada e enriquecida com metadados, como registro de data e hora, identidade do ator, estado do sistema e ID de rastreamento.
3. Razões para usar o LLM
   - O agente usa um LLM ou módulo lógico para interpretar entradas analisadas identificando anomalias, resumindo tendências e correlacionando traços distribuídos ou janelas de tempo.
4. Classifique ou alerte

- O agente determina se o comportamento observado justifica o seguinte:
  - Um alerta ou escalonamento
  - Uma atualização de relatório ou painel
  - Um gatilho de resposta (por exemplo, remediação automática e aplicação de políticas)

5. Memória de log ou loops de feedback
   - O agente armazena eventos e decisões para aprendizado de longo prazo, auditorias ou referência futura para outros agentes.

Capacidades

- Passivo e não invasivo (o agente não age diretamente)
- Altamente escalável e assíncrono
- AI-driven correlação entre sinais ruidosos ou distribuídos
- Oferece suporte a auditoria, conformidade e visão em tempo real
- Pode alimentar agentes posteriores ou fluxos de trabalho humanos

Casos de uso comuns

- AI-augmented observabilidade para microsserviços e APIs
- Monitoramento de desvios de modelos, violações de políticas ou comportamento fora da banda
- Análise da atividade do cliente ou resumos de interações
- Agentes de revisão de código que monitoram confirmações ou implantações
- Monitoramento de registros de segurança ou conformidade usando o raciocínio LLM

Orientação para implementação

Você pode criar um observador e um agente de monitoramento usando as seguintes ferramentas e Serviços da AWS:

Componente	AWS service (Serviço da AWS)	Finalidade

Ingestão de eventos	Amazon EventBridge, Amazon CloudWatch Logs, Amazon Kinesis, Amazon S3	Ingira telemetria estruturada e não estruturada
Pré-processamento	AWS Lambda, AWS Glue, AWS Step Functions	Transforme dados brutos em solicitações estruturadas
Motor de raciocínio	Amazon Bedrock, Amazônia SageMaker, AWS Lambda	Analise eventos, classifique comportamentos, gere insights
Armazenamento e memória	Amazon S3, Amazon DynamoDB, OpenSearch	Observações, resumos e resultados persistentes
Alertas e escalonamento	Amazon SNS, AWS AppFabric Amazon EventBridge	Acione sistemas ou agentes posteriores

A seguir estão os aplicativos adicionais:

- AWS Security Hub CSPMpara monitoramento de registros de segurança
- Amazon Quick para visualizar as saídas do agente

Resumo

Agentes observadores e de monitoramento rastreiam sistemas e comportamentos em tempo real. Eles detectam anomalias, auditam a segurança e coletam inteligência operacional identificando padrões que humanos ou regras possam ignorar. Esse recurso ajuda a criar sistemas que podem se adaptar às mudanças nas condições e tomar decisões com base em uma análise abrangente de dados.

Multi-agent colaboração

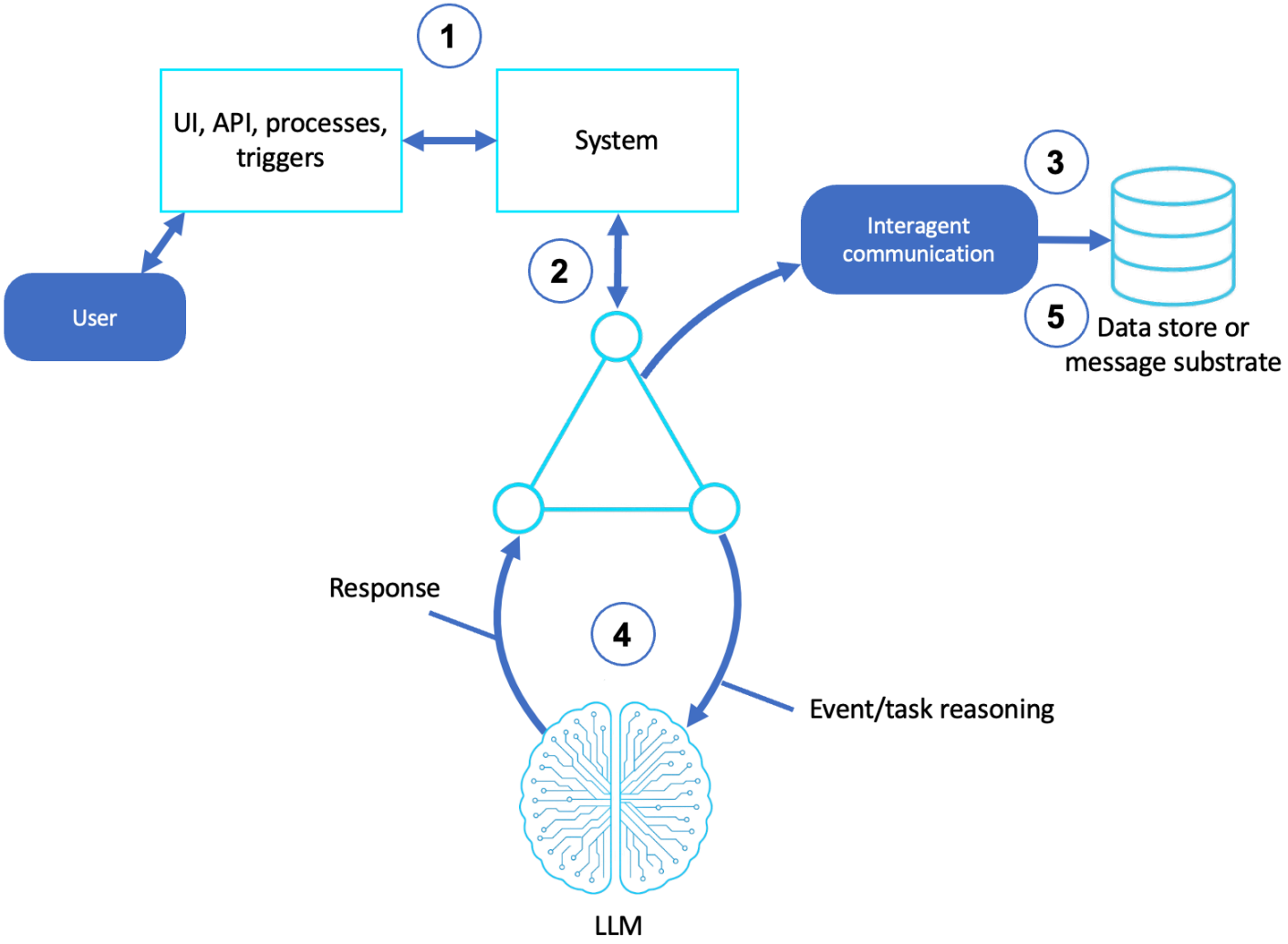
Multi-agent colaboração se refere a um padrão no qual vários agentes autônomos, cada um com uma função, especialização ou objetivo distinto, negociam para resolver tarefas complexas. Esses agentes podem operar de forma independente ou com outros agentes compartilhando informações, dividindo responsabilidades e raciocinando coletivamente em direção a uma meta.

Esse padrão difere dos agentes de fluxo de trabalho, que coordenam e delegam tarefas centralmente aos agentes subordinados em um fluxo estruturado. Em contraste, a colaboração multiagente enfatiza a coordenação ponto a ponto ou emergente, permitindo a adaptabilidade, o paralelismo e a divisão da cognição. A tabela a seguir compara a colaboração de vários agentes com agentes de fluxo de trabalho:

Recurso	agentes de fluxo de trabalho	Finalidade
Controle	Coordenador centralizado	Pares descentralizados, distribuídos ou baseados em funções
Interação	Um agente delega e monitora a execução	Vários agentes negociam, compartilham e se adaptam
Projeto	Sequência predefinida de tarefas	Distribuição de tarefas emergente e flexível
Coordenação	Orquestração processual	Interações cooperativas ou competitivas
Casos de uso	Automação de processos corporativos	Raciocínio complexo, exploração e estratégias emergentes

Arquitetura

O diagrama a seguir mostra a colaboração entre vários agentes:



Description

1. Inicia uma tarefa
   - Um usuário ou sistema emite uma meta ou problema de alto nível.
   - Um agente “gerente” ou contexto iniciador define o objetivo.
2. Atribui ou descobre funções
   - Os agentes se autoatribuem (lógica simbólica ou raciocínio) ou são delegados (mediadores de eventos) a outras funções, como planejador, pesquisador, executor, crítico ou explicador.
3. Comunica-se com outros agentes
   - Os agentes se comunicam por meio de memória compartilhada, filas de mensagens ou encadeamento de solicitações.
   - Eles podem debater, questionar ou propor subtarefas uns aos outros.

4. Usa raciocínio especializado
   - Cada agente usa seu próprio modelo ou lógica de domínio para resolver sua parte do problema.
   - Os agentes podem usar LLMs com prompts e memória específicos da função.
5. Coordena saídas ou metas
   - Os agentes sintetizam as contribuições em uma resposta final, plano ou ação.
   - (Opcional) Um agente supervisor pode validar ou resumir a saída sintetizada.

Capacidades

- Peer-level agentes com funções ou habilidades especializadas
- Comportamento emergente por meio de comunicação ou negociação
- Processamento paralelo de problemas complexos ou multifacetados
- Suporta deliberação, autocorreção e iteração reflexiva
- Modele a dinâmica social, a colaboração científica ou as funções da equipe empresarial

Casos de uso comuns

- Equipes de pesquisa autônomas (agente de busca, resumido e validador)
- Desenvolvimento de software (planejador, codificador e testador)
- Modelagem de cenários de negócios (finanças, políticas e conformidade)
- Negociação, licitação ou raciocínio multipartidário
- Tarefas multimodais (imagem, texto e lógica)

Orientação para implementação

Você pode criar um sistema multiagente usando as seguintes ferramentas e Serviços da AWS:

Componente	AWS service (Serviço da AWS)	Finalidade
Hospedagem de agentes	Amazon Bedrock, Amazônia SageMaker, AWS Lambda	Hospede LLM-driven agentes individuais

Camada de comunicação	Amazon SQS, Amazon, EventBridge AWS AppFabric	Mensagens e coordenação entre agentes
Memória compartilhada	Amazon DynamoDB, Amazon S3 ou OpenSearch	Multi-agent sistema de memória ou quadro-negro
Camada de orquestração	AWS Step Functions, AWS Lambda oleodutos	Lógica de início, tempo limite, retorno e repetição
Identificação do agente	Agentes do Amazon Bedrock (definidos por função) e API converse do AWS AppConfig Amazon Bedrock (agentes fora do Amazon Bedrock)	Role-based invocação de ferramenta ou agente e aplicação de limites
Interação emergente	EventBridge Pipelines da Amazon ou registros de agentes	Habilite o roteamento ou escalonamento dinâmico de tarefas

Resumo

Multi-agent a colaboração distribui tarefas de solução de problemas entre agentes modulares e orientados por funções. Diferentemente da orquestração do fluxo de trabalho, os padrões de colaboração usam inteligência, resiliência e escalabilidade emergentes que refletem a forma como os humanos resolvem problemas. É especialmente valioso para domínios abertos, tarefas criativas, raciocínio multimodal e ambientes que se beneficiam de diversas perspectivas.

Conclusão

Os padrões discutidos anteriormente ilustram abordagens fundamentais para implementações reais de IA agente. Do raciocínio básico à inteligência aumentada pela memória, cada padrão é configurado exclusivamente para percepção, cognição e ação com base na autonomia, assincronia e agência.

Esses padrões compartilham vocabulários e planos técnicos para criar sistemas inteligentes e direcionados a objetivos. Se um padrão está incorporado em uma interface de usuário, orquestrado por meio de serviços em nuvem ou coordenado entre equipes de agentes, cada padrão é adaptável e modular.

Takeaways

- Os padrões do agente podem ser compostos — a maioria dos agentes do mundo real combina dois ou mais padrões (por exemplo, um agente de voz com raciocínio e memória baseados em ferramentas).
- O design do agente é contextual — escolha padrões com base na superfície de interação, na complexidade da tarefa, na tolerância à latência e nas restrições específicas do domínio.
- AWSa implementação nativa é possível — Com o Amazon Bedrock, o Amazon SageMaker, AWS Lambda AWS Step Functions, e as arquiteturas orientadas a eventos, cada padrão de agente pode ser entregue em grande escala.

Fluxos de trabalho do LLM

Em padrões de agentes, exploramos os padrões comuns de agentes de IA, cada um construído em torno de um conjunto de recursos modulares: percepção, ação, aprendizado e cognição. No centro do módulo cognitivo em muitos padrões de agentes está um grande modelo de linguagem (LLM) capaz de raciocinar, planejar e tomar decisões. No entanto, invocar um LLM sozinho não é suficiente para produzir um comportamento inteligente e direcionado a objetivos.

Para realizar tarefas complexas de forma confiável, os agentes devem incorporar o LLM em um fluxo de trabalho estruturado, onde os recursos do modelo são aumentados com ferramentas, memória, ciclos de planejamento e lógica de coordenação. Esses fluxos de trabalho de LLM permitem que um agente defina metas, encaminhe subtarefas, chame serviços externos, reflita sobre os resultados e se coordene com outros agentes.

Este capítulo apresenta os principais padrões de design para criar módulos cognitivos robustos, extensíveis e inteligentes orientados por LLM, organizados em torno de fluxos de trabalho reutilizáveis.

Nesta seção

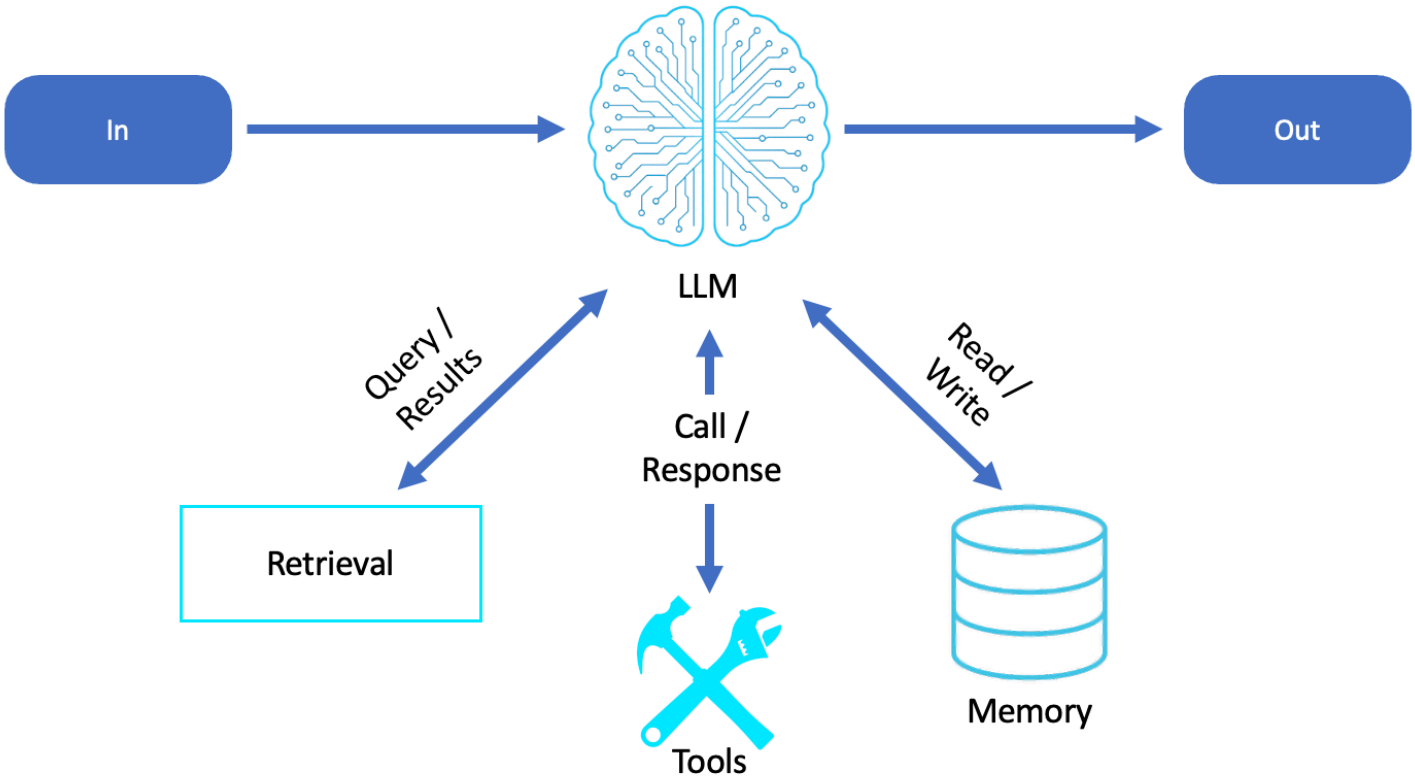
Visão geral da cognição aumentada por LLM

