



Habilitar a persistência de dados em microsserviços

# AWS Orientação prescritiva



# AWS Orientação prescritiva: Habilitar a persistência de dados em microsserviços

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

---

# Table of Contents

Introdução .....	1
Resultados de negócios desejados .....	3
Padrões para permitir a persistência de dados .....	4
atabase-per-service Padrão D .....	4
Padrão de composição da API .....	6
Padrão CQRS .....	8
Padrão de fornecimento de eventos .....	11
Implementação do Amazon Kinesis Data Streams .....	12
EventBridge Implementação da Amazon .....	13
Padrão Saga .....	15
hared-database-per-service Padrão S .....	16
Perguntas frequentes .....	18
Quando posso modernizar meu banco de dados monolítico como parte da minha jornada de modernização? .....	18
Posso manter um banco de dados monolítico legado para vários microsserviços? .....	18
O que devo considerar ao projetar bancos de dados para uma arquitetura de microsserviços? .....	18
Qual é o padrão comum para manter a consistência de dados em diferentes microsserviços? ...	18
Como faço para manter a automação de transações? .....	19
Preciso usar um banco de dados separado para cada microsserviço? .....	19
Como posso manter a privacidade dos dados persistentes de um microsserviço se todos compartilham um único banco de dados? .....	19
Recursos .....	20
Guias e padrões relacionados .....	20
Outros recursos .....	20
Histórico do documento .....	21
Glossário .....	22
# .....	22
A .....	23
B .....	26
C .....	27
D .....	31
E .....	35
F .....	37

---

G .....	38
H .....	39
I .....	40
L .....	42
M .....	43
O .....	47
P .....	49
Q .....	51
R .....	52
S .....	54
T .....	58
U .....	59
V .....	60
W .....	60
Z .....	61
.....	lxiii

# Habilitar a persistência de dados em microsserviços

Tabby Ward e Balaji Mohan, da Amazon Web Services ( ) AWS

Dezembro de 2023 ([histórico do documento](#))

As organizações buscam constantemente novos processos para criar oportunidades de crescimento e reduzir o tempo de lançamento no mercado. Você pode aumentar a agilidade e a eficiência da sua organização modernizando seus aplicativos, software e sistemas de TI. A modernização também ajuda você a oferecer serviços melhores e mais rápidos aos seus clientes.

A modernização de aplicativos é um gateway para a melhoria contínua de sua organização e começa com a refatoração de um aplicativo monolítico em um conjunto de microsserviços desenvolvidos, implantados e gerenciados de forma independente. Esse processo inclui as seguintes etapas:

- [Decompõe monólitos em microsserviços](#) – Use padrões para decompor aplicativos monolíticos em microsserviços.
- [Integre microsserviços](#) – Integre os microsserviços recém-criados em uma [arquitetura de microsserviços](#) usando serviços de tecnologia sem servidor dos [Amazon Web Services \(AWS\)](#)
- Habilite a persistência de dados para arquitetura de microsserviços – Promova a [persistência poliglota](#) entre seus microsserviços descentralizando os [armazenamentos de dados](#).

Embora você possa usar uma arquitetura de aplicativo monolítica para alguns casos de uso, os atributos modernos de aplicativos geralmente não funcionam em uma arquitetura monolítica. Por exemplo, o aplicativo inteiro não pode permanecer disponível enquanto você atualiza componentes individuais e você não pode escalar componentes individuais para resolver gargalos ou [hotspots](#) (regiões relativamente densas nos dados do seu aplicativo). Os monólitos podem se tornar aplicativos grandes e difíceis de administrar, exigindo esforço e coordenação significativos entre várias equipes para introduzir pequenas mudanças.

Os aplicativos legados geralmente usam um banco de dados monolítico centralizado, o que dificulta as mudanças no esquema, cria uma dependência tecnológica com o escalonamento vertical como a única forma de responder ao crescimento e impõe um único ponto de falha. Um banco de dados monolítico também impede que você crie os componentes descentralizados e independentes necessários para implementar uma arquitetura de microsserviços.

Anteriormente, uma abordagem arquitetônica típica seria modelar todos os requisitos do usuário em um banco de dados relacional usado pelo aplicativo monolítico. Essa abordagem foi apoiada pela arquitetura popular de banco de dados relacional e os arquitetos de aplicativos geralmente projetavam o esquema relacional nos estágios iniciais do processo de desenvolvimento, criavam um esquema altamente normalizado e o enviavam para a equipe de desenvolvedores. No entanto, isso significava que o banco de dados orientava o modelo de dados para o caso de uso do aplicativo, e não o contrário.

Ao optar por descentralizar seus armazenamentos de dados, você promove a [persistência poliglota entre seus microsserviços](#) e identifica sua tecnologia de armazenamento de dados com base nos padrões de acesso aos dados e em outros requisitos de seus microsserviços. Cada microsserviço tem seu próprio armazenamento de dados e pode ser escalado de forma independente com alterações de esquema de baixo impacto, e os dados são controlados por meio da API do microsserviço. Dividir um banco de dados monolítico não é fácil e um dos maiores desafios é estruturar seus dados para obter o melhor desempenho possível. A persistência poliglota descentralizada também geralmente resulta em uma eventual consistência de dados e outros desafios potenciais que exigem uma avaliação completa incluem sincronização de dados durante transações, integridade transacional, duplicação de dados, junções e latência.

Este guia é para proprietários de aplicativos, empresários, arquitetos, líderes técnicos e gerentes de projeto. O guia fornece os seis padrões a seguir para permitir a persistência de dados entre seus microsserviços:

- [atabase-per-service Padrão D](#)
- [Padrão de composição da API](#)
- [Padrão CQRS](#)
- [Padrão de fornecimento de eventos](#)
- [Padrão Saga](#)
  - Para ver as etapas para implementar o padrão saga usando o AWS Step Functions, consulte o padrão [Implementar o padrão saga sem servidor usando o AWS Step Functions](#) no site de Recomendações da AWS.
- [hared-database-per-service Padrão S](#)

O guia é parte de uma série de conteúdos que discutem a abordagem de modernização de aplicativos recomendada pela AWS. A série também inclui:

- [Estratégia para modernizar aplicações na AWS Cloud](#)

- [Abordagem em fases para modernizar aplicações na AWS Cloud](#)
- [Avaliar a preparação para modernização de aplicações na AWS Cloud](#)
- [Decompor monólitos em microsserviços](#)
- [Integrando microsserviços usando serviços de tecnologia sem servidor AWS](#)

## Resultados de negócios desejados

Muitas organizações descobrem que inovar e melhorar a experiência do usuário sofre impacto negativo por bancos de dados, tecnologias e aplicativos monolíticos. Aplicativos e bancos de dados legados reduzem suas opções de adoção de estruturas tecnológicas modernas e restringem sua competitividade e inovação. No entanto, quando você moderniza os aplicativos e seus armazenamentos de dados, eles se tornam mais fáceis de escalar e mais rápidos de desenvolver. Uma estratégia de dados desacoplados melhora a tolerância a falhas e a resiliência, o que ajuda a acelerar o tempo de lançamento no mercado de seus novos atributos de aplicativos.

Você deve esperar os seis resultados a seguir ao promover a persistência de dados entre seus microsserviços:

- Remover bancos de dados monolíticos legados do seu portfólio de aplicativos.
- Melhorar a tolerância a falhas, a resiliência e a disponibilidade de seus aplicativos.
- Reduzir seu tempo de lançamento no mercado para novos atributos de aplicativos.
- Reduzir suas despesas gerais de licenciamento e custos operacionais.
- Aproveite as soluções de código aberto (por exemplo, [MySQL](#) ou [PostgreSQL](#)).
- Crie aplicativos altamente escaláveis e distribuídos, escolhendo entre mais de [15 mecanismos de bancos de dados com propósitos específicos](#) na AWS Cloud.

# Padrões para permitir a persistência de dados

Os padrões a seguir são usados para permitir a persistência de dados em seus microsserviços.