Em sua essência, o módulo cognitivo de um agente de software pode ser visto como um LLM envolto em aprimoramentos. O agente pode usar os seguintes elementos básicos para raciocinar de forma eficaz em seu ambiente:

- Solicitação — enquadrando a entrada usando contexto, instruções, exemplos e memória

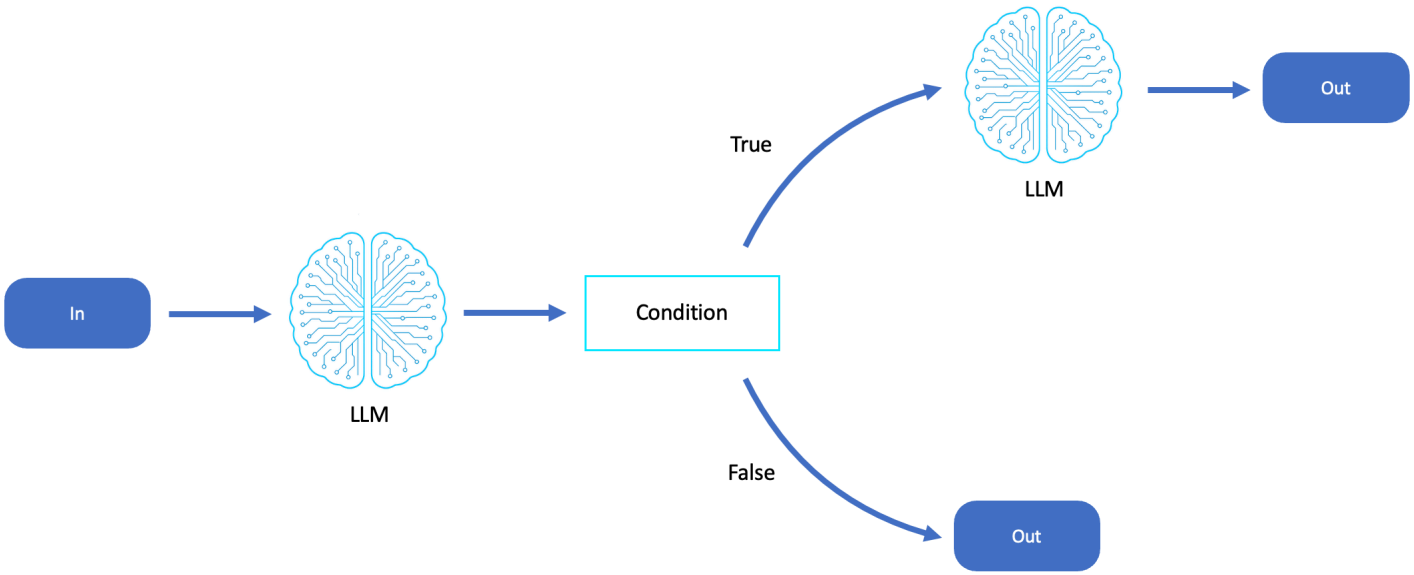
- Recuperação — Fornecimento up-to-date de conhecimento específico de domínio para o prompt do LLM por meio de pesquisa vetorial ou memória semântica, por exemplo, por meio de geração aumentada de recuperação (RAG)
- Uso da ferramenta — Permitindo que o LLM invoque APIs ou chame funções para recuperar ou agir sobre as informações
- Memória — Incorporar um estado persistente ou baseado em sessão ao ciclo de raciocínio, usando bancos de dados estruturados ou resumos contextuais

Esses aumentos são compostos por fluxos de trabalho que definem como o LLM é usado ao longo do tempo e em todas as tarefas, transformando-o de um mecanismo sem estado em um agente de raciocínio dinâmico.



Fluxo de trabalho para encadeamento imediato

O encadeamento imediato decompõe tarefas complexas em uma sequência de etapas, em que cada etapa é uma invocação LLM discreta que processa ou se baseia na saída da anterior.



O fluxo de trabalho de encadeamento imediato é adequado para cenários em que as tarefas podem ser divididas logicamente em etapas de raciocínio sequencial e em que os resultados intermediários informam o próximo estágio. Ele se destaca em fluxos de trabalho que exigem pensamento estruturado, transformação progressiva ou análise em camadas, como revisão de documentos, geração de código, extração de conhecimento e refinamento de conteúdo.

Description

- A complexidade da tarefa excede a janela de contexto ou a profundidade de raciocínio de uma única chamada LLM.
- Os resultados de uma etapa (por exemplo, análise, resumo ou planejamento) se tornam entradas para uma decisão de acompanhamento ou fase de geração.
- Você precisa de transparência e controle em todos os estágios de raciocínio (por exemplo, resultados intermediários auditáveis).
- Você deseja conectar a lógica externa de validação, filtragem ou enriquecimento entre as etapas.
- É ideal para agentes que operam em ciclos de raciocínio no estilo pipeline, como agentes de pesquisa, assistentes editoriais, sistemas de planejamento e copilotos de vários estágios.

Capacidades

- Cadeias lineares ou ramificadas de chamadas LLM

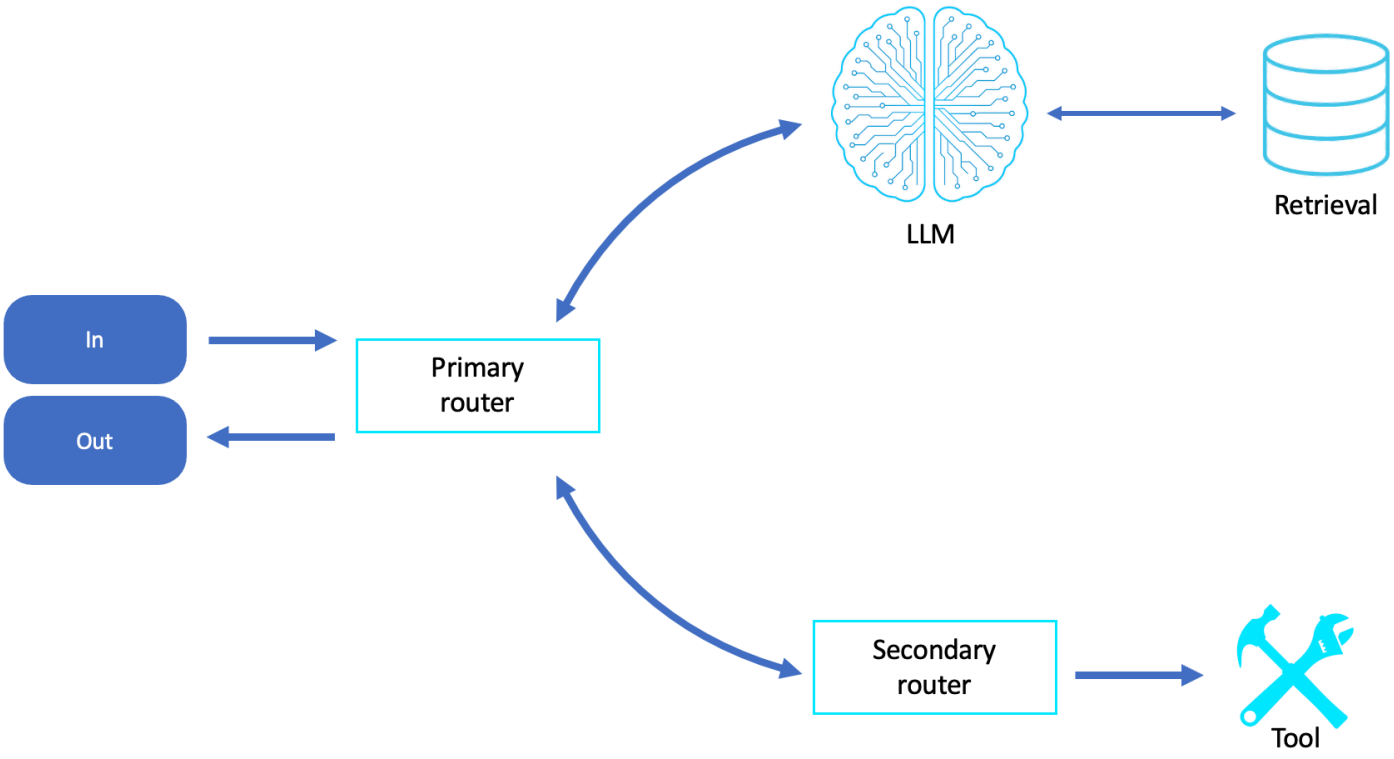
- Resultados intermediários passados como entrada estruturada ou incorporados em solicitações de acompanhamento
- Pode ser orquestrado com AWS Step Functions, ou com corredores AWS Lambda específicos do agente

Casos de uso comuns

- Tarefas de raciocínio em várias etapas (por exemplo, “resumir, reescrever uma crítica”)
- Assistentes de pesquisa sintetizando resultados em camadas (por exemplo, “pesquisar, extrair fatos, responder à pergunta”)
- Pipelines de geração de código (“gerar plano, escrever código de teste, explicar a saída”)

Fluxo de trabalho para roteamento

No padrão de roteamento, um classificador ou agente de roteador usa um LLM para interpretar a intenção ou a categoria de uma consulta e, em seguida, encaminha a entrada para uma tarefa ou agente downstream especializado.



O fluxo de trabalho de roteamento é usado em cenários em que um agente deve classificar rapidamente a intenção de entrada, o tipo de tarefa ou o domínio e, em seguida, delegar a solicitação a um subagente, ferramenta ou fluxo de trabalho especializado. É especialmente útil em agentes capacitados, como aqueles que atuam como assistentes gerais, portas de entrada para funções corporativas ou interfaces de IA voltadas para o usuário que abrangem domínios.

O roteamento é particularmente eficaz quando:

- Triagem de solicitações em uma variedade de tarefas (por exemplo, pesquisa, resumo, agendamento, cálculos).
- As entradas devem ser pré-processadas ou normalizadas antes de entrar em fluxos de trabalho mais especializados.
- Diferentes tipos de entrada (por exemplo, imagens versus texto, consultas estruturadas versus consultas não estruturadas) exigem tratamento personalizado.
- Um agente está atuando como uma central telefônica conversacional, delegando tarefas a agentes especializados ou microsserviços.
- Esse fluxo de trabalho é comum em copilotos específicos de domínio, bots de suporte ao cliente, roteadores de serviços corporativos e agentes multimodais, nos quais o despacho inteligente determina a qualidade e a eficiência do comportamento do agente.

Capacidades

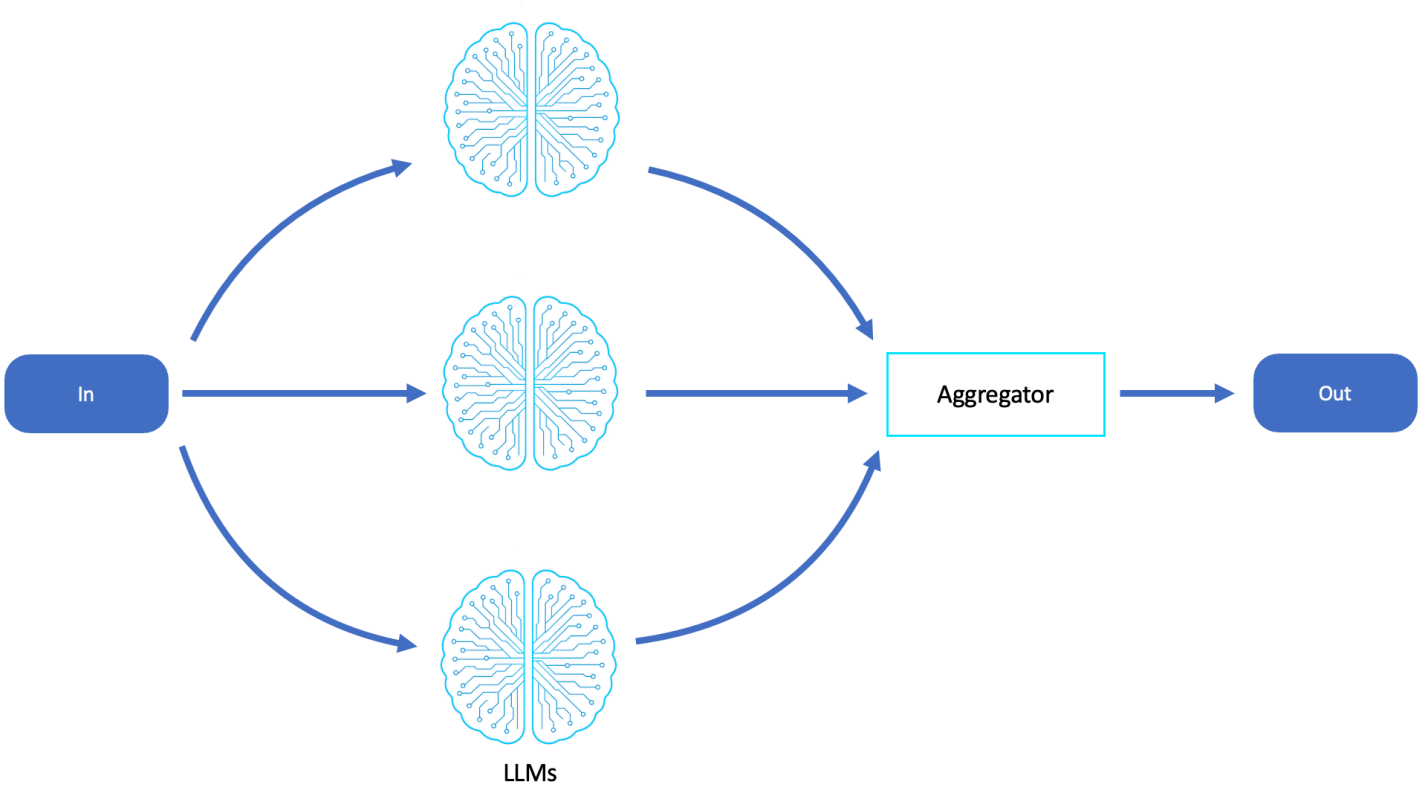
- Um LLM de primeira passagem atua como despachante
- As rotas podem invocar fluxos de trabalho distintos ou até mesmo outros padrões de agentes
- Oferece suporte à expansão modular dos recursos

Casos de uso comuns

- Assistentes de vários domínios (“essa é uma questão legal, médica ou financeira?”)
- Árvores de decisão aprimoradas com o raciocínio LLM
- Seleção dinâmica de ferramentas (por exemplo, pesquisa versus geração de código)

Fluxo de trabalho para paralelização

Esse fluxo de trabalho envolve dividir uma tarefa em subtarefas independentes que podem ser tratadas simultaneamente por várias chamadas ou agentes do LLM. As saídas são então agregadas programaticamente e sintetizadas em um resultado.



O fluxo de trabalho de paralelização é usado quando uma tarefa pode ser dividida em subtarefas independentes e não sequenciais que podem ser processadas simultaneamente, melhorando significativamente a eficiência, a produtividade e a escalabilidade. É especialmente poderoso em espaços problemáticos com muitos dados, orientados por lotes ou multiperspectivas, nos quais o agente deve analisar ou gerar conteúdo em várias entradas.

A paralelização é particularmente eficaz quando:

- As subtarefas não dependem dos resultados intermediários umas das outras, permitindo que elas sejam executadas paralelamente sem coordenação.
- Uma tarefa envolve repetir o mesmo processo de raciocínio em vários itens (por exemplo, resumir vários documentos ou avaliar uma lista de opções).
- Várias hipóteses ou perspectivas são exploradas paralelamente para promover diversidade, criatividade ou robustez.

- Você precisa reduzir a latência para solicitações de alto volume ou alta frequência por meio da execução simultânea do LLM.
- Esse fluxo de trabalho é comumente usado em agentes de processamento de documentos, mecanismos de pesquisa ou comparação, resumos de lotes, brainstormers multiagentes e tarefas escaláveis de classificação ou rotulagem, especialmente quando o raciocínio rápido e paralelo é uma vantagem de desempenho.

Capacidades

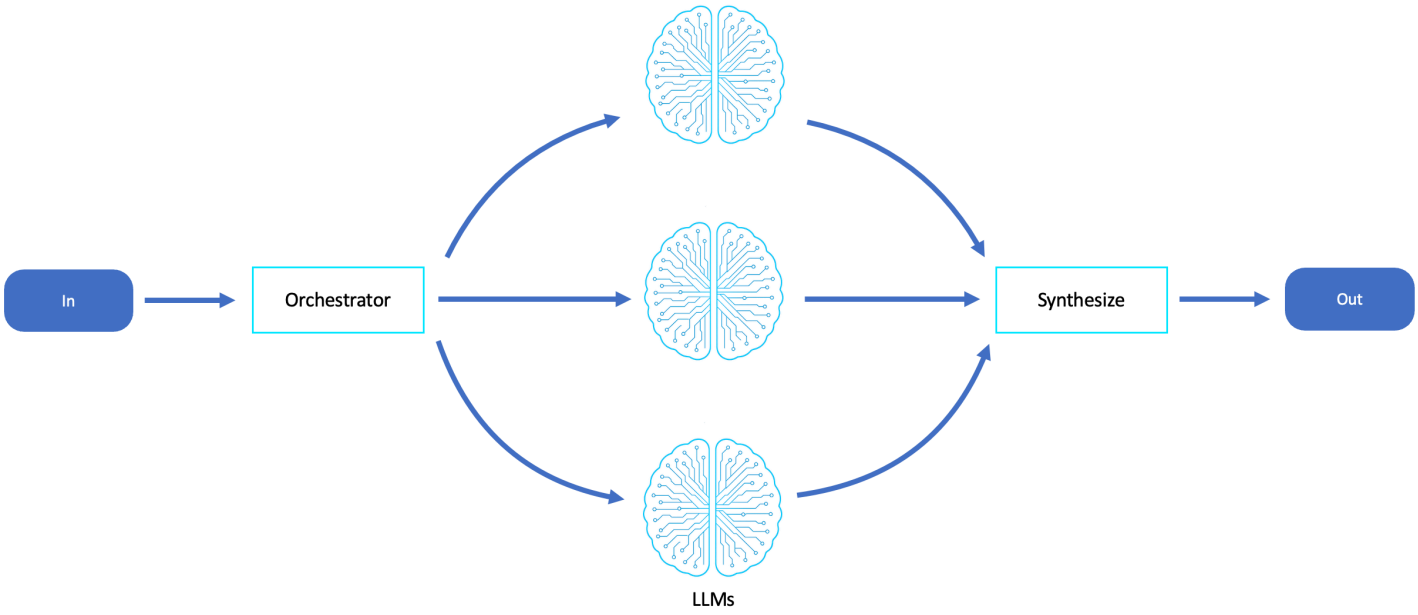
- Execução paralela de tarefas LLM (usando AWS Lambda, AWS Fargate, ou um estado do AWS Step Functions mapa)
- Requer alinhamento, validação ou desduplicação de resultados na fase de síntese
- Adequado para circuitos de agentes apátridas

Casos de uso comuns

- Analisando vários documentos ou perspectivas em paralelo
- Gerando diversos rascunhos, resumos ou planos
- Acelerando a produtividade em trabalhos em lotes

Fluxo de trabalho para orquestração

Um agente orquestrador central usa um LLM para planejar, decompor e delegar subtarefas a agentes ou modelos de trabalhadores especializados, cada um com uma função ou domínio específico. Isso reflete as estruturas da equipe humana e apoia o comportamento emergente em vários agentes.



O fluxo de trabalho de orquestração é ideal para cenários complexos, hierárquicos ou multidisciplinares, que exigem decomposição estruturada e execução especializada. É particularmente adequado para tarefas que exigem divisão de trabalho, em que diferentes subcomponentes de uma tarefa são mais bem administrados por agentes com capacidades, conhecimentos ou conjuntos de ferramentas distintos.

Esse fluxo de trabalho é particularmente eficaz quando:

- As tarefas podem ser divididas em subtarefas que variam em escopo, tipo ou raciocínio (por exemplo, planejar, pesquisar, implementar e testar).
- Um LLM ou meta-agente deve coordenar outros agentes, monitorar o progresso e sintetizar os resultados.
- Você deseja modularizar as responsabilidades do agente, permitindo escalabilidade, reutilização e ajuste especializado.
- O sistema exige um comportamento baseado em funções, imitando como equipes humanas (por exemplo, gerentes de projeto, desenvolvedores e revisores) operam em colaboração.

A orquestração é ideal para agentes de planejamento de vários turnos, copilotos de desenvolvimento de software, agentes de processos corporativos e executores de projetos autônomos. É especialmente útil ao implementar sistemas multiagentes que exigem divisão centralizada de tarefas, mas lógica de execução distribuída, permitindo extensibilidade e comportamento mais explicável em todas as camadas do agente.

Capacidades

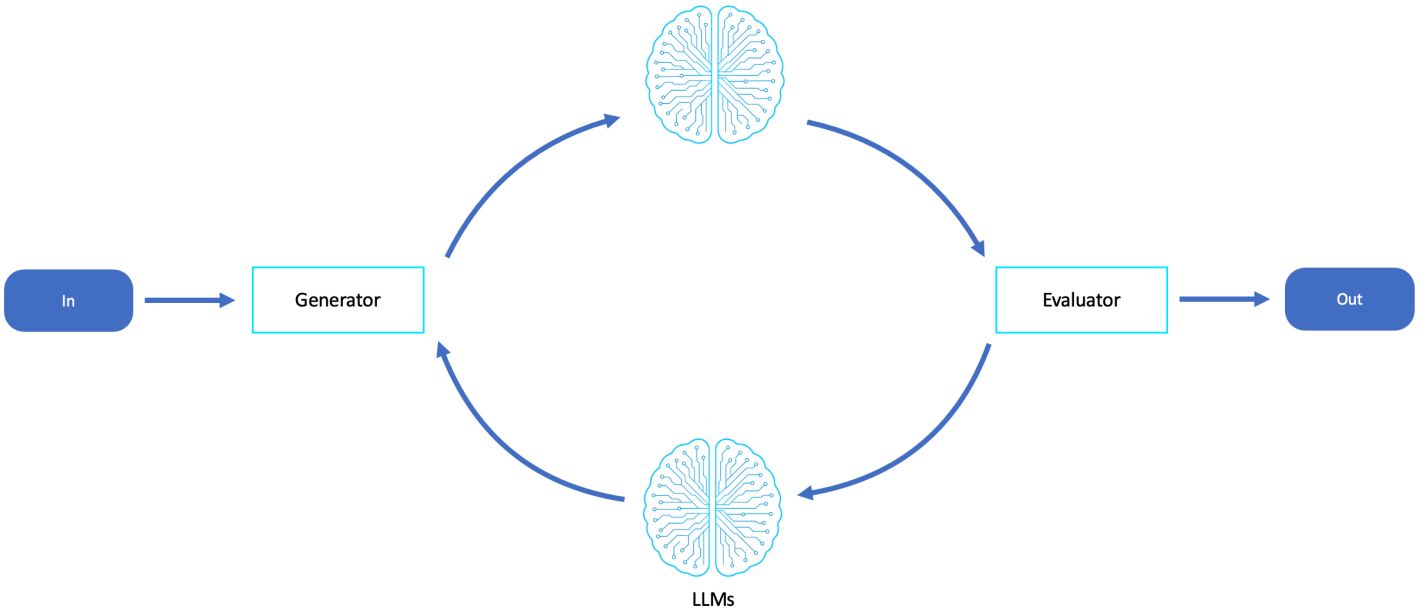
- O Orchestrator executa o metarraciocínio de metas
- Os agentes de trabalho podem incluir acesso à ferramenta, memória ou solicitação específica do domínio
- Pode ser hierárquico (ou seja, delegação de tarefas em vários níveis)

Casos de uso comuns

- Gerentes de projeto, pesquisadores coordenadores, escritores e agentes de garantia de qualidade
- Co-pilotos de codificação que combinam planejamento, execução e teste
- Agentes que supervisionam cadeias de ferramentas ou padrões de acesso à API

Fluxo de trabalho para avaliadores e ciclos de reflexão e refinamento

Esse fluxo de trabalho fornece um ciclo de feedback em que um LLM gera um resultado e outro avalia ou critica o resultado. Isso promove autorreflexão, otimização e melhorias iterativas.



O fluxo de trabalho do avaliador é ideal para cenários em que a qualidade, a precisão e o alinhamento da saída são importantes e em que a geração de passagem única não é confiável ou é

insuficiente. Esse fluxo de trabalho é excelente quando os agentes precisam fazer uma autocrítica, iterar e refinar seus resultados, seja para atender a um padrão mais alto de exatidão ou para explorar alternativas aprimoradas com base no feedback.

Esse fluxo de trabalho é particularmente eficaz quando:

- A saída envolve métricas de qualidade subjetivas (por exemplo, estilo, tom e legibilidade) ou critérios objetivos (por exemplo, exatidão, segurança e desempenho).
- O agente deve raciocinar por meio de compensações, avaliar restrições ou otimizar em direção a uma meta.
- Você precisa de redundância e garantia de qualidade integradas, especialmente em domínios regulamentados, voltados para o cliente ou criativos.
- Human-in-the-loop a revisão é cara ou não está disponível, e a validação autônoma é desejada.

Esse fluxo de trabalho é usado para geração de conteúdo, síntese e revisão de código, aplicação de políticas, verificação de alinhamento, ajuste de instruções e pós-processamento de RAG. Também é útil para agentes de autoaperfeiçoamento, onde o feedback contínuo ajuda a moldar respostas melhores ao longo do tempo para criar ciclos de decisão autônomos e confiáveis.

Casos de uso comuns

- Agentes da equipe vermelha em comparação com agentes da equipe azul
- Agentes que geram, avaliam e revisam códigos ou planos
- Garantia de qualidade, detecção de alucinações e aplicação de estilo

Capacidades

- Suporta geração e avaliação dissociadas usando modelos diferentes (por exemplo, Claude para geração e Mistral para avaliação)
- O feedback é estruturado e usado para gerar resultados revisados
- Suporta várias iterações ou limites de convergência

Conclusão

LLMs fornecem o núcleo cognitivo dos agentes de software modernos, mas a invocação bruta do modelo não é suficiente para obter inteligência objetiva, robusta e controlável. Para passar da geração de resultados para o raciocínio estruturado e o comportamento alinhado a metas, LLMs deve ser incorporado em padrões de fluxo de trabalho intencionais que definam como os modelos processam entradas, gerenciam contextos e coordenam ações.

Os fluxos de trabalho do LLM introduzem os fundamentos para criar o módulo cognitivo de um agente:

- O encadeamento imediato divide o raciocínio complexo em etapas modulares e auditáveis.
- O roteamento permite a classificação inteligente de tarefas e a delegação direcionada.
- A paralelização acelera a produtividade e promove um raciocínio diverso.
- A orquestração de agentes estrutura a colaboração entre vários agentes por meio da decomposição de tarefas e execução baseada em funções.
- O avaliador (loop reflect-refine) permite o autoaperfeiçoamento, o controle de qualidade e a verificação do alinhamento.

Cada fluxo de trabalho representa um padrão composto que pode ser adaptado às necessidades do agente, à complexidade da tarefa e às expectativas do usuário. Esses fluxos de trabalho não são mutuamente exclusivos. Eles são blocos de construção que geralmente são combinados em arquiteturas híbridas que oferecem suporte ao raciocínio dinâmico, à coordenação de vários agentes e à confiabilidade de nível corporativo.

À medida que você faz a transição para o próximo capítulo sobre padrões de fluxo de trabalho agentes, esses fluxos de trabalho de LLM reaparecerão como estruturas incorporadas em sistemas maiores, dando suporte à delegação de metas, orquestração de ferramentas, ciclos de decisão e autonomia do ciclo de vida. Dominar esses fluxos de trabalho de LLM é essencial para projetar agentes de software que não apenas prevejam texto, mas raciocinem, adaptem e ajam com propósito.

Padrões de fluxo de trabalho agentes

Os padrões de fluxo de trabalho agênticos integram agentes de software modulares com fluxos de trabalho estruturados de modelo de linguagem grande (LLM), permitindo raciocínio e ação autônomos. Embora inspirados nas arquiteturas tradicionais sem servidor e orientadas por eventos, esses padrões mudam a lógica central do código estático para agentes aumentados pelo LLM, fornecendo maior adaptabilidade e tomada de decisão contextual. Essa evolução transforma as arquiteturas de nuvem convencionais de sistemas determinísticos em sistemas capazes de interpretação dinâmica e aumento inteligente, mantendo os princípios fundamentais de escalabilidade e capacidade de resposta.