Tópicos

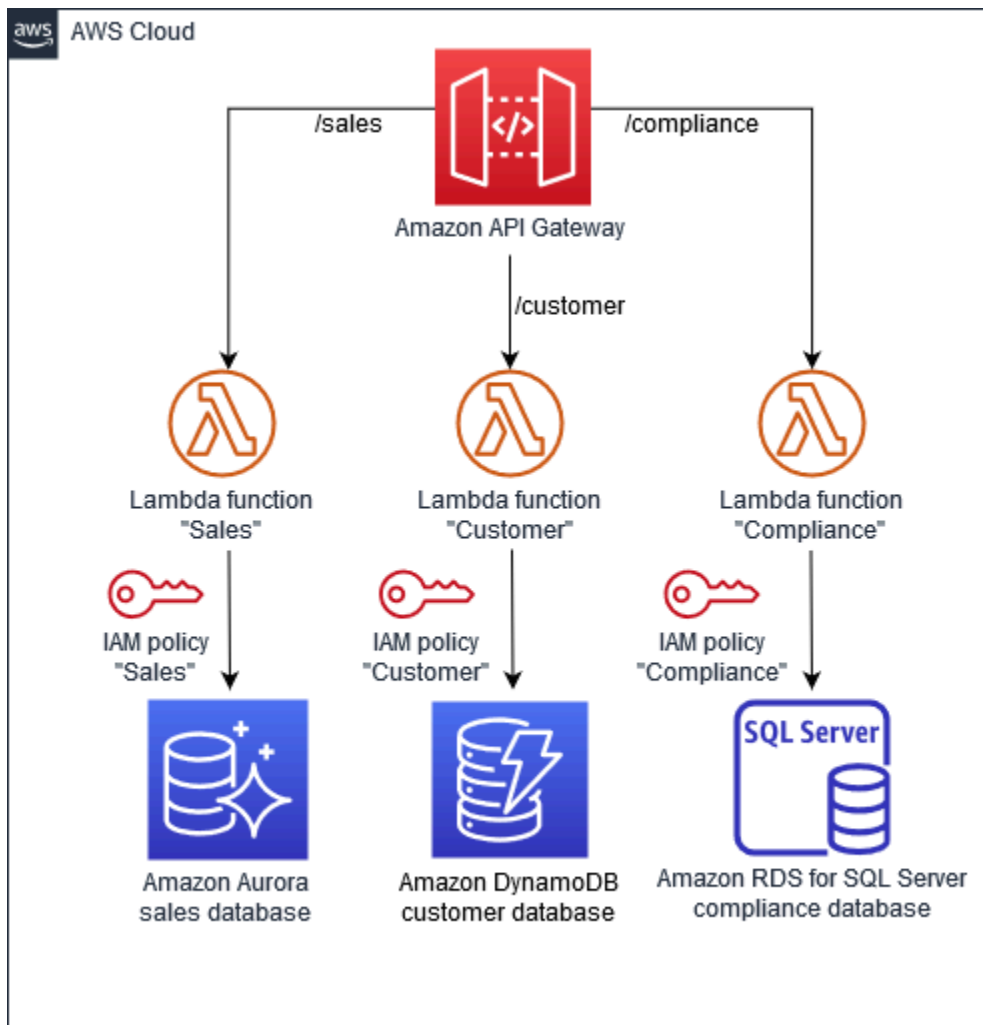
- [atabase-per-service Padrão D](#)
- [Padrão de composição da API](#)
- [Padrão CQRS](#)
- [Padrão de fornecimento de eventos](#)
- [Padrão Saga](#)
- [hared-database-per-service Padrão S](#)

## atabase-per-service Padrão D

O acoplamento fraco é a principal característica de uma arquitetura de microsserviços, porque cada microsserviço individual pode armazenar e recuperar informações de seu próprio armazenamento de dados de forma independente. Ao implantar o database-per-service padrão, você escolhe os armazenamentos de dados mais adequados (por exemplo, bancos de dados relacionais ou não relacionais) para seus requisitos de aplicativos e negócios. Isso significa que os microsserviços não compartilham uma camada de dados, as alterações no banco de dados individual de um microsserviço não afetam outros microsserviços, os armazenamentos de dados individuais não podem ser acessados diretamente por outros microsserviços e os dados persistentes são acessados somente por APIs. A dissociação dos armazenamentos de dados também melhora a resiliência de seu aplicativo geral e garante que um único banco de dados não seja um único ponto de falha.

Na ilustração a seguir, bancos de dados da AWS diferentes são usados pelos microsserviços “Vendas”, “Cliente” e “Conformidade”. Esses microsserviços são implantados como funções da AWS Lambda e acessados por meio de uma API AWS Identity and Access Management do Amazon API Gateway. As políticas (IAM) garantem que os dados sejam mantidos em sigilo e não compartilhados entre os microsserviços. Cada microsserviço usa um tipo de banco de dados que atende aos seus requisitos individuais; por exemplo, “Vendas” usa o Amazon Aurora, “Cliente” usa o Amazon DynamoDB e “Compliance” usa o Amazon Relational Database Service (Amazon RDS) para SQL Server.





Você deve considerar o uso desse padrão se:

- Um acoplamento fraco for necessário entre seus microsserviços.
- Os microsserviços tiverem diferentes requisitos de conformidade ou segurança para seus bancos de dados.
- Um controle mais granular do escalonamento for necessário.

Existem as seguintes desvantagens em usar o database-per-service padrão:

- Pode ser difícil implementar transações e consultas complexas que abrangem vários microsserviços ou armazenamentos de dados.
- Você precisa gerenciar vários bancos de dados relacionais e não relacionais.
- Seus armazenamentos de dados devem atender a dois dos requisitos do [teorema CAP](#): consistência, disponibilidade ou tolerância à partição.

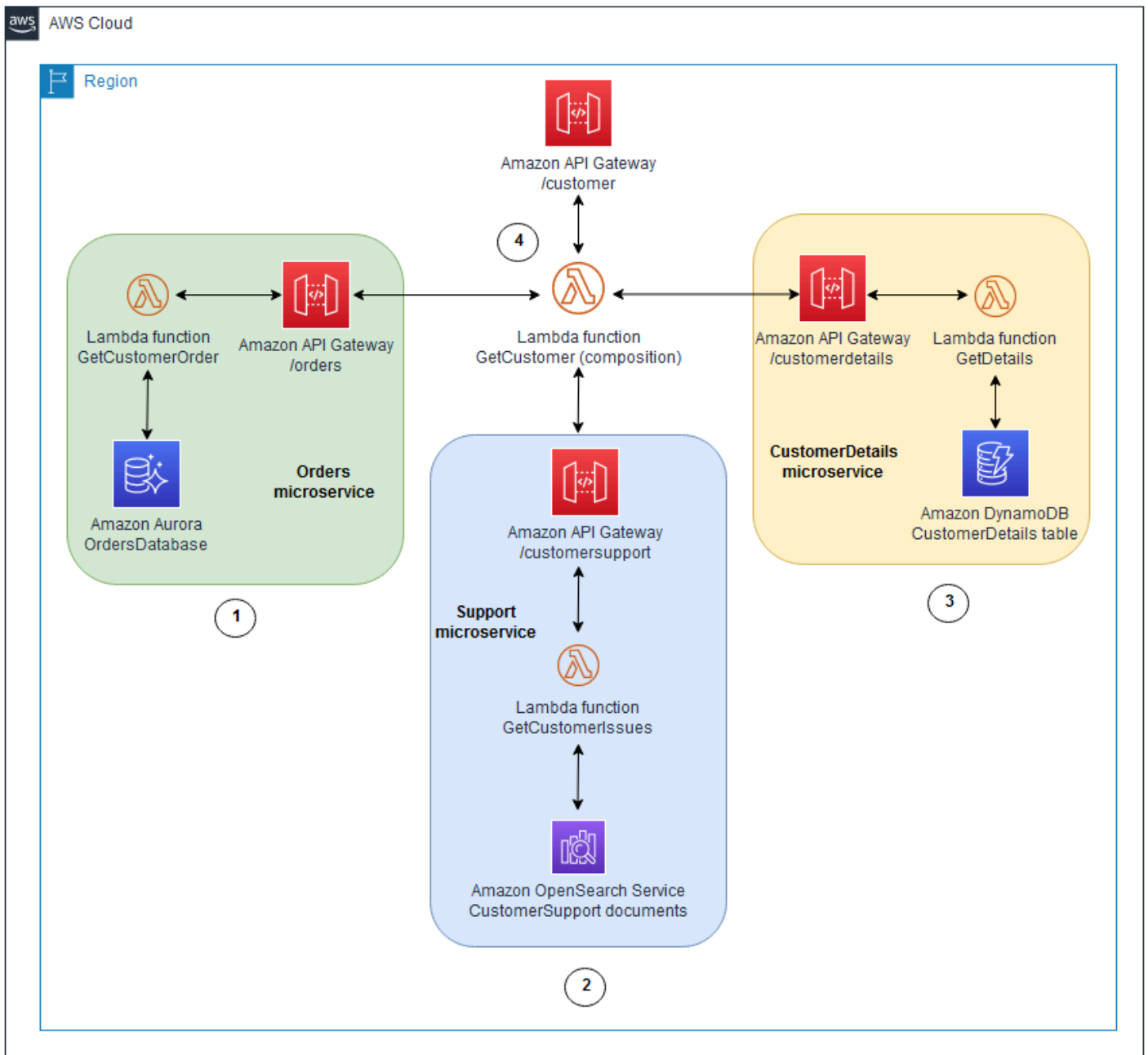
**Note**

Se você usar o database-per-service padrão, deverá implantar o [Padrão de composição da API](#) ou o [Padrão CQRS](#) para implementar consultas que abrangem vários microsserviços.