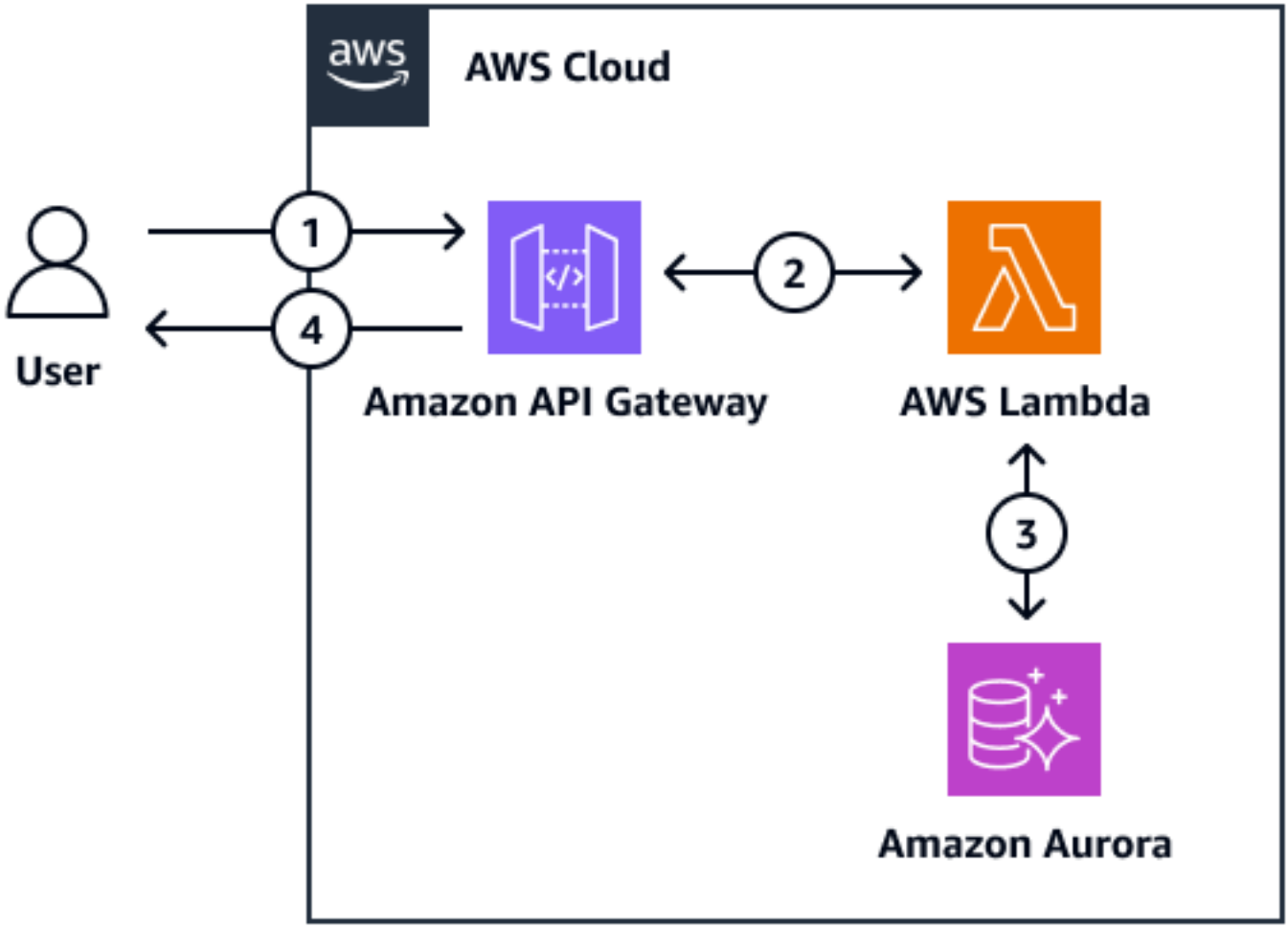
Nesta seção

De sistemas orientados a eventos a sistemas aumentados por cognição

As arquiteturas de nuvem modernas, especialmente aquelas baseadas em princípios sem servidor e orientadas por eventos, tradicionalmente se baseiam em padrões como roteamento, distribuição e enriquecimento para criar sistemas responsivos e escaláveis. Os sistemas de IA da Agentic se baseiam nessas bases, ao mesmo tempo que as reformulam em torno do raciocínio aumentado e da flexibilidade cognitiva do LLM. Essa abordagem permite recursos mais sofisticados de resolução de problemas e automação, potencialmente revolucionando a forma como tarefas complexas são tratadas em ambientes de nuvem.

Arquitetura orientada a eventos

O diagrama a seguir mostra um sistema distribuído típico:

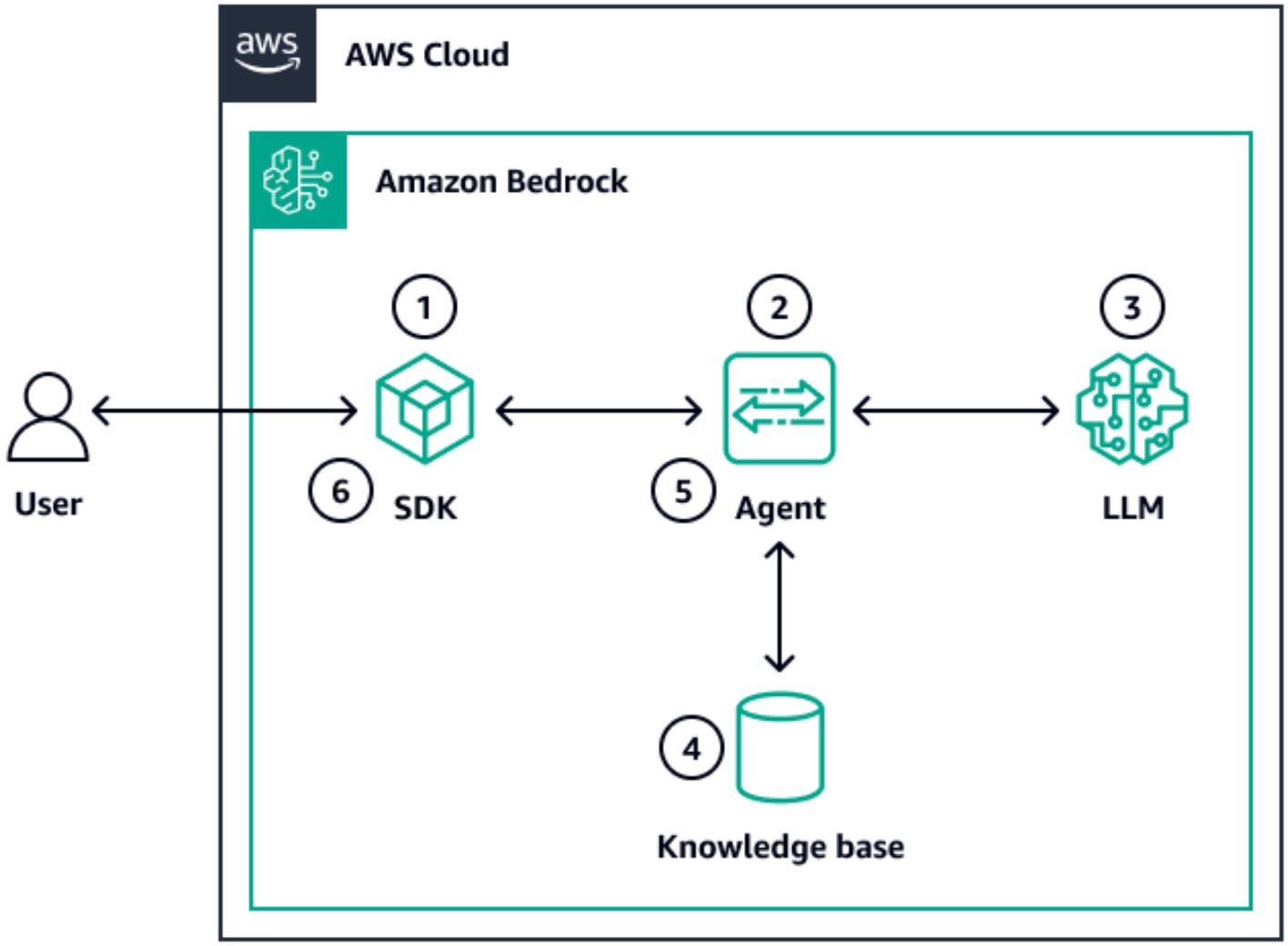


1. Um usuário envia uma solicitação para o Amazon API Gateway.
2. O Amazon API Gateway encaminha a solicitação para uma AWS Lambda função.
3. AWS Lambda realiza o enriquecimento de dados consultando um banco de dados Amazon Aurora
4. O Amazon API Gateway retorna a carga útil enriquecida para o chamador.

Essa estrutura é confiável e escalável, mas é fundamentalmente estática. As regras de negócios e os caminhos lógicos devem ser codificados explicitamente, e a adaptação a contextos em mudança ou a informações incompletas é limitada.

Fluxos de trabalho aprimorados com cognição

As arquiteturas agênticas adicionam aumento cognitivo a um sistema orientado por eventos. O diagrama a seguir mostra um equivalente agente:



1. Um usuário envia uma consulta por meio de uma chamada de SDK ou API.
2. Um agente do Amazon Bedrock recebe a consulta.
3. O agente interpreta a consulta invocando um LLM
4. O agente realiza o enriquecimento semântico pesquisando na base de conhecimento do Amazon Bedrock ou em outras fontes de dados externas.
5. O LLM sintetiza uma resposta rica em contexto e alinhada a objetivos.
6. O sistema retorna uma resposta sintetizada para o usuário.

Nesse fluxo, o LLM usa a lógica, entende a intenção, recupera e combina o contexto relevante e, em seguida, decide a melhor forma de responder. Esse padrão reflete o padrão de enriquecimento tradicional, em que as mensagens são aumentadas com dados externos antes de serem roteadas posteriormente. Em sistemas agênticos, no entanto, esse enriquecimento não é uma pesquisa estática. Em vez disso, o enriquecimento é dinâmico, guiado semanticamente e orientado por um propósito.

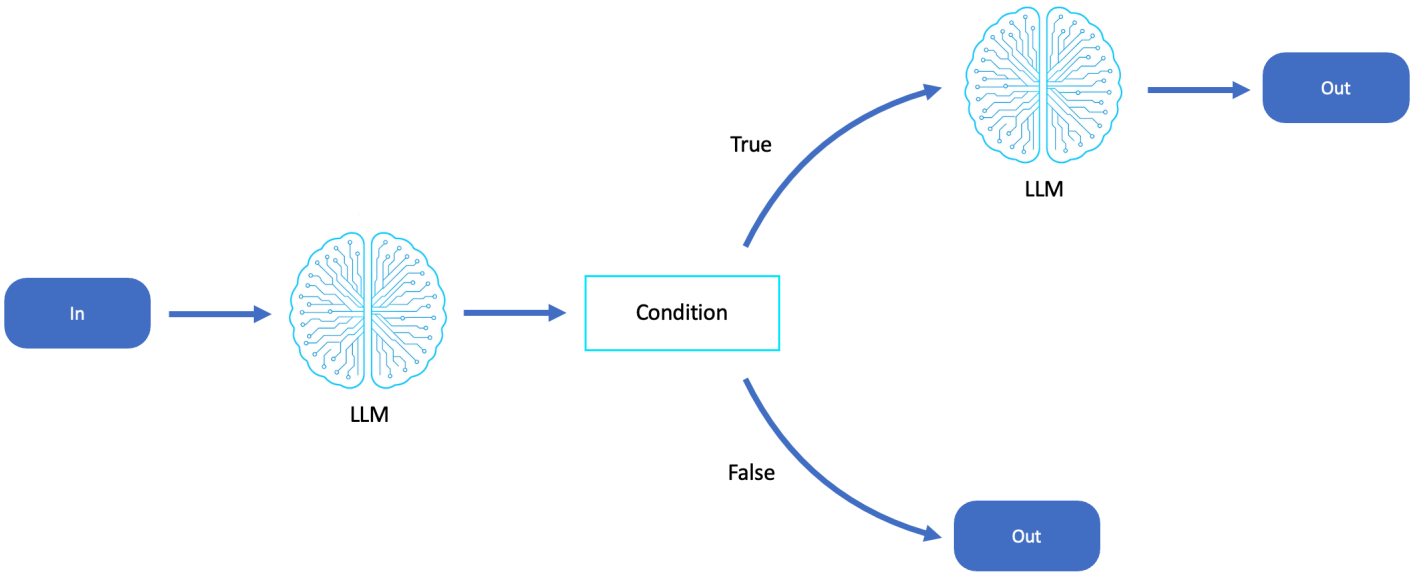
Insights principais

Cada fluxo de trabalho do LLM pode ser mapeado para um padrão de fluxo de trabalho agente, que reflete e desenvolve os estilos tradicionais de arquitetura orientada a eventos. Um alicerce básico dos fluxos de trabalho agentes é a capacidade de ampliar o contexto de um LLM com dados, ferramentas e memória. Isso cria um ciclo de raciocínio informado, adaptável e alinhado com a intenção do usuário. Enquanto os sistemas tradicionais enriquecem as mensagens com dados de pesquisa, os sistemas agentes permitem que o software atue menos como scripts e mais como colaboradores inteligentes.

Padrões de saga de encadeamento imediato

Ao reimaginar o encadeamento imediato do LLM como uma saga orientada por eventos, desbloqueamos um novo modelo operacional: os fluxos de trabalho se tornam distribuídos, recuperáveis e coordenados semanticamente entre agentes autônomos. Cada etapa de resposta rápida é reformulada como uma tarefa atômica, emitida como um evento, consumida por um agente dedicado e enriquecida com metadados contextuais.

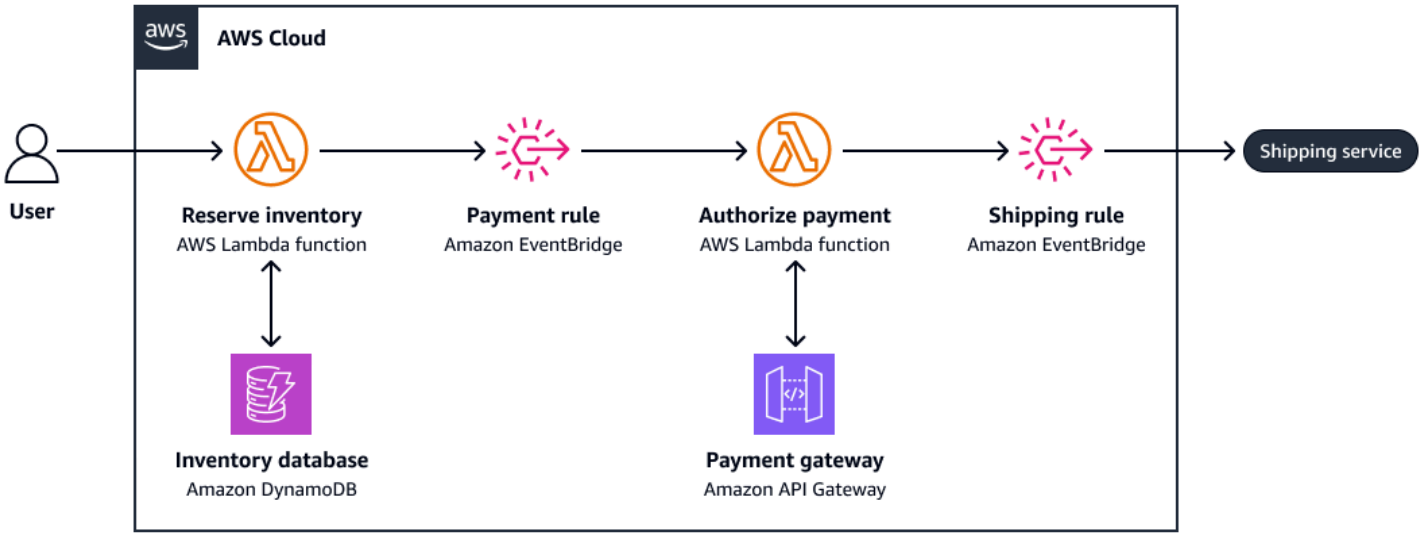
O diagrama a seguir é um exemplo do encadeamento de prompts do LLM:



Coreografia saga

O padrão coreográfico da saga é uma abordagem de implementação em sistemas distribuídos que não tem um coordenador central. Em vez disso, cada serviço ou componente publica eventos que acionam a próxima ação do fluxo de trabalho. Esse padrão é amplamente usado em sistemas distribuídos para gerenciar transações em vários serviços. Em uma saga, o sistema executa uma série de transações locais coordenadas. Se um falhar, o sistema aciona ações compensatórias para manter a consistência.

O diagrama a seguir é um exemplo de coreografia de saga:



1. Reservar inventário

2. Autorizar pagamento
3. Criar pedido de envio

Se a etapa 3 falhar, o sistema invoca ações compensatórias (por exemplo, cancelar um pagamento ou liberar o inventário).

Esse padrão é especialmente valioso em arquiteturas orientadas a eventos, nas quais os serviços são fracamente acoplados e os estados devem ser resolvidos de forma consistente ao longo do tempo, mesmo na presença de falhas parciais.

Padrão de encadeamento imediato

O encadeamento imediato se assemelha ao padrão da saga, tanto na estrutura quanto no propósito. Ele executa uma série de etapas de raciocínio que são construídas sequencialmente, preservando o contexto e permitindo reversões e revisões.

Coreografia do agente

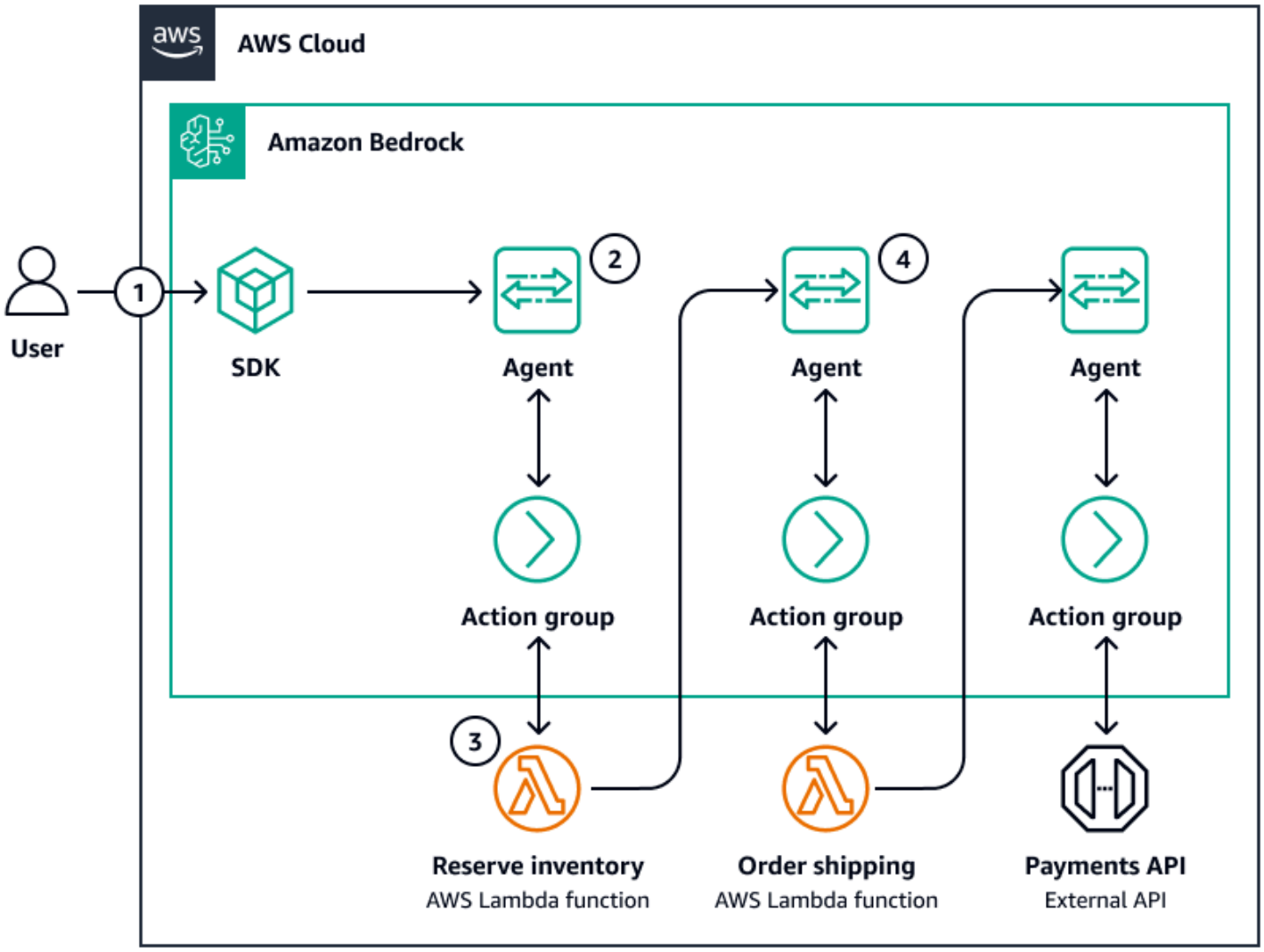
1. O LLM interpreta uma consulta complexa do usuário e gera uma hipótese
2. LLM elabora um plano para resolver a tarefa
3. O LLM executa uma subtarefa (por exemplo, usando uma chamada de ferramenta ou recuperando conhecimento)
4. O LLM refina a saída ou revisita uma etapa anterior se considerar um resultado insatisfatório

Se um resultado intermediário apresentar falhas, o sistema poderá fazer o seguinte:

- Repita as etapas usando uma abordagem diferente
- Reverta para uma solicitação anterior e replaneje
- Use um loop de avaliação (por exemplo, do padrão avaliador-otimizador) para detectar e corrigir falhas

Assim como o padrão da saga, o encadeamento imediato permite mecanismos parciais de progresso e reversão. Isso acontece por meio de refinamento iterativo e correção direcionada ao LLM, em vez de compensar as transações do banco de dados.

O diagrama a seguir é um exemplo de coreografia de agentes:



1. Um usuário envia uma consulta por meio de um SDK.
2. Um agente do Amazon Bedrock orquestra o raciocínio por meio do seguinte:
   - Interpretação (LLM)
   - Planejamento (LLM)
   - Execução por meio de uma ferramenta ou base de conhecimento
   - Construção da resposta
3. Se uma ferramenta falhar ou retornar dados insuficientes, o agente poderá replanejar ou reformular dinamicamente a tarefa.
4. A memória (por exemplo, um armazenamento vetorial de curto prazo) pode preservar seu estado em todas as etapas

Takeaways

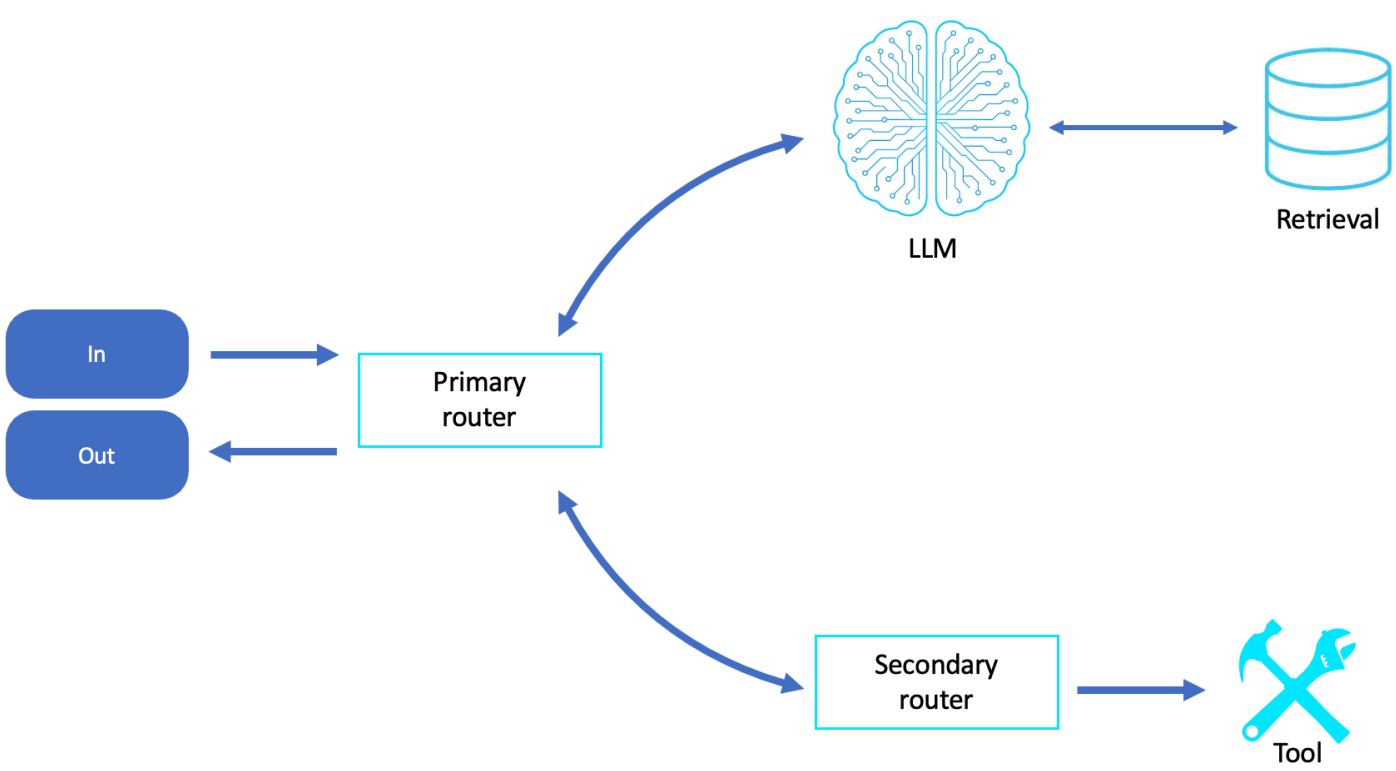
Enquanto o padrão da saga gerencia chamadas de serviço distribuídas com lógica de compensação, o encadeamento imediato gerencia as tarefas de raciocínio com sequenciamento reflexivo e replanejamento adaptativo. Ambos os sistemas permitem progresso incremental, pontos de decisão descentralizados e recuperação de falhas, e tudo isso por meio de raciocínio informado, em vez de reversão rígida.

O encadeamento imediato introduz o raciocínio transacional, que é o equivalente cognitivo das sagas. Ou seja, cada “pensamento” é reavaliado, revisado ou abandonado como parte de um diálogo mais amplo direcionado a um objetivo.

Padrões de despacho dinâmico de roteamento

Em sistemas agentes modernos, onde as tarefas variam da análise de documentos à geração autônoma de software, a capacidade de rotear solicitações dinamicamente para o modelo de linguagem grande (LLM) ou agente mais capaz se torna essencial. A lógica de roteamento estático, geralmente incorporada em scripts de orquestração ou camadas de API, carece da adaptabilidade necessária para ambientes em tempo real, com vários modelos e com vários recursos. Para resolver isso, os fluxos de trabalho de roteamento do LLM podem ser transformados em uma arquitetura orientada a eventos que aproveita um padrão de despacho dinâmico, transformando as chamadas do LLM em eventos roteados de forma inteligente e sensíveis ao contexto.

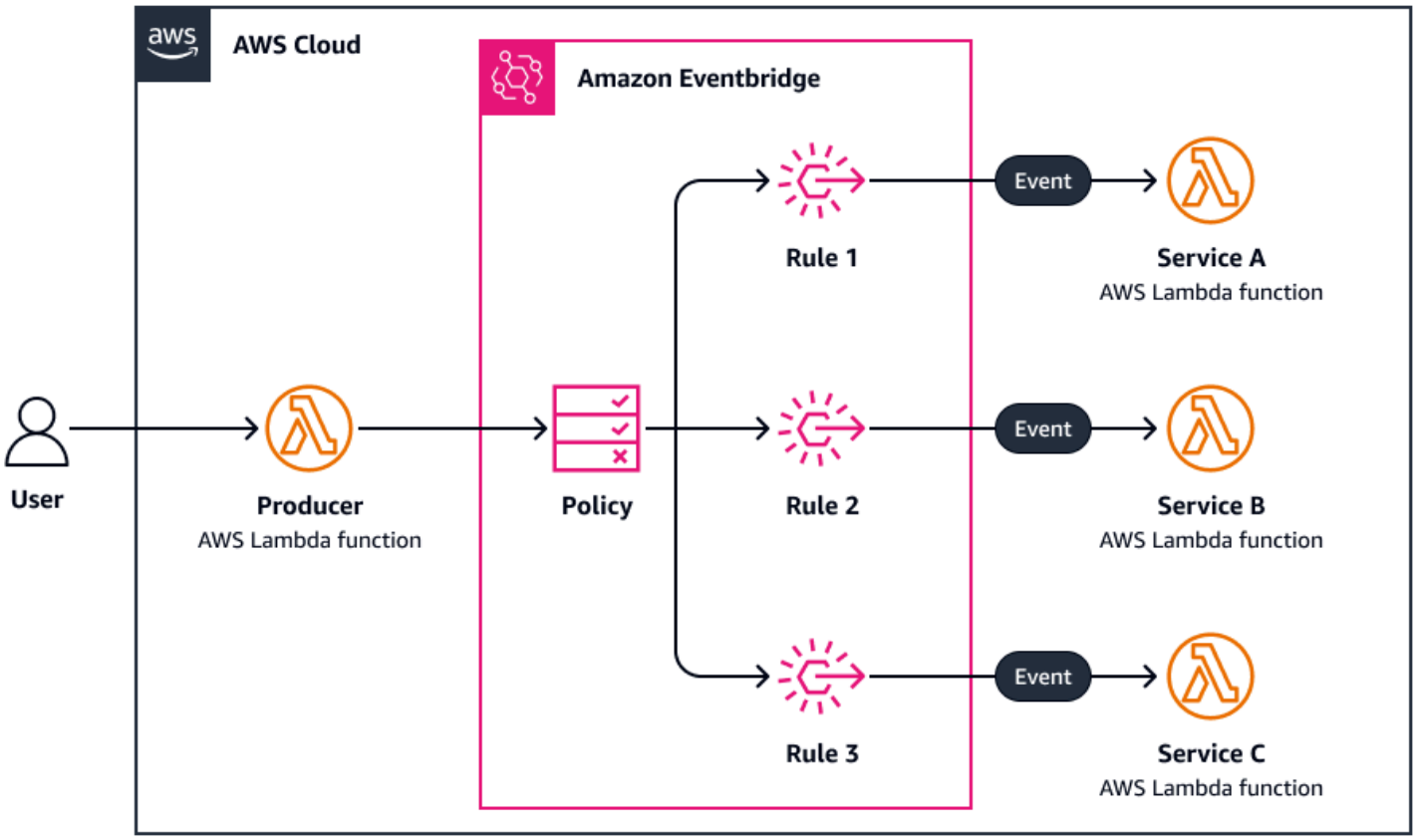
O diagrama a seguir é um exemplo de roteamento LLM:



Despacho dinâmico

Em sistemas distribuídos tradicionais, o padrão de despacho dinâmico seleciona e invoca serviços específicos em tempo de execução com base nos atributos de eventos de entrada, como tipo de evento, origem e carga útil. Isso geralmente é implementado usando a Amazon EventBridge, que pode avaliar e rotear eventos recebidos para destinos apropriados (por exemplo AWS Step Functions, AWS Lambda funções ou tarefas do Amazon Elastic Container Service).

O diagrama a seguir é um exemplo de despacho dinâmico:



1. Um aplicativo emite um evento (por exemplo, {"type": “orderCreated”, “priority”: “high"}).
2. A Amazon EventBridge avalia o evento de acordo com suas regras de roteamento.
3. Com base nos atributos de um evento, o sistema despacha dinamicamente para o seguinte:
   - `HighPriorityOrderProcessor`(serviço A)
   - `StandardOrderProcessor`(serviço B)
   - `UpdateOrderProcessor`(serviço C)

Esse padrão oferece suporte a acoplamento frouxo, especialização baseada em domínio e extensibilidade de tempo de execução. Isso permite que os sistemas respondam de forma inteligente às mudanças nos requisitos e na semântica dos eventos.