## Padrão de composição da API

Esse padrão usa um compositor ou agregador de API para implementar uma consulta ao invocar microsserviços individuais que possuem os dados. Em seguida, ele combina os resultados realizando uma junção na memória.

O diagrama a seguir ilustra como esse padrão é implementado.



O diagrama mostra o seguinte fluxo de trabalho:

1. Um gateway de API serve a API “/customer”, que tem um microserviço “Pedidos” que rastreia os pedidos dos clientes em um banco de dados Aurora.
2. O microserviço “Support” rastreia problemas de suporte ao cliente e os armazena em um banco de dados do Amazon OpenSearch Service.

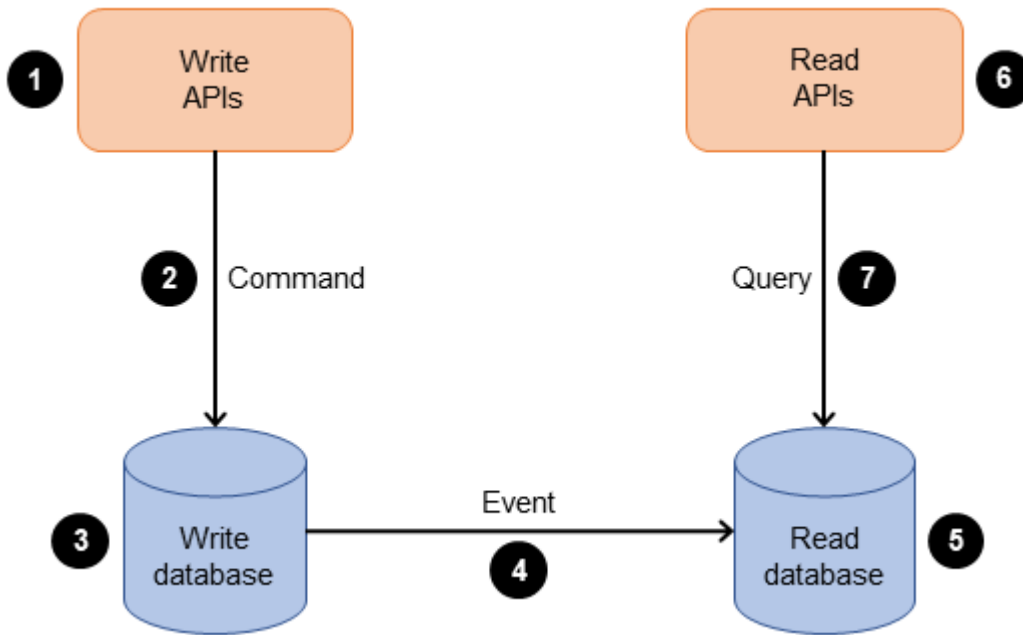
3. O microsserviço CustomerDetails "" mantém os atributos do cliente (por exemplo, endereço, número de telefone ou detalhes de pagamento) em uma tabela do DynamoDB.
4. A função “GetCustomer” Lambda executa as APIs desses microsserviços e realiza uma junção na memória dos dados antes de devolvê-los ao solicitante. Isso ajuda a recuperar facilmente as informações do cliente em uma chamada de rede para a API voltada para o usuário e mantém a interface muito simples.

O padrão de composição da API oferece a maneira mais simples de coletar dados de vários microsserviços. No entanto, há as seguintes desvantagens em usar o padrão de composição da API:

- Talvez não seja adequado para consultas complexas e grandes conjuntos de dados que exigem junções na memória.
- Seu sistema geral fica menos disponível se você aumentar o número de microsserviços conectados ao compositor da API.
- O aumento das solicitações de banco de dados cria mais tráfego de rede, o que aumenta seus custos operacionais.

## Padrão CQRS

O padrão de segmentação de responsabilidade de consulta de comando (CQRS) separa a mutação de dados, ou a parte do comando de um sistema, da parte de consulta. Você pode usar o padrão CQRS para separar atualizações e consultas se elas tiverem requisitos diferentes de taxa de throughput, latência ou consistência. O padrão CQRS divide o aplicativo em duas partes — o lado do comando e o lado da consulta — conforme mostrado no diagrama a seguir. O lado do comando trata das solicitações create, update e delete. O lado da consulta executa a parte query usando as réplicas de leitura.



O diagrama mostra o seguinte processo:

1. A empresa interage com o aplicativo enviando comandos por meio de uma API. Comandos são ações como criar, atualizar ou excluir dados.
2. O aplicativo processa o comando recebido no lado do comando. Isso envolve validar, autorizar e executar a operação.
3. O aplicativo persiste os dados do comando no banco de dados de gravação (comando).
4. Depois que o comando é armazenado no banco de dados de gravação, os eventos são acionados para atualizar os dados no banco de dados de leitura (consulta).
5. O banco de dados de leitura (consulta) processa e persiste os dados. Os bancos de dados de leitura foram projetados para serem otimizados para requisitos específicos de consulta.
6. A empresa interage com as APIs de leitura para enviar consultas ao lado da consulta do aplicativo.
7. O aplicativo processa a consulta recebida no lado da consulta e recupera os dados do banco de dados lido.

Você pode implementar o padrão CQRS usando várias combinações de bancos de dados, incluindo:

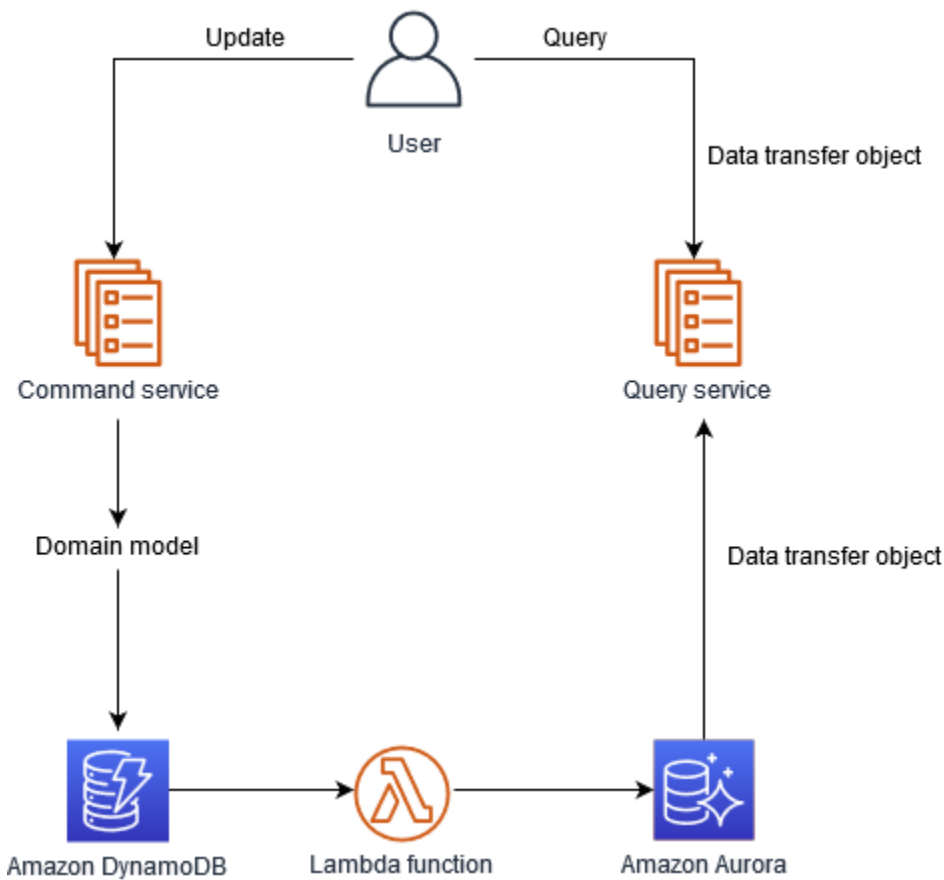
- Usando bancos de dados do sistema de gerenciamento de banco de dados relacional (RDBMS) tanto para o comando quanto para o lado da consulta. As operações de gravação vão para o

banco de dados principal e as operações de leitura podem ser roteadas para réplicas de leitura.