Roteamento baseado em LLM

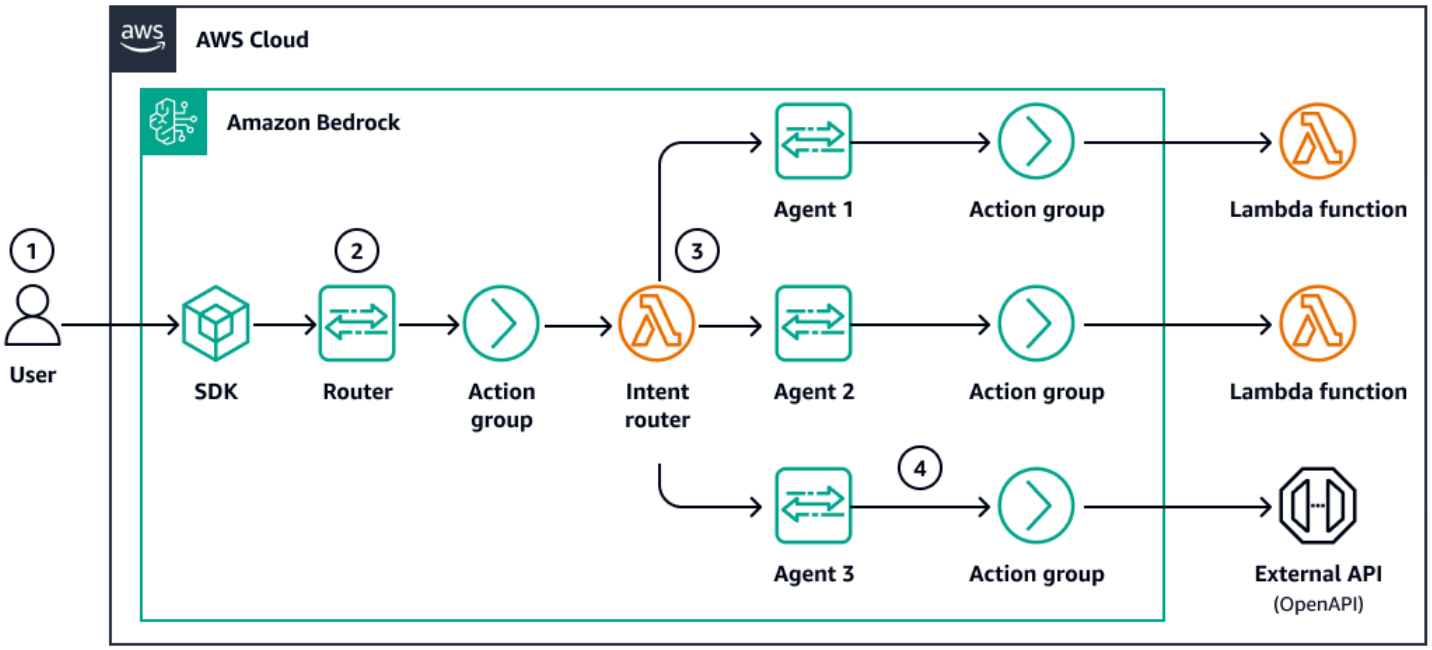
Em sistemas agentes, o roteamento também executa a delegação dinâmica de tarefas — mas, em vez EventBridge das regras ou filtros de metadados da Amazon, o LLM classifica e interpreta a intenção do usuário por meio da linguagem natural. O resultado é uma forma de despacho flexível, semântica e adaptável.

Roteador de agente

Essa arquitetura permite um envio rico baseado em intenção sem esquemas ou tipos de eventos predefinidos, o que é ideal para entradas não estruturadas e consultas complexas.

1. Um usuário envia a solicitação “Você pode me ajudar a revisar os termos do meu contrato?”
2. O LLM interpreta isso como uma tarefa de documento legal.
3. O agente encaminha a tarefa para uma ou mais das seguintes opções:
   - Modelo de solicitação de revisão de contrato
   - Subagente de raciocínio jurídico
   - Ferramenta de análise de documentos

O diagrama a seguir é um exemplo de um roteador de agente:



1. Um usuário envia uma solicitação de linguagem natural por meio de um SDK.
2. Um agente do Amazon Bedrock usa um LLM para classificar a tarefa (por exemplo, jurídica, técnica ou agendamento).
3. O agente roteia dinamicamente a tarefa por meio de um grupo de ações para invocar o agente necessário:
   - Agente específico de domínio
   - Cadeia de ferramentas especializada

- Configuração de prompt personalizada

4. O manipulador selecionado processa a tarefa e retorna uma resposta personalizada.

Takeaways

Enquanto o despacho dinâmico tradicional usa EventBridge as regras da Amazon para roteamento com base em atributos de eventos estruturados, o roteamento agente usa LLMs para classificar e rotear semanticamente as tarefas com base no significado e na intenção. Isso expande a flexibilidade do sistema ao permitir o seguinte:

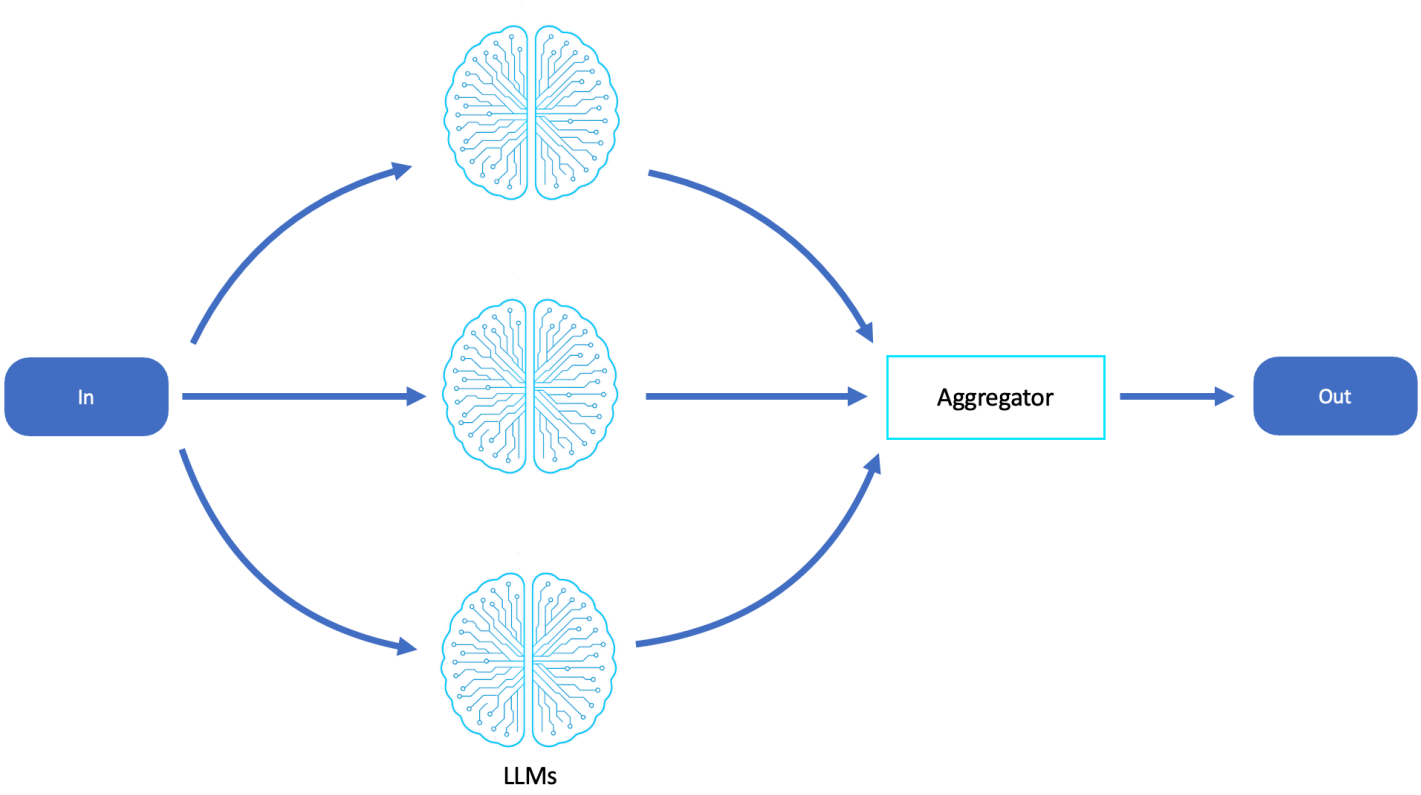
- Compreensão mais ampla dos insumos
- Recuo inteligente e seleção de ferramentas
- Extensibilidade natural por meio de novas funções de agente ou estilos de solicitação

O roteamento agente substitui regras rígidas pelo despacho cognitivo dinâmico, o que permite que os sistemas evoluam com a linguagem e não com o código.

Padrões de paralelização e coleta de dispersão

Muitas tarefas avançadas de raciocínio e geração — como resumir documentos grandes, avaliar vários caminhos de solução ou comparar perspectivas diversas — se beneficiam da execução paralela de prompts. Os fluxos de trabalho sequenciais tradicionais são insuficientes quando são necessárias escalabilidade, capacidade de resposta e tolerância a falhas. Para superar isso, a paralelização baseada em LLM pode ser reimaginada usando um padrão de dispersão e coleta orientado por eventos, em que as tarefas são distribuídas dinamicamente para agentes autônomos e os resultados sintetizados de forma inteligente.

O diagrama a seguir é um exemplo de um fluxo de trabalho de paralelização do LLM:



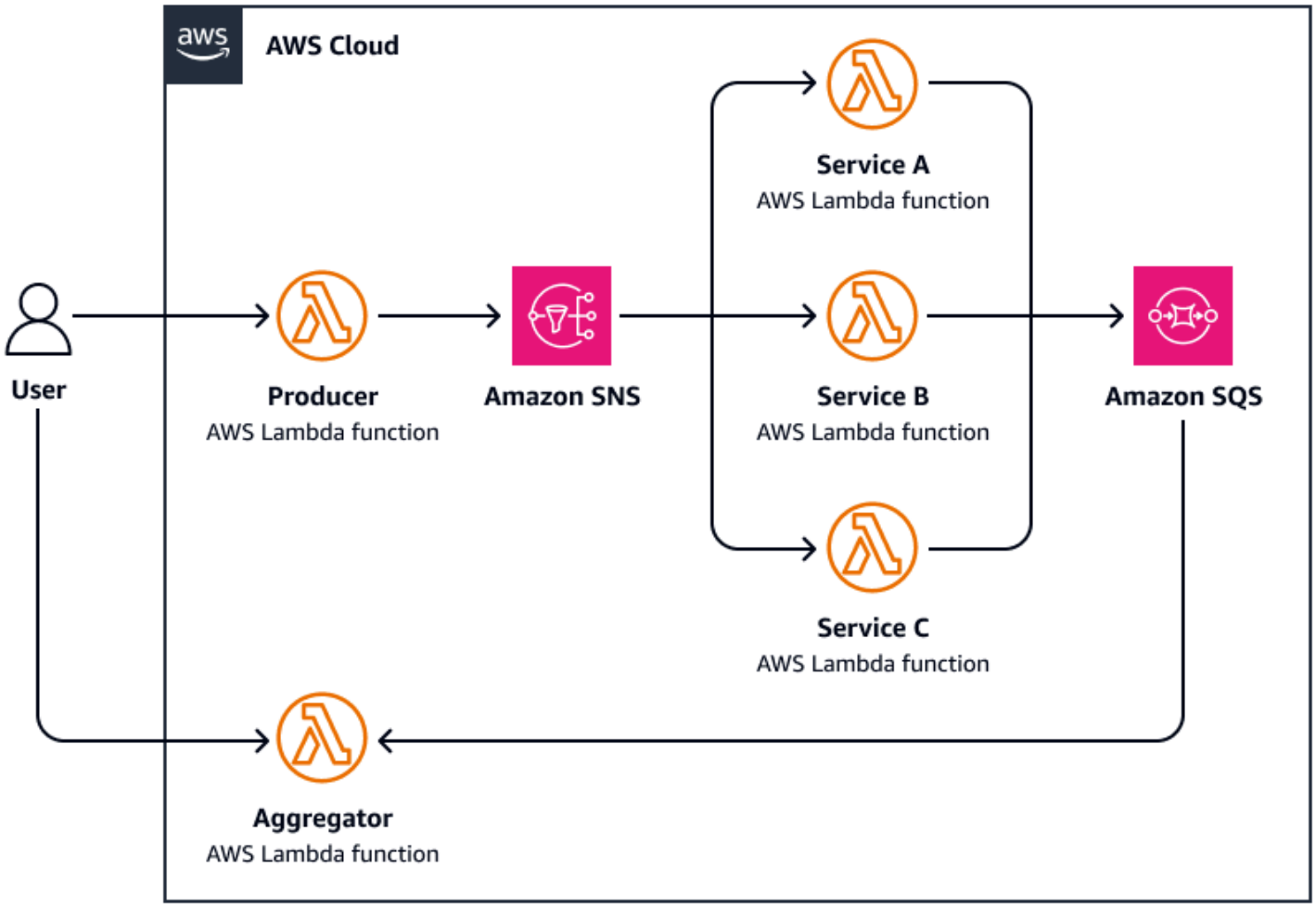
Scatter-gather

Em sistemas distribuídos, um padrão de dispersão envia tarefas para vários serviços ou unidades de processamento em paralelo, aguarda suas respostas e, em seguida, agrega os resultados em uma saída consolidada. Ao contrário do fan-out, o scatter-gather é coordenado porque espera respostas e geralmente aplica a lógica para combinar, comparar e selecionar resultados.

As implementações comuns para paralelização e coleta de dispersão incluem o seguinte:

- AWS Step Functions mapear um estado para execução paralela de tarefas
- AWS Lambda com simultaneidade, coordenando resultados de várias funções invocadas
- Amazon EventBridge com fluxos de trabalho de correlação IDs e agregação
- Padrão de controlador personalizado para gerenciar o fan-out e coletar resultados usando o Amazon Simple Storage Service (Amazon S3), o Amazon DynamoDB ou filas

O diagrama a seguir é um exemplo de coleta dispersa:



1. Um usuário envia uma solicitação para uma função de coordenador central que dispersa a tarefa publicando mensagens paralelas em um tópico do Amazon Simple Notification Service (Amazon SNS).
2. Cada mensagem inclui metadados da tarefa e é encaminhada para um funcionário especializado. AWS Lambda
3. Cada funcionário processa de AWS Lambda forma independente sua subtarefa atribuída (por exemplo, consultar uma API externa, processar um documento e analisar dados).
4. Os resultados são gravados em uma camada de armazenamento comum, como o Amazon Simple Queue Service (Amazon SQS).
5. A função agregadora espera que todas as respostas sejam concluídas e, em seguida, faz o seguinte:
   - Reúne e agrega os resultados (por exemplo, mescla resumos, seleciona as melhores correspondências)

- Envia uma resposta final ou aciona um fluxo de trabalho posterior

Os casos de uso comuns para padrões de coleta de dispersão incluem o seguinte:

- Pesquisa federada
- Mecanismos de comparação de preços
- Análise de dados agregados
- Inferência multimodelo

Paralelização baseada em LLM (cognição de dispersão e coleta)

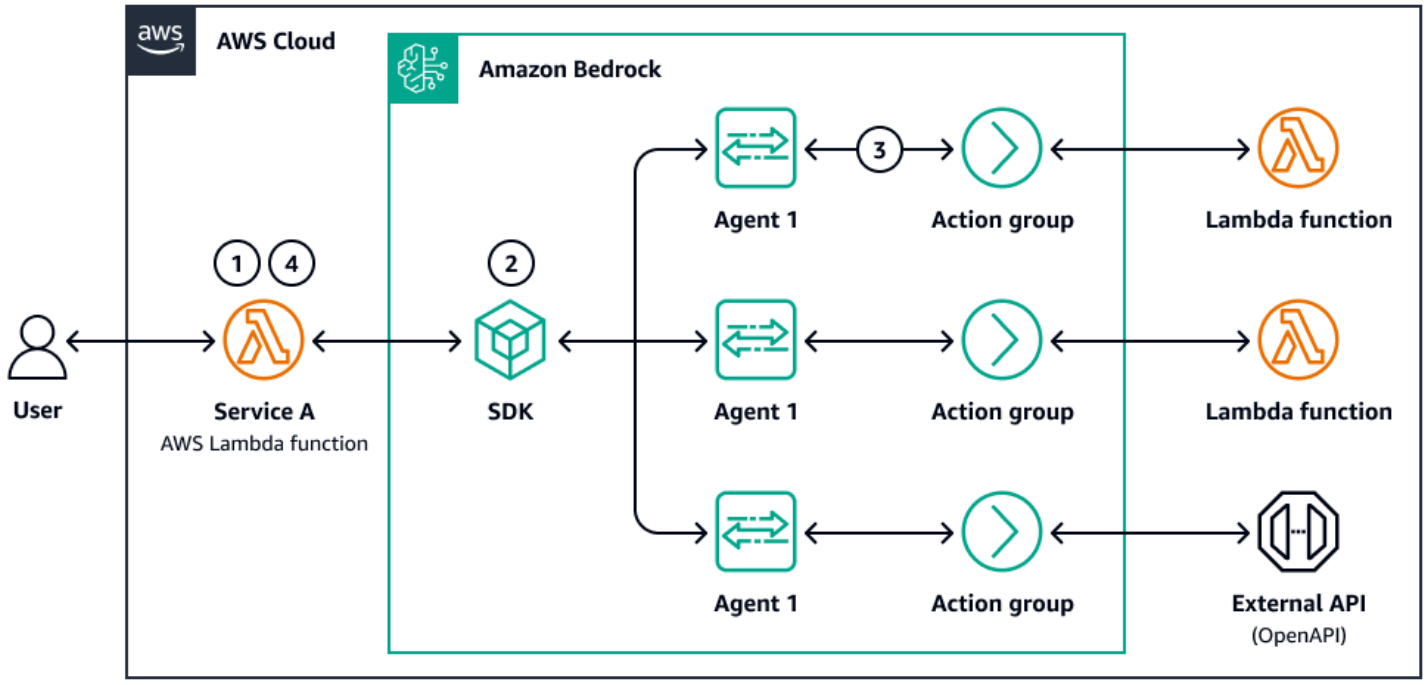
Em sistemas agentes, a paralelização reflete de perto a coleta dispersa, distribuindo subtarefas em várias chamadas ou agentes do LLM, cada um raciocinando de forma independente sobre uma parte do problema. Os resultados retornados são coletados e sintetizados por um processo de agregação, que geralmente é outro agente LLM ou controlador.

Paralelização de agentes

1. Um agente envia uma solicitação “Resuma os insights nesses 10 relatórios”.
2. Ele dispersa os relatórios em 10 tarefas paralelas de resumo do LLM.
3. Ao retornar todos os resumos, o agente faz o seguinte:
   - Agrega resumos em um briefing unificado
   - Identifica temas ou contradições
   - Envia a saída sintetizada para o usuário

Esse fluxo de trabalho agente permite um raciocínio paralelo escalável, modular e adaptável. Isso é ideal para casos de uso que exigem alto rendimento cognitivo.

O diagrama a seguir é um exemplo de paralelização de agentes:



1. Um usuário envia uma consulta com várias partes ou um conjunto de documentos.
2. Um controlador AWS Lambda ou função de etapa distribui as subtarefas. Cada tarefa invoca uma chamada ou subagente do Amazon Bedrock LLM com seu próprio prompt.
3. Quando as chamadas e subtarefas são concluídas, os resultados são armazenados (por exemplo, no Amazon S3 ou no armazenamento de memória) e uma etapa de agregação mescla, compara ou filtra as saídas.
4. O sistema retorna a resposta final ao usuário ou ao agente downstream.

Esse sistema tem um ciclo de raciocínio distribuído com rastreabilidade, tolerância a falhas e ponderação opcional de resultados ou lógica de seleção.

Takeaways

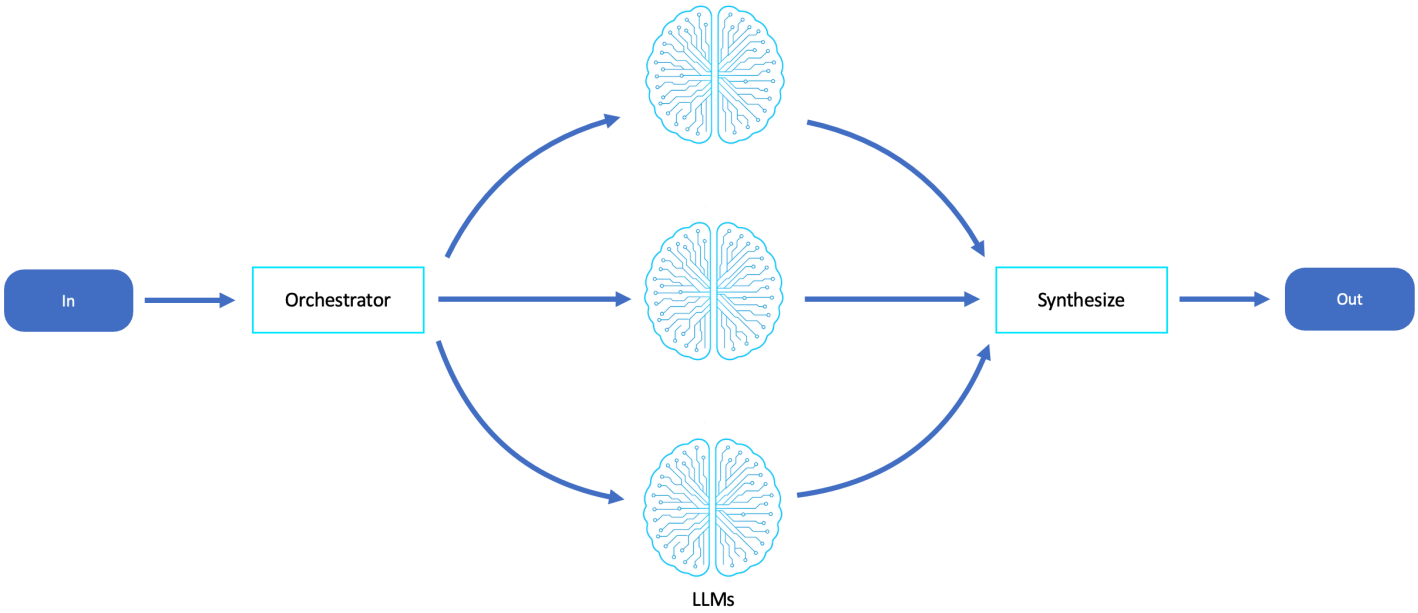
A paralelização agente usa padrões de coleta de dispersão para distribuir tarefas de LLM, permitindo processamento paralelo e síntese inteligente de resultados.

Padrões de orquestração da saga

À medida que os fluxos de trabalho orientados por LLMs se tornam cada vez mais complexos, abrangendo cadeias de solicitações, etapas de processamento de dados, invocações de ferramentas

e colaboração de agentes, a necessidade de orquestração inteligente se torna essencial. Em vez de depender de scripts fortemente acoplados ou fluxos de execução estáticos predeterminados, esses fluxos de trabalho podem ser implementados como padrões de orquestração orientados por eventos, permitindo que sistemas baseados em LLM coordenem, monitorem e adaptem dinamicamente tarefas de várias etapas em agentes autônomos.

O diagrama a seguir é um exemplo de um orquestrador:



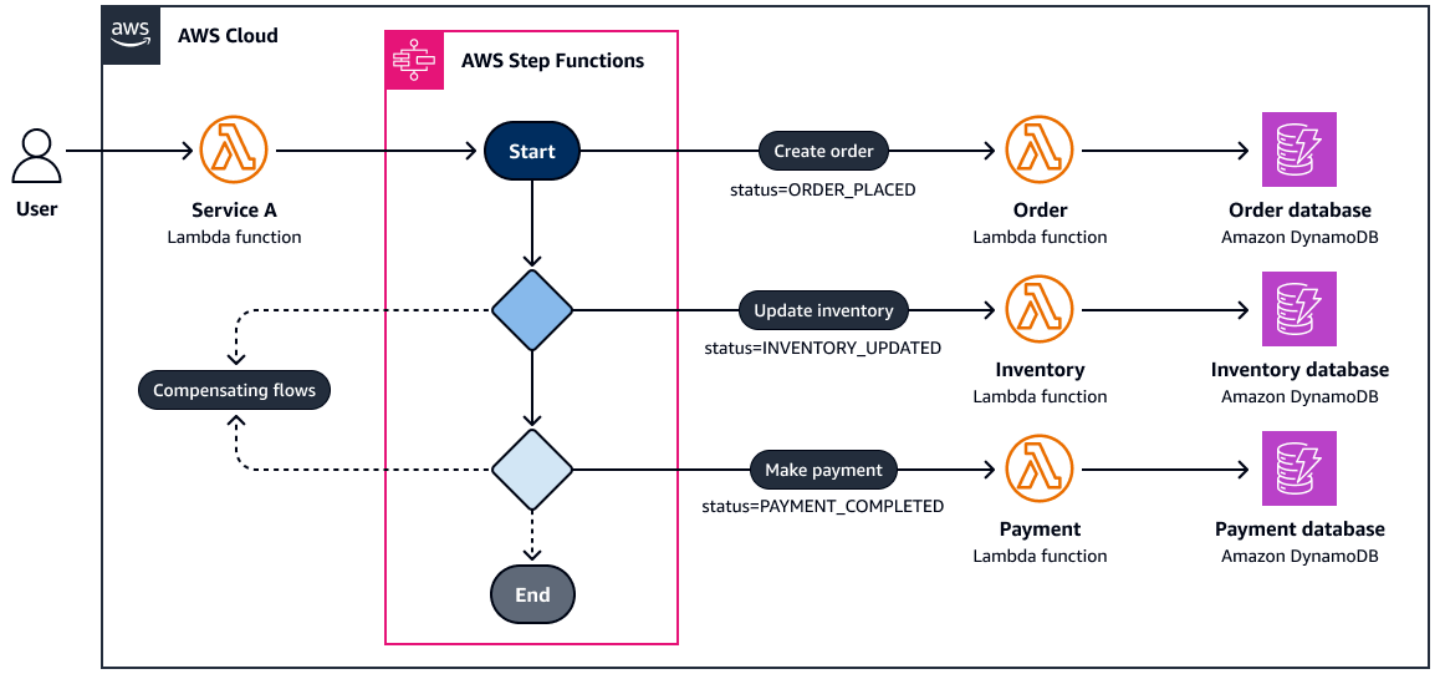
Orquestração de eventos

Em sistemas distribuídos tradicionais, a orquestração de eventos se refere a um padrão no qual um coordenador central gerencia um fluxo de trabalho complexo direcionando explicitamente o fluxo de controle em vários serviços ou tarefas. Diferentemente da coreografia de eventos (em que cada serviço reage de forma independente), a orquestração fornece lógica, visibilidade e controle centralizados sobre todo o processo.

Isso geralmente é implementado usando as seguintes ferramentas:

- AWS Step Functions— Definir e executar fluxos de trabalho com estado
- AWS Lambda— Execute tarefas discretas dentro do fluxo orquestrado
- Amazon SQS ou Amazon EventBridge — aciona etapas ou respostas assíncronas

O diagrama a seguir é um exemplo de orquestração de saga:



Um AWS Step Functions fluxo de trabalho gerencia o processo de pedido de um cliente:

1. Criar pedido (AWS Lambda)
2. Atualizar inventário (AWS Lambda)
3. Efetue o pagamento (AWS Lambda)

O orquestrador coordena cada etapa gerenciando novas tentativas, ramificações paralelas, tempos limite e falhas.

Sistema de agentes baseado em funções (orquestrador)

Em sistemas agentes, o padrão do orquestrador reflete a orquestração de eventos, mas distribui a lógica entre vários agentes de raciocínio, cada um com uma função ou especialização definida. Um agente orquestrador central interpreta a tarefa geral, a decompõe em subtarefas e as delega aos agentes de trabalho, cada um otimizado para um domínio específico (por exemplo, pesquisa, codificação, resumo, revisão).

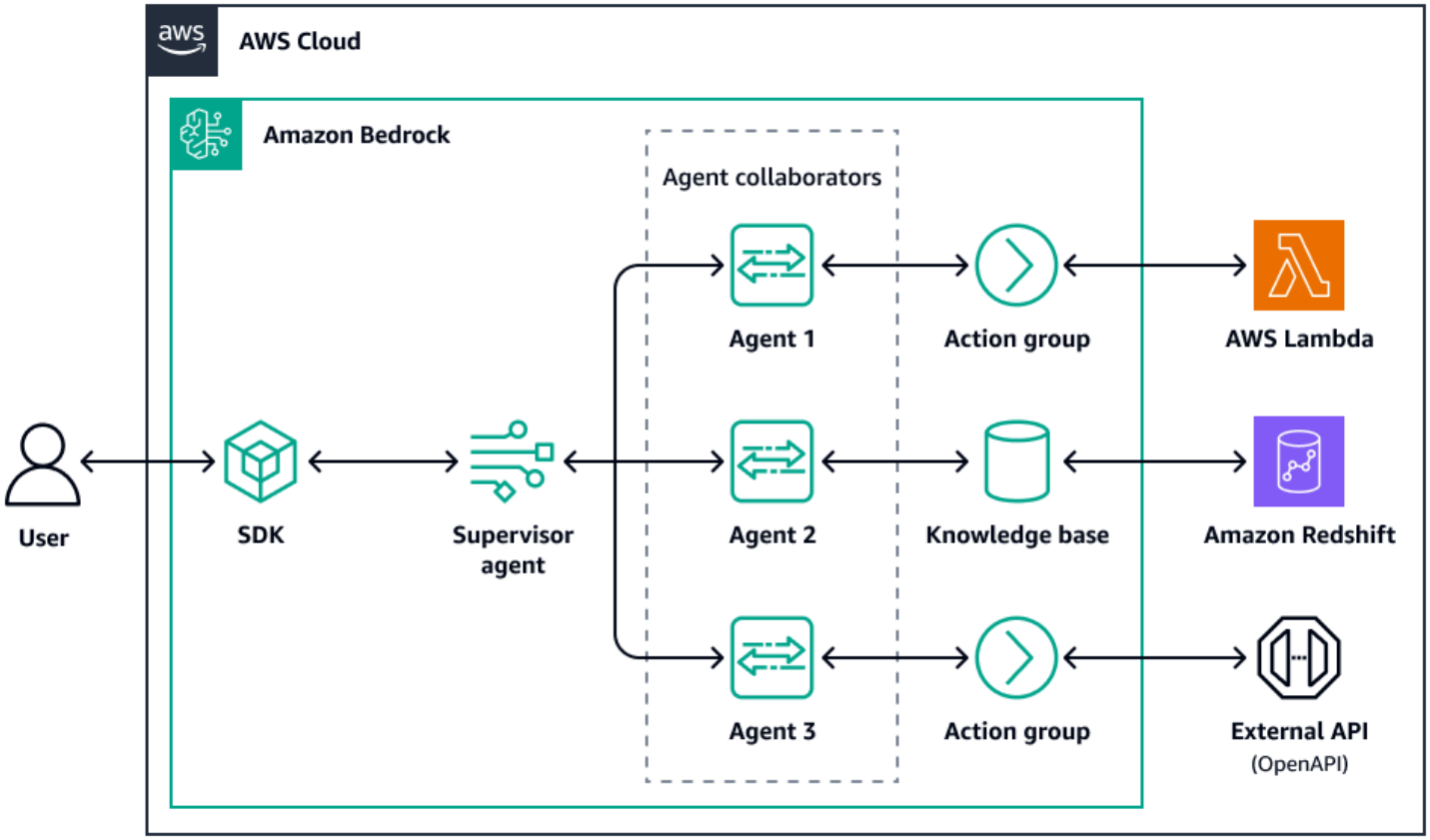
Supervisor

1. Um usuário envia a consulta “Crie um resumo do projeto e resuma os 5 principais concorrentes”.
2. O agente orquestrador faz o seguinte:

- Designa um agente de pesquisa para encontrar dados da concorrência
- Envia as descobertas brutas para um agente de sumarização
- Transmite os resultados para um agente redator de resumos
- Compila a saída final para o usuário

Cada agente opera de forma independente, mas o orquestrador coordena as tarefas. É como uma função Lambda que lida com tarefas de fluxo de trabalho.