Exemplo: réplicas de [leitura do Amazon RDS](#)

- Usando um banco de dados RDBMS para o lado do comando e um banco de dados NoSQL para o lado da consulta. Exemplo: [modernize bancos de dados legados usando o fornecimento de eventos e o CQRS com AWS DMS](#)
- Usando bancos de dados NoSQL tanto para o comando quanto para o lado da consulta. Exemplo: [Crie uma loja de eventos CQRS com o Amazon](#) DynamoDB
- Usando um banco de dados NoSQL para o lado do comando e um banco de dados RDBMS para o lado da consulta, conforme discutido no exemplo a seguir.

Na ilustração a seguir, um armazenamento de dados NoSQL, como o DynamoDB, é usado para otimizar o throughput e fornecer capacidades de consulta flexíveis. Isso proporciona alta escalabilidade de gravação em cargas de trabalho que têm padrões de acesso bem definidos quando você adiciona dados. Um banco de dados relacional, como o Amazon Aurora, fornece uma funcionalidade de consulta complexa. Um stream do DynamoDB envia dados para uma função do Lambda que atualiza a tabela Aurora.



A implementação do padrão CQRS com o DynamoDB e o Aurora oferece os seguintes benefícios principais:

- O DynamoDB é um banco de dados NoSQL totalmente gerenciado que pode lidar com operações de gravação de alto volume, e o Aurora oferece alta escalabilidade de leitura para consultas complexas no lado da consulta.
- O DynamoDB fornece acesso aos dados de baixa latência e alta taxa de transferência, o que o torna ideal para lidar com operações de comando e atualização, e o desempenho do Aurora pode ser ajustado e otimizado para consultas complexas.
- Tanto o DynamoDB quanto o Aurora oferecem opções sem servidor, o que permite que sua empresa pague por recursos com base somente no uso.
- O DynamoDB e o Aurora são serviços totalmente gerenciados, o que reduz a carga operacional de gerenciar bancos de dados, backups e escalabilidade.

Você deve considerar o uso do padrão CQRS se:

- Você implementou o database-per-service padrão e deseja unir dados de vários microsserviços.
- Suas cargas de trabalho de leitura e gravação têm requisitos separados de escalabilidade, latência e consistência.
- A consistência eventual é aceitável para as consultas de leitura.

#### Important

O padrão CQRS normalmente resulta em uma eventual consistência entre os armazenamentos de dados.

## Padrão de fornecimento de eventos

O padrão de fornecimento de eventos é normalmente usado com o [Padrão CQRS](#) para separar as cargas de trabalho de leitura das de gravação e otimizar o desempenho, a escalabilidade e a segurança. Os dados são armazenados como uma série de eventos, em vez de atualizações diretas nos armazenamentos de dados. Os microsserviços reproduzem eventos de um armazenamento de eventos para calcular o estado apropriado de seus próprios armazenamentos de dados. O padrão fornece visibilidade ao estado atual do aplicativo e contexto adicional de como o aplicativo

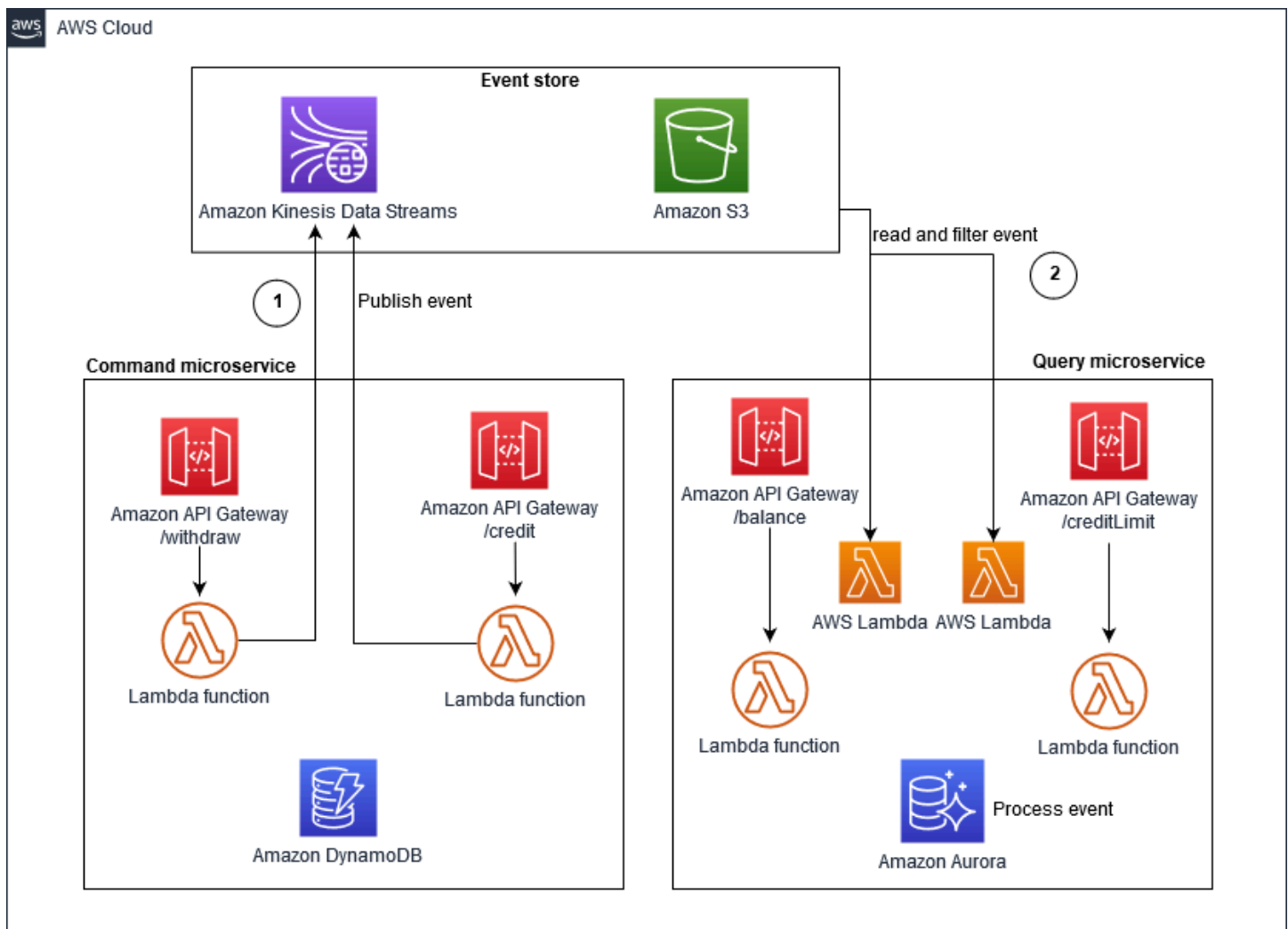
chegou a esse estado. O padrão de fornecimento de eventos funciona de forma eficaz com o padrão CQRS porque os dados podem ser reproduzidos para um evento específico, mesmo que os armazenamentos de dados de comando e consulta tenham esquemas diferentes.

Ao escolher esse padrão, você pode identificar e reconstruir o estado do aplicativo em qualquer momento. Isso produz uma trilha de auditoria persistente e facilita a depuração. No entanto, os dados acabam se tornando consistentes e isso pode não ser apropriado para alguns casos de uso.

Esse padrão pode ser implementado usando o Amazon Kinesis Data Streams EventBridge ou o Amazon.

## Implementação do Amazon Kinesis Data Streams

Na ilustração a seguir, o Kinesis Data Streams é o principal componente de um armazenamento de eventos centralizado. O armazenamento de eventos captura alterações da aplicação como eventos e as persiste no Amazon Simple Storage Service (Amazon S3).



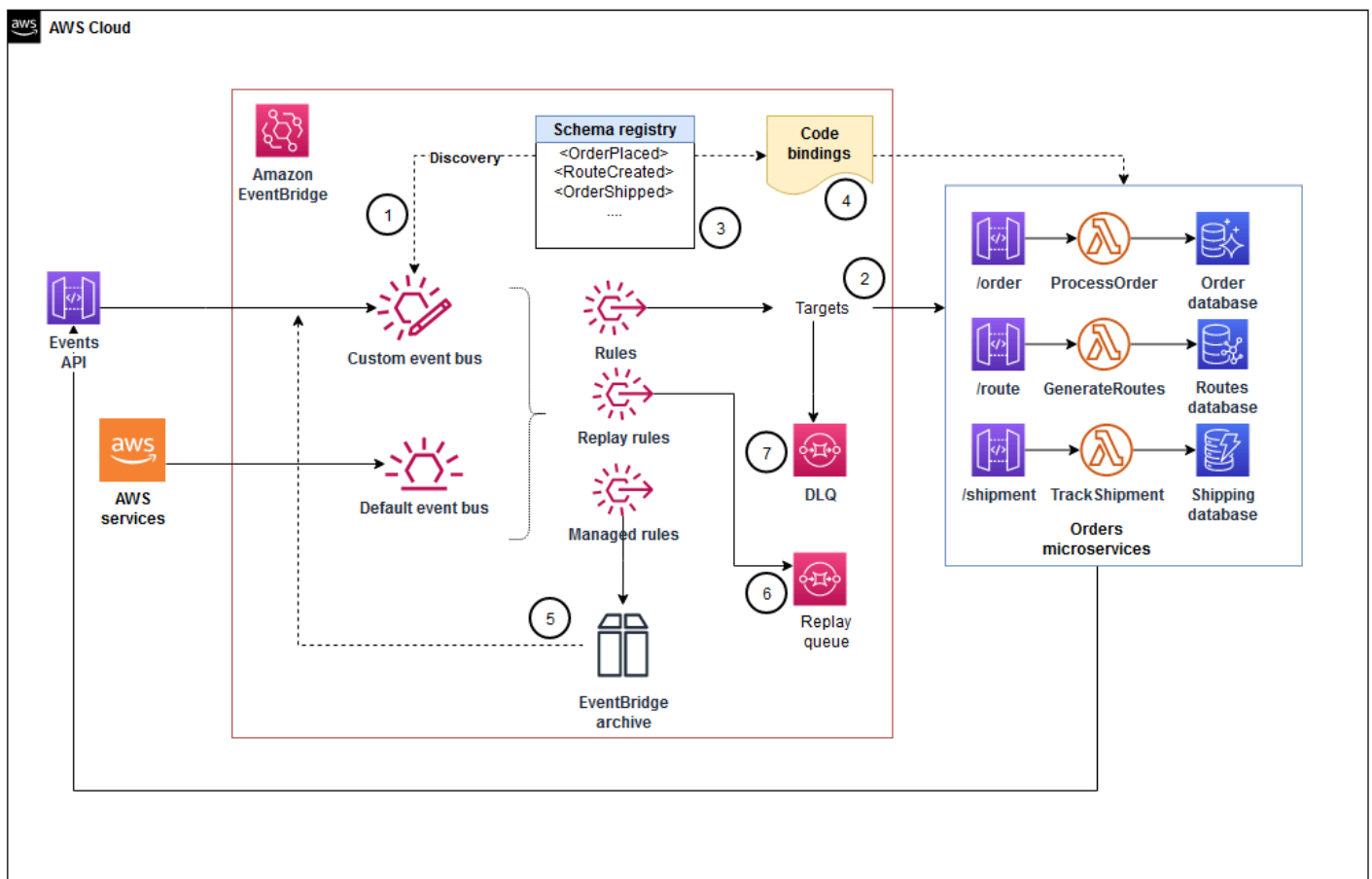


O fluxo de trabalho consiste nas seguintes etapas:

1. Quando os microsserviços “/withdraw” ou “/credit” sofrem uma mudança no estado do evento, eles publicam um evento gravando uma mensagem no Kinesis Data Streams.
2. Outros microsserviços, como “/balance” ou “/creditLimit”, leem uma cópia da mensagem, filtram por relevância e a encaminham para processamento adicional.

## EventBridge Implementação da Amazon

A arquitetura na ilustração a seguir usa EventBridge. EventBridge é um serviço sem servidor que usa eventos para conectar componentes do aplicativo, o que facilita a criação de aplicativos escaláveis e orientados por eventos. A arquitetura orientada a eventos é um estilo de criar sistemas de software fracamente acoplados que funcionam juntos emitindo e respondendo a eventos. EventBridge fornece um [barramento de eventos padrão](#) para eventos publicados por AWS serviços, e você também pode criar um barramento de [eventos personalizado para barramentos](#) específicos do domínio.




O fluxo de trabalho consiste nas seguintes etapas:

1. Os eventos "OrderPlaced" são publicados pelo microsserviço "Pedidos" no barramento de eventos personalizado.
2. Os microsserviços que precisam agir após a realização de um pedido, como o microsserviço "route", são iniciados por regras e metas.
3. Esses microsserviços geram uma rota para enviar o pedido ao cliente e emitir um evento "RouteCreated".
4. Os microsserviços que precisam tomar medidas adicionais também são iniciados pelo evento "RouteCreated".
5. Os eventos são enviados para um arquivo de eventos (por exemplo, EventBridge arquivo) para que possam ser reproduzidos para reprocessamento, se necessário.
6. Eventos históricos de pedidos são enviados para uma nova fila do Amazon SQS (fila de repetição) para reprocessamento, se necessário.
7. Se os alvos não forem iniciados, os eventos afetados serão colocados em uma fila de mensagens não entregues (DLQ) para análise e reprocessamento adicionais.

Você deve considerar o uso desse padrão se:

- Os eventos forem usados para reconstruir completamente o estado do aplicativo.
- Você exige que os eventos sejam reproduzidos no sistema e que o estado de um aplicativo possa ser determinado a qualquer momento.
- Você desejar poder reverter eventos específicos sem precisar começar com um estado de aplicativo em branco.
- Seu sistema requerer um fluxo de eventos que possa ser facilmente serializado para criar um log automatizado.
- Seu sistema exigir operações de leitura pesadas, mas for leve em operações de gravação; operações de leitura pesadas podem ser direcionadas para um banco de dados na memória que é mantido atualizado com o fluxo de eventos.

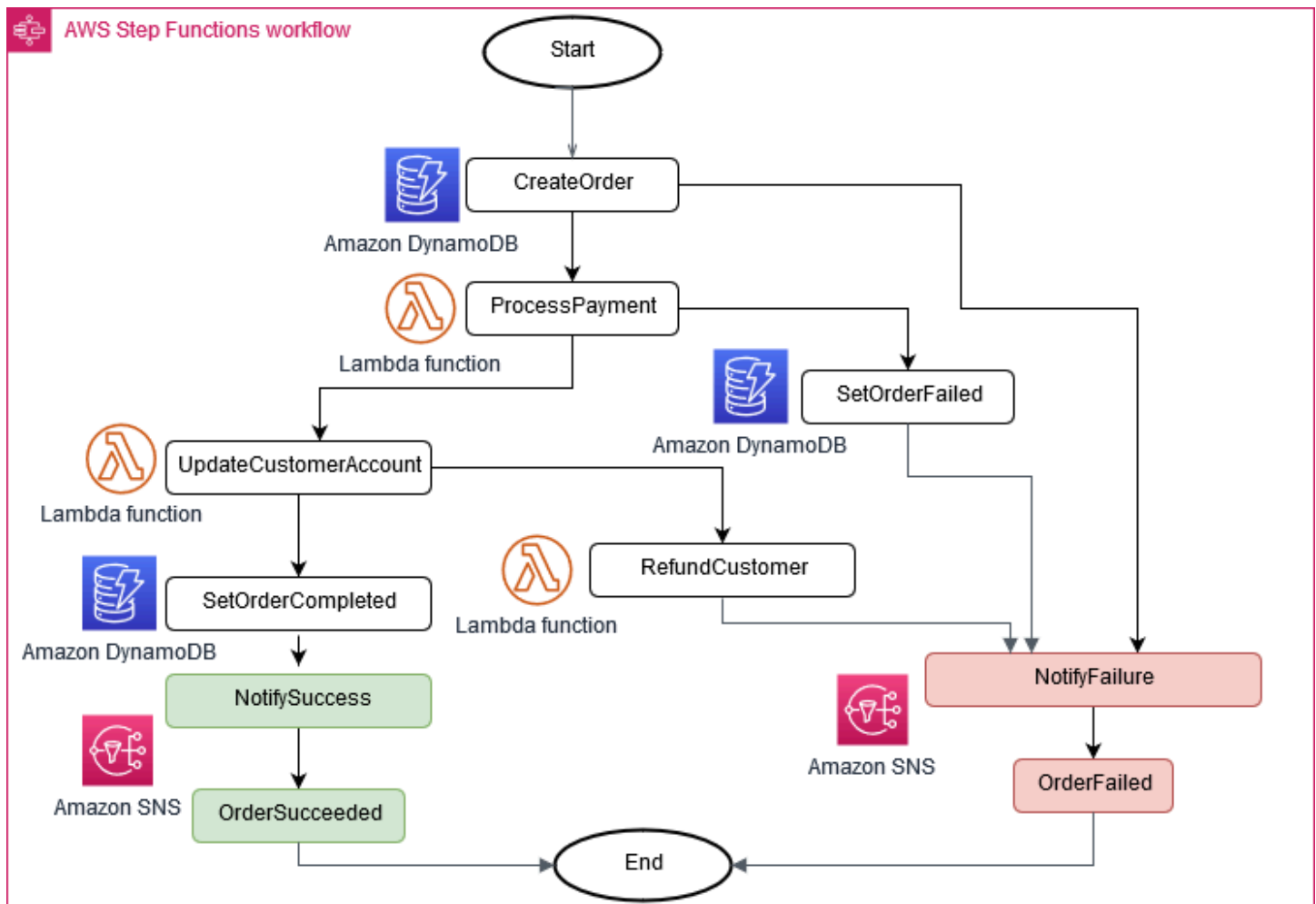
 Important

Se você usar o padrão de fornecimento de eventos, deverá implantar o [Padrão Saga](#) para manter a consistência de dados nos microsserviços.

# Padrão Saga

O padrão saga é um padrão de gerenciamento de falhas que ajuda a estabelecer a consistência em aplicativos distribuídos e coordena as transações entre vários microsserviços para manter a consistência de dados. Um microsserviço publica um evento para cada transação e a próxima transação é iniciada com base no resultado do evento. Ele pode seguir dois caminhos diferentes, dependendo do êxito ou da falha das transações.

A ilustração a seguir mostra como o padrão saga implementa um sistema de processamento de pedidos usando AWS Step Functions. Cada etapa (por exemplo, "ProcessPayment") também tem etapas separadas para lidar com o sucesso (por exemplo, "UpdateCustomerAccount") ou a falha (por exemplo, "SetOrderFailure") do processo.



Você deve considerar o uso desse padrão se:

- O aplicativo precisar manter a consistência de dados em vários microsserviços sem um acoplamento forte.
- Houver transações de longa duração e você não quiser que outros microsserviços sejam bloqueados se um microsserviço for executado por muito tempo.
- Você precisar ser capaz de reverter se uma operação falhar na sequência.

#### Important

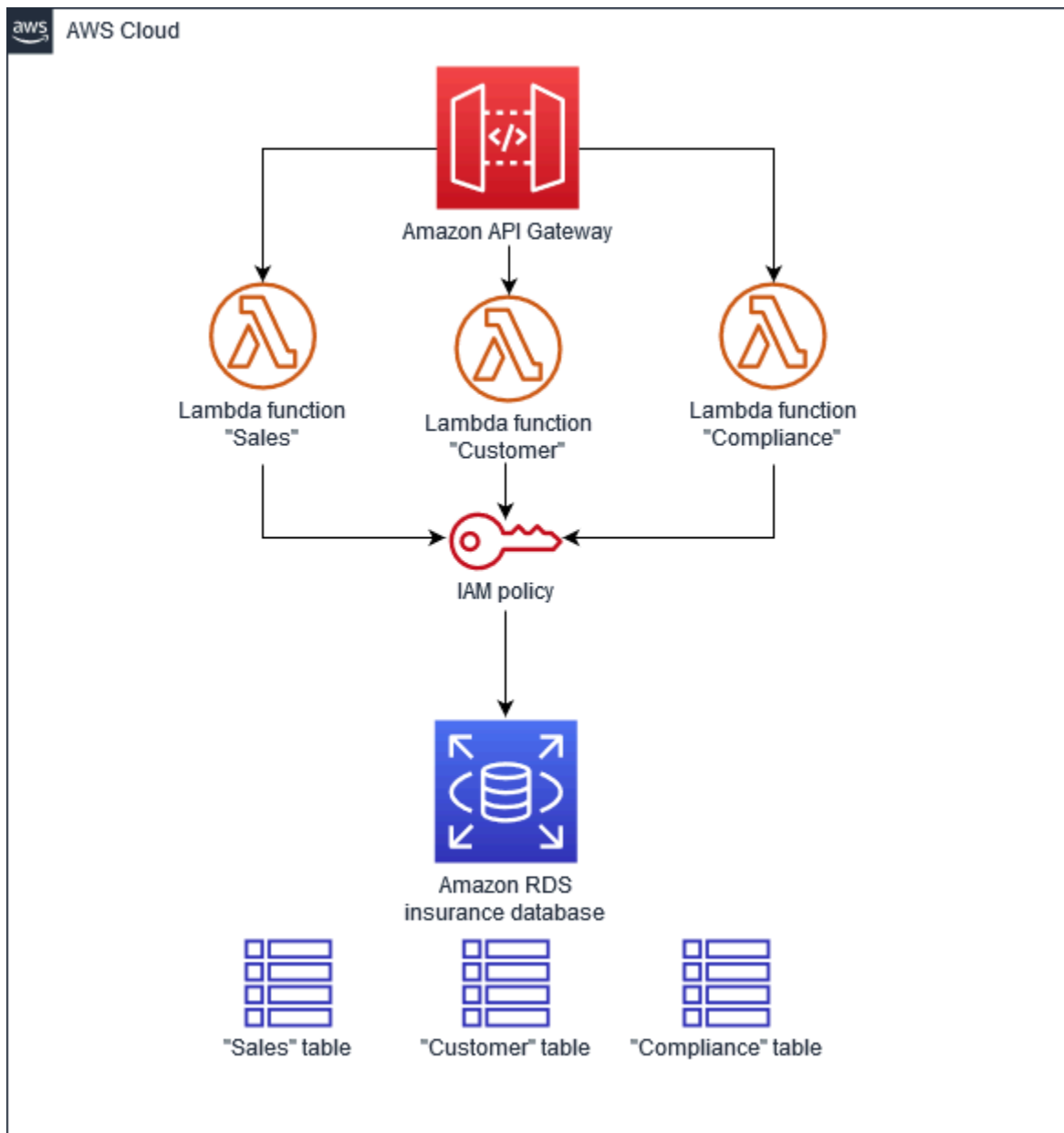
O padrão saga é difícil de depurar e sua complexidade aumenta com o número de microsserviços. O padrão requer um modelo de programação complexo que desenvolva e projete transações compensatórias para reverter e desfazer mudanças.

Para obter mais informações sobre a implementação do padrão saga em uma arquitetura de microsserviços, consulte o padrão [Implementar o padrão saga sem servidor usando o AWS Step Functions](#) no site de Orientação prescritiva da AWS.

## shared-database-per-service Padrão S

No shared-database-per-service padrão, o mesmo banco de dados é compartilhado por vários microsserviços. Você precisa avaliar cuidadosamente a arquitetura do aplicativo antes de adotar esse padrão e evitar tabelas dinâmicas (tabelas únicas que são compartilhadas entre vários microsserviços). Todas as alterações no banco de dados também devem ser compatíveis com versões anteriores; por exemplo, os desenvolvedores só podem descartar colunas ou tabelas se os objetos não forem referenciados pelos versionamentos atuais e anteriores de todos os microsserviços.

Na ilustração a seguir, um banco de dados de seguros é compartilhado por todos os microsserviços e uma política do IAM fornece acesso ao banco de dados. Isso cria um acoplamento do tempo de desenvolvimento. Por exemplo, uma mudança no microsserviço “Vendas” precisa coordenar as mudanças de esquema com o microsserviço “Cliente”. Esse padrão não reduz as dependências entre as equipes de desenvolvimento e introduz o acoplamento de runtime porque todos os microsserviços compartilham o mesmo banco de dados. Por exemplo, transações de “Vendas” de longa duração podem bloquear a tabela “Cliente” e isso bloqueia as transações de “Cliente”.



Você deve considerar o uso desse padrão se:

- Você não quiser muita refatoração da sua base de código existente.
- Você impuser consistência de dados usando transações que fornecem atomicidade, consistência, isolamento e durabilidade (ACID).
- Você deseja manter e operar somente um banco de dados.
- Implementar o database-per-service padrão é difícil devido às interdependências entre seus microsserviços existentes.
- Você não quiser redesenhar completamente sua camada de dados existente.

## Perguntas frequentes

Esta seção fornece respostas às perguntas mais comuns sobre a habilitação de persistência de dados em microsserviços.

### Quando posso modernizar meu banco de dados monolítico como parte da minha jornada de modernização?

Você deve se concentrar em modernizar seu banco de dados monolítico ao começar a decompor aplicativos monolíticos em microsserviços. Certifique-se de criar uma estratégia para dividir seu banco de dados em vários bancos de dados pequenos que estejam alinhados com seus aplicativos.

### Posso manter um banco de dados monolítico legado para vários microsserviços?

Manter um banco de dados monolítico compartilhado para vários microsserviços cria um acoplamento forte, o que significa que você não pode implantar alterações de forma independente nos seus microsserviços e que todas as alterações de esquema devem ser coordenadas entre seus microsserviços. Embora você possa usar um armazenamento de dados relacionais como seu banco de dados monolítico, os bancos de dados NoSQL podem ser uma escolha melhor para alguns dos seus microsserviços.

### O que devo considerar ao projetar bancos de dados para uma arquitetura de microsserviços?

Você deve projetar seu aplicativo com base em domínios que se alinhem à funcionalidade do seu aplicativo. Certifique-se de avaliar a funcionalidade do aplicativo e decidir se ele requer um esquema de banco de dados relacional. Você também deve considerar o uso de um banco de dados NoSQL, se ele atender às suas necessidades.

### Qual é o padrão comum para manter a consistência de dados em diferentes microsserviços?

O padrão mais comum é usar uma [arquitetura orientada a eventos](#).

## Como faço para manter a automação de transações?

Em uma arquitetura de microsserviços, uma transação consiste em várias transações locais gerenciadas por diferentes microsserviços. Se uma transação local falhar, você precisará reverter as transações bem-sucedidas concluídas anteriormente. Você pode usar a [Padrão Saga](#) para evitar isso.

## Preciso usar um banco de dados separado para cada microsserviço?

A principal vantagem de uma arquitetura de microsserviços é o acoplamento fraco. Os dados persistentes de cada microsserviço devem ser mantidos privados e acessíveis somente por meio da API de um microsserviço. As alterações no esquema de dados devem ser cuidadosamente avaliadas se seus microsserviços compartilharem o mesmo banco de dados.

## Como posso manter a privacidade dos dados persistentes de um microsserviço se todos compartilham um único banco de dados?

Se seus microsserviços compartilham um banco de dados relacional, verifique se você tem tabelas privadas para cada microsserviço. Você também pode criar esquemas individuais que sejam privados para os microsserviços individuais.

# Recursos

## Guias e padrões relacionados

- [Estratégia para modernizar aplicações na AWS Cloud](#)
- [Abordagem em fases para modernizar aplicações na AWS Cloud](#)
- [Avaliando a preparação para modernizar aplicações na AWS Cloud](#)
- [Decompor monólitos em microsserviços](#)
- [Integrando microsserviços usando serviços AWS com tecnologia sem servidor](#)
- [Implementar o padrão saga sem servidor usando o AWS Step Functions](#)

## Outros recursos

- [Modernização de aplicativos com o AWS](#)
- [Criar microsserviços altamente disponíveis para potencializar aplicativos de qualquer tamanho e escala](#)
- [Modernização de aplicações nativas de nuvem com o AWS](#)
- [Otimização de custos e inovação: uma introdução à modernização de aplicações](#)
- [Guia do desenvolvedor: escale com microsserviços](#)
- [Gerenciamento distribuído de dados – Padrão do Saga](#)
- [Implementação de arquiteturas de microsserviços usando serviços da AWS: padrão de segmentação de responsabilidade por consulta de comando](#)
- [Implementação de arquiteturas de microsserviços usando serviços da AWS: padrão de fornecimento de eventos](#)
- [Aplicativos modernos: criação de valor por meio do design de aplicativos](#)
- [Modernize seus aplicativos, impulse o crescimento e reduza o TCO](#)



## 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
<a href="#">Padrão atualizado</a>	Atualizamos a seção de <a href="#">EventBridge implementação do padrão de fornecimento de eventos na Amazon</a> .	4 de dezembro de 2023
<a href="#">Seção expandida</a>	Atualizamos o <a href="#">padrão CQRS</a> com mais informações.	17 de novembro de 2023
<a href="#">Foi adicionado um link para implementar o padrão da saga com Funções em Etapas</a>	Atualizamos as seções <a href="#">Início</a> e <a href="#">Padrão da Saga</a> com o link para o padrão <a href="#">Implementar o padrão saga sem servidor usando o AWS Step Functions</a> no site <a href="#">Recomendações da AWS</a> .	23 de fevereiro de 2021
<a href="#">Publicação inicial</a>	—	27 de janeiro de 2021

# Glossário de Recomendações da AWS

Os termos a seguir são comumente usados em estratégias, guias e padrões fornecidos pelas Recomendações da AWS. 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:

- **Refatorar/rearquitetar:** mova uma aplicação e modifique sua arquitetura aproveitando ao máximo os recursos nativos de nuvem para melhorar a agilidade, a performance e a escalabilidade. Isso normalmente envolve a portabilidade do sistema operacional e do banco de dados. Exemplo: migrar seu banco de dados Oracle on-premises para o Amazon Aurora Edição Compatível com PostgreSQL.
- **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 AWS Cloud.
- **Recomprar (drop and shop):** mude para um produto diferente, normalmente migrando de uma licença tradicional para um modelo SaaS. Exemplo: migrar 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 AWS Cloud.
- **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. Esse cenário de migração é específico para o VMware Cloud na AWS, o qual oferece suporte a compatibilidade de máquina virtual (VM) e portabilidade de workloads entre seu ambiente on-premises e a AWS. É possível usar as tecnologias VMware Cloud Foundation de seus datacenters on-premises ao migrar sua infraestrutura para o VMware Cloud na AWS. Exemplo: realocar o hipervisor que hospeda seu banco de dados Oracle para o VMware Cloud na 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

### ABAC

Consulte controle de [acesso baseado em atributos](#).

serviços abstratos

Veja os [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 no qual 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.

função agregada

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 a [inteligência artificial](#).

## AIOps

Veja as [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 aplicativos

Uma abordagem de segurança que permite o uso somente de aplicativos aprovados 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 descryptografia. É possível compartilhar a chave pública porque ela não é usada na descryptografia, 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 para AWS](#) na documentação do 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 em uma Região da AWS que é isolado das falhas em outras zonas de disponibilidade e fornece conectividade de rede de baixa latência e baixo custo para outras zonas de disponibilidade na mesma região.

## AWS Cloud Adoption Framework (AWS CAF)

Uma estrutura de diretrizes e práticas recomendadas da AWS para ajudar as organizações a desenvolverem um plano eficiente e eficaz para migrar com sucesso para a nuvem. AWS 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. Para essa perspectiva, a 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 Workload Qualification Framework (AWS WQF)

Uma ferramenta que avalia os workloads de migração do banco de dados, recomenda estratégias de migração e fornece estimativas de trabalho. A WQF é fornecida com o 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

### BCP

Veja o [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.

### 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 em vidro quebrado

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 vidro na orientação do Well-ArchitectedAWS](#).

#### 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 containerizados 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

Consulte [Estrutura de adoção da AWS nuvem](#).

## CCoE

Veja o [Centro de Excelência em Nuvem](#).

## CDC

Veja [a captura de dados de alterações](#).

### 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 [a integração e a 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.

## criptografia no lado do cliente

Criptografia de dados feita localmente, antes que o AWS service (Serviço da AWS) de destino os 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 recomendadas 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 [publicações da CCoE](#) no blog de Estratégia Empresarial na AWS Cloud.



## computação em nuvem

A tecnologia de nuvem normalmente usada para armazenamento de dados remoto e gerenciamento de dispositivos de IoT. A computação em nuvem geralmente está conectada à tecnologia de [computação de ponta](#).

## 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 AWS Cloud:

- 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
- Reinvenção: otimizar produtos e serviços e inovar na nuvem

Esses estágios foram definidos por Stephen Orban na publicação no blog [A jornada rumo à nuvem em primeiro lugar e os estágios da adoção](#) no Blog de estratégia empresarial na AWS Cloud. Para obter informações sobre como eles se relacionam com a estratégia de migração da AWS, consulte o [guia de preparação para migração](#).

## CMDB

Consulte o [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 GitHub ou AWS CodeCommit. 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 pipeline de CI/CD 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

Um campo da IA utilizado por máquinas para identificar pessoas, lugares e coisas em imagens com precisão igual ou superior aos níveis humanos. Geralmente construído com modelos de aprendizado profundo, ele automatiza a extração, análise, classificação e compreensão de informações úteis de uma única imagem ou sequência de imagens.

## 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 regras e ações de remediação do AWS Config que você pode montar para personalizar suas verificações de conformidade e segurança. É possível implantar um pacote de conformidade como uma entidade única em uma região e uma Conta da AWS ou em toda a organização usando um modelo YAML. Para obter mais informações, consulte [Pacotes de conformidade](#) na documentação do 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. O CI/CD é comumente descrito como um pipeline. O CI/CD pode ajudar você 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](#).

## 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 no AWS Well-Architected Framework. 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.

### 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 que oferece suporte à inteligência comercial, como análises. 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

Consulte a [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.

## defense-in-depth

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 na AWS, você adiciona vários controles em diferentes camadas da estrutura

do AWS Organizations para ajudar a proteger os recursos. Por exemplo, uma *defense-in-depth* abordagem pode combinar autenticação multifatorial, segmentação de rede e criptografia.

### administrador delegado

No AWS Organizations, um serviço compatível pode registrar uma conta-membro da AWS para administrar as contas da organização e gerenciar permissões para esse 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 o [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 são comumente 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 (DR)

A estratégia e o processo que você usa para minimizar o tempo de inatividade e a perda de dados causados por um [desastre](#). Para obter mais informações, consulte [Recuperação de desastres em workloads na AWS: Recuperação na Nuvem](#) na Documentação do Well-Architected Framework AWS.

## DML

Veja a [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, Design orientado por domínio: lidando com a complexidade no coração do software (Boston: Addison-Wesley Professional, 2003). Para obter informações sobre como usar o design orientado por domínio com o padrão strangler fig, consulte [Modernizar incrementalmente os serviços web herdados do Microsoft ASP.NET \(ASMX\) usando contêineres e o Amazon API Gateway](#).

## DR

Veja a [recuperação de desastres](#).

## detecção de deriva

Rastreamento de desvios de uma configuração básica. 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 o [mapeamento do fluxo de valor do desenvolvimento](#).

## E

### EDA

Veja a [análise exploratória de dados](#).

### computação de borda

A tecnologia que aumenta o poder computacional de dispositivos inteligentes nas bordas de uma rede de IoT. Quando comparada à [computação em nuvem](#), a computação de ponta pode reduzir a latência da comunicação e melhorar o tempo de resposta.

### 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. Os sistemas big-endian armazenam o byte mais significativo antes. Os sistemas little-endian armazenam o byte menos significativo antes.

### endpoint

Veja o [endpoint do serviço](#).

### serviço de endpoint

Um serviço que pode ser hospedado em uma nuvem privada virtual (VPC) para ser compartilhado com outros usuários. É possível criar um serviço de endpoint com o AWS PrivateLink e conceder permissões a outras Contas da AWS ou a entidades principais do 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).

### criptografia envelopada

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

### environment (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 pipeline de CI/CD, 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 epics de segurança da AWS CAF incluem gerenciamento de identidade e acesso, controles detectivos, 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](#).

### 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](#). Ele armazena dados quantitativos sobre as operações comerciais. Normalmente, uma tabela de fatos contém dois tipos de colunas: aquelas que contêm medidas e aquelas que contêm uma chave externa para uma tabela de dimensões.

### falham rapidamente

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

### limite de isolamento de falhas

NoNuvem 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 [Limites de isolamento de AWS falhas](#).

### ramificação de recursos

Veja a [filial](#).

### 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.

## FGAC

Veja o [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 por meio da [captura de dados alterados](#) para migrar dados no menor tempo possível, em vez de usar uma abordagem em fases. O objetivo é reduzir ao mínimo o tempo de inatividade.

## G

bloqueio geográfico

Veja as [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 baseado em troncos](#) é a abordagem moderna e preferida.

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, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector e verificações personalizadas AWS Lambda.

## H

### HA

Veja a [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 HA 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.

#### 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.

## I

### IaC

Veja a [infraestrutura como código](#).

### Política baseada em identidade

Uma política associada a uma ou mais entidades principais do IAM que define suas permissões dentro do ambiente da Nuvem AWS.

### 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 a [Internet das Coisas industrial](#).

### infraestrutura imutável

Um modelo que implanta uma nova infraestrutura para cargas de trabalho de produção em vez de atualizar, corrigir ou modificar a infraestrutura existente. [Infraestruturas imutáveis são inerentemente mais consistentes, confiáveis e previsíveis do que infraestruturas mutáveis](#). Para

obter mais informações, consulte as melhores práticas de [implantação usando infraestrutura imutável](#) no Well-Architected AWS Framework.

### VPC de entrada (admissão)

Em uma arquitetura de várias contas da AWS, uma VPC que aceita, inspeciona e roteia conexões de rede de fora de uma aplicação. 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.

### 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 várias contas da AWS, uma VPC centralizada que gerencia as inspeções do tráfego de rede entre VPCs (na mesma ou em diferentes Regiões da AWS), na Internet e em redes on-premises. 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 machine learning com a AWS](#).

## IoT

Consulte [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

Consulte [a biblioteca de informações](#) de TI.

## ITSM

Veja o [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 zona de pouso é um ambiente da AWS com várias contas que é bem arquitetado, 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](#).

## migração de grande porte

Uma migração de 300 servidores ou mais.

## LBAC

Veja controle de [acesso baseado em rótulos](#).

## 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](#).

## ambientes inferiores

Veja o [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 a [filial](#).

## 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 abstratos.

## MAP

Consulte [Migration Acceleration Program](#).

## mecanismo

Um processo completo no qual 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 [Construindo mecanismos](#) no AWS Well-Architected Framework.

## conta-membro

Todas as Contas da AWS, exceto a conta de gerenciamento que faz parte de uma organização no AWS Organizations. Uma conta só pode ser membro de uma organização de cada vez.

## 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 [Integrar microsserviços usando serviços da AWS sem 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 [Implementar microsserviços na AWS](#).

## Programa de Aceleração da Migração (MAP)

Um programa da AWS 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

Equipes multifuncionais que simplificam a migração de workloads 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 de destino, o grupo de segurança e conta da AWS.

## 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: redefina a hospedagem da migração para o Amazon EC2 com o Application Migration Service da AWS.

## Avaliação de Portfólio para Migração (MPA)

Uma ferramenta online que fornece informações para validar o caso de negócios para migrar para a AWS Cloud. 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 de MPA](#) (login necessário) está disponível gratuitamente para todos os consultores da AWS e consultores parceiros da APN.

## Avaliação de Preparação para Migração (MRA)

O processo de obter insights sobre o status de preparação de uma organização para a nuvem, identificando pontos fortes e fracos e criar um plano de ação para fechar as lacunas identificadas usando a AWS CAF. 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 um workload para a AWS Cloud. Para obter mais informações, consulte a entrada de [7 Rs](#) neste glossário e consulte [Mobilize sua organização para acelerar migrações em grande escala](#).

### ML

Veja o [aprendizado de máquina](#).

### MAPA

Consulte [Avaliação do portfólio de migração](#).

### 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 [Estratégia para modernizar aplicações na AWS Cloud](#).

### 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 [Avaliar a preparação para modernização de aplicações na AWS Cloud](#).

### 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

crece. Para resolver esses problemas, é possível criar uma arquitetura de microsserviços. Para obter mais informações, consulte [Decompor monólitos em microsserviços](#).

### 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 cargas de trabalho de produção. Para melhorar a consistência, confiabilidade e previsibilidade, o AWS Well-Architected Framework recomenda o uso de infraestrutura [imutável](#) como uma prática recomendada.

## O

### OAC

Veja o [controle de acesso de origem](#).

### CARVALHO

Veja a [identidade de acesso de origem](#).

### OCM

Veja o [gerenciamento de mudanças 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 a [integração de operações](#).

### OLA

Veja o [contrato em 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.

#### 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 melhores práticas 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 \(ORR\)](#) no Well-Architected AWS Framework.

#### 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 pelo AWS CloudTrail que registra todos os eventos para todas as Contas da AWS em uma organização no 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 migração para a AWS, essa estrutura é chamada de aceleração de pessoas devido à velocidade das alterações exigida 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 em todas as Regiões da AWS, criptografia do lado do servidor com o AWS KMS (SSE-KMS) e solicitações PUT e DELETE dinâmicas para o 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.

OU

Veja a [análise de prontidão operacional](#).

## VPC de saída (egresso)

Em uma arquitetura de várias contas da AWS, uma VPC que lida com conexões de rede que são iniciadas de dentro de uma aplicação. 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 as [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.

## política

Um objeto que pode definir permissões (consulte a [política baseada em identidade](#)), especificar as condições de acesso (consulte a [política baseada em recursos](#)) ou definir as permissões máximas para todas as contas em uma organização em AWS