O diagrama a seguir é um exemplo de um supervisor:



1. Um usuário envia uma tarefa para um agente supervisor do Amazon Bedrock.
2. O agente supervisor analisa a solicitação em subtarefas para cada agente colaborador.
3. Cada subtarefa é atribuída a um agente colaborador com solicitações ou conjuntos de ferramentas específicos da função.
4. Agentes de trabalho ligam para ferramentas externas APIs ou por meio de um grupo de ação.
5. Cada agente de trabalho retorna a saída em um formato estruturado.

6. Quando todos os trabalhadores retornam seus resultados, o supervisor avalia, sintetiza e retorna a resposta final.

Essa estrutura permite modularidade, adaptabilidade e introspecção em fluxos de trabalho complexos de agentes de várias etapas.

Takeaways

Onde a orquestração de eventos usa controle centralizado (por exemplo, AWS Step Functions) para direcionar a execução do serviço, os sistemas de agentes baseados em funções usam um agente orquestrador baseado em LLM para raciocinar sobre a meta, delegar subtarefas aos agentes de trabalho e sintetizar a saída final.

Em ambos os paradigmas, o orquestrador faz o seguinte:

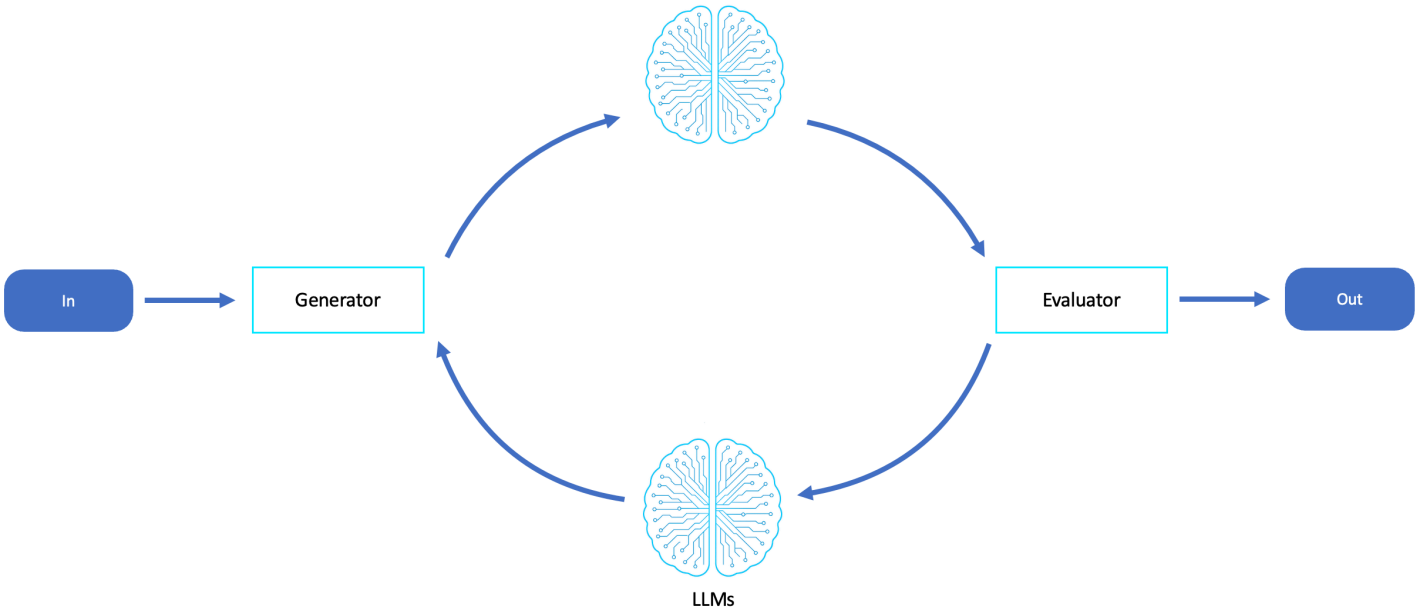
- Mantém o contexto e o fluxo de execução
- Lida com ramificação, sequenciamento e tratamento de erros
- Produz um resultado unificado a partir de componentes distribuídos

A orquestração agente, no entanto, acrescenta raciocínio, adaptabilidade e delegação semântica. Isso o torna adequado para tarefas abertas, ambíguas e em evolução.

O avaliador reflete e refina os padrões de loop

Tarefas como geração de código, resumo ou tomada de decisão autônoma se beneficiam muito do feedback de tempo de execução, permitindo que o sistema evolua por meio de observação e refinamento. Para operacionalizar isso, o ciclo reflect-refine pode ser implementado como um circuito de controle de feedback orientado por eventos — um padrão inspirado na engenharia de sistemas, adaptado para fluxos de trabalho autônomos e inteligentes.

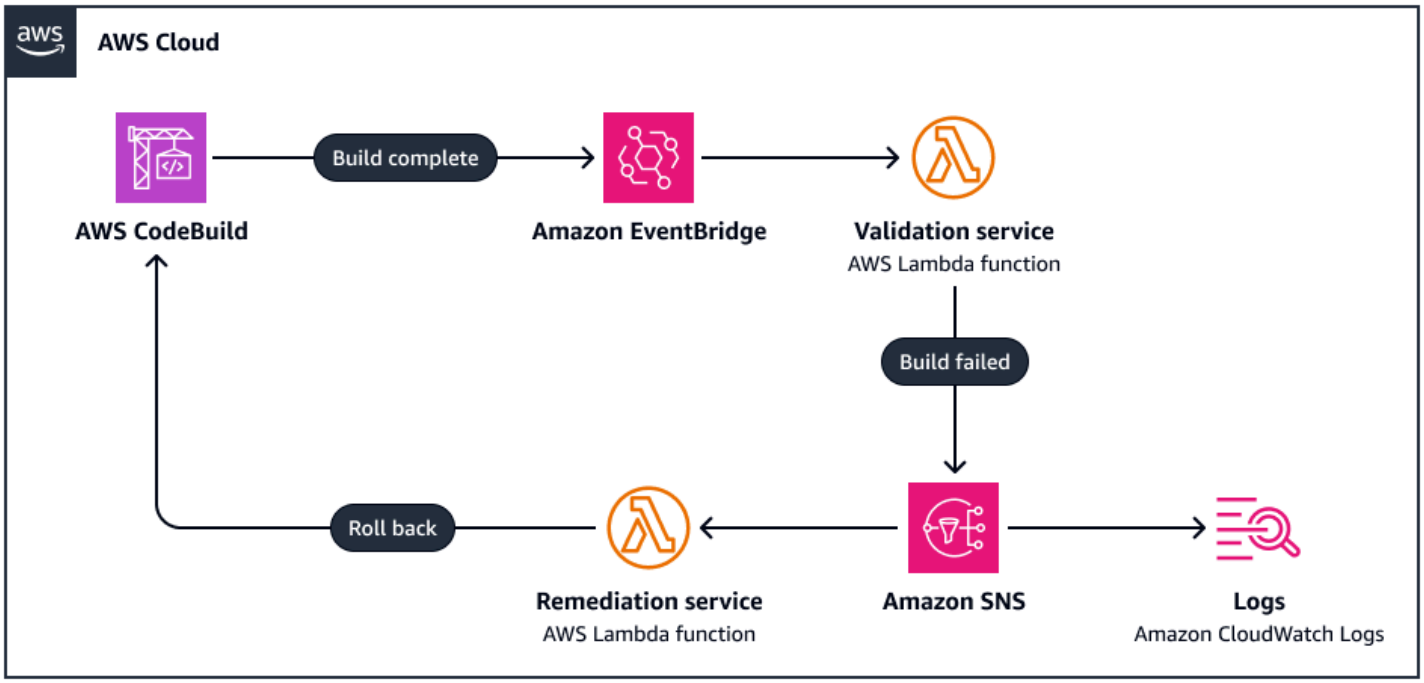
O diagrama a seguir é um exemplo de um ciclo de feedback de reflexão e refinamento do avaliador:



Circuito de controle de feedback

Um loop de controle de feedback é um padrão que monitora suas próprias saídas e comportamentos, os avalia em relação a critérios definidos ou a um estado desejado e, em seguida, ajusta suas ações de acordo. Essa arquitetura é inspirada na teoria de controle e é fundamental em domínios como automação, pipelines de integração e entrega contínuas (CI/CD) e operações de aprendizado de máquina.

O diagrama a seguir é um exemplo de um circuito de controle de feedback:



1. Um pipeline de implantação emite um evento BuildComplete.
2. O evento aciona um teste automatizado ou um trabalho de avaliação que valida a compilação.
3. Se a validação falhar (por exemplo, devido a falhas nos testes, problemas de segurança ou violação da política), o sistema:
   - Emite um evento BuildComplete
   - Registra o problema ou envia uma notificação
   - Aciona uma ação corretiva ou de remediação, como reversão, aplicação de patches ou nova tentativa

O ciclo continua até produzir um resultado ou escalonamento aceitável, ou até que ocorra um tempo limite. Esse padrão é comumente usado para o seguinte:

- EventBridge Regras da Amazon para encaminhar eventos para tarefas de avaliação ou remediação
- AWS Step Functions para lógica de repetição iterativa e ramificação dos resultados da avaliação
- Amazon Simple Notification Service (Amazon SNS) ou alarmes da Amazon para CloudWatch acionadores e alertas de feedback
- AWS Lambda funções ou trabalhadores em contêineres para aplicar ações corretivas

Circuito de controle de feedback (avaliador)

O fluxo de trabalho do avaliador é um ciclo de feedback cognitivo desenvolvido por nossos agentes LLMs de raciocínio. O processo consiste no seguinte:

1. Um agente gerador ou LLM produz uma saída (por exemplo, um plano, uma resposta ou um rascunho).
2. Um agente avaliador analisa o resultado usando um aviso de crítica ou uma rubrica de avaliação.
3. Com base no feedback, o agente original ou um novo agente otimizador revisa a saída.

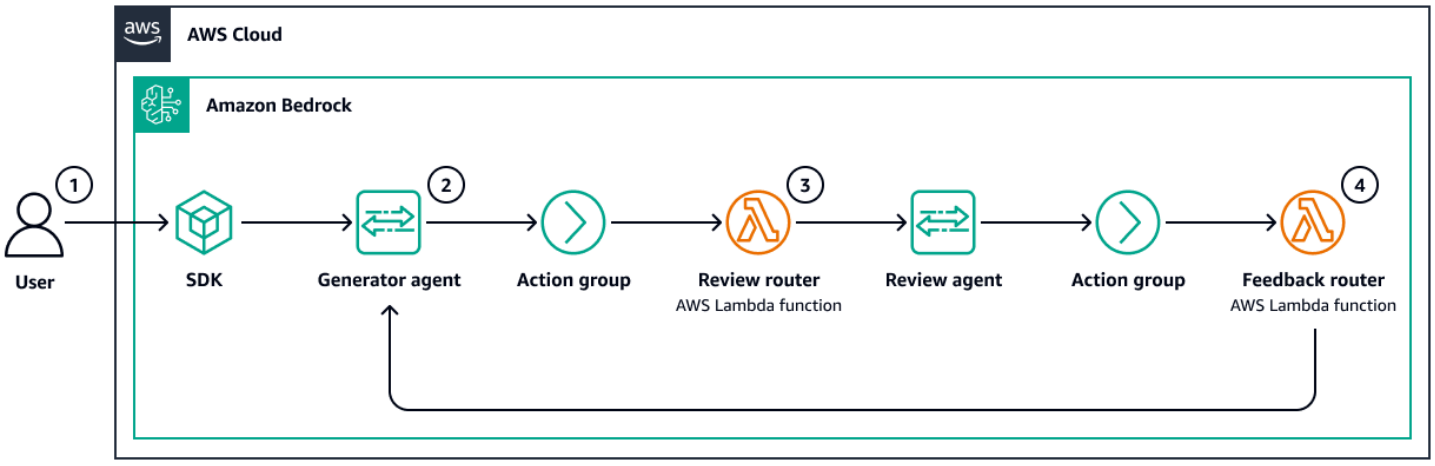
O loop se repete até que o resultado atenda a um conjunto de critérios, seja aprovado ou atinja um limite de repetição.

Avaliador

1. Um usuário pede que um agente escreva um resumo da política.
2. O agente gerador o elabora.
3. Um agente avaliador verifica a cobertura, o tom e a correção legal.
4. Se a resposta for inadequada, ela será refinada e reenviada até que o ciclo de feedback converja.

Isso permite a autoavaliação, o refinamento iterativo e o controle de saída adaptativo, tudo sem a intervenção humana.

O diagrama a seguir é um exemplo de um circuito de controle de feedback (avaliador):



1. Um usuário emite uma tarefa (por exemplo, elabore uma estratégia de negócios).

2. Um agente do Amazon Bedrock gera um rascunho inicial usando um LLM.
3. Um segundo agente (ou uma solicitação de acompanhamento) realiza uma avaliação estruturada (por exemplo, “classifique essa saída por clareza, integridade e tom”).
4. Se a classificação cair abaixo de um limite, a resposta será revisada por:
   - Reinvocando o gerador com uma crítica incorporada
   - Enviando o feedback para um agente refinador especializado
   - Iterando até que uma resposta aceitável seja alcançada

Componentes opcionais, como AWS Lambda controladores, AWS Step Functions podem gerenciar limites de feedback, novas tentativas e estratégias alternativas.

Takeaways

Enquanto os ciclos tradicionais de controle de feedback usam eventos, métricas e lógica de remediação para validar e ajustar o comportamento do sistema, os loops avaliadores agentes usam agentes de raciocínio para avaliar, refletir e revisar a saída dinamicamente.

Em ambos os paradigmas:

- A saída é avaliada após ser gerada
- Ações corretivas ou de refinamento são acionadas com base no feedback
- O sistema se adapta continuamente a uma meta ou qualidade alvo

A versão agente transforma a validação estática em reflexão semântica, permitindo que agentes de autoaperfeiçoamento avaliem sua própria eficácia.

Projetando fluxos de trabalho agentes em AWS

Cada padrão neste guia pode ser criado usando Serviços da AWS. Os agentes do Amazon Bedrock fornecem canais de orquestração, acesso a dados e interação.

Componente	AWS service (Serviço da AWS)	Finalidade
Raciocínio LLM	Amazon Bedrock	Lógica do agente, planejamento, uso de ferramentas

Execução da ferramenta	AWS Lambda, Amazon ECS, Amazon SageMaker	Hospede ferramentas externas para agentes
Memória e RAG	Base de conhecimento Amazon Bedrock, Amazon S3, OpenSearch	Memória persistente e semântica
Orquestração	AWS Step Functions	Coordenação de tarefas e agentes em várias etapas
Roteamento de eventos	Amazon EventBridge, Amazon SQS	Mensagens interagentes desacopladas
Interface do usuário	Amazon API Gateway AWS AppSync, SDK	Pontos de entrada para aplicativos ou sistemas
Monitoramento	Amazon CloudWatch AWS X-Ray, AWS Distro para OpenTelemetry	Observabilidade e introspec ção do agente

Conclusão

Os padrões de fluxo de trabalho agênticos são o próximo estágio evolutivo das arquiteturas orientadas a eventos, em que a lógica de negócios não é definida estaticamente, mas fundamentada dinamicamente por meio do uso de cognição aprimorada por meio do modelo de linguagem grande (LLM). Ao combinar primitivos nativos da nuvem tradicionais com fluxos de trabalho de LLM e padrões de design de agentes, as organizações podem criar sistemas adaptáveis, inteligentes e modulares que respondam com propósito e aprendam com a experiência.

Nesses padrões, o Amazon Bedrock é a porta de entrada para a cognição agente, permitindo que agentes baseados em LLM acessem fluxos de trabalho de eventos, interajam com ferramentas e memória e forneçam resultados estruturados, rastreáveis e alinhados.

À medida que você projeta e implanta sistemas agentes, esses padrões de fluxo de trabalho fornecem planos para a criação de arquiteturas de IA autônomas e combináveis. Esses sistemas são baseados nas melhores práticas sem servidor e aumentados com modelos básicos inteligentes.

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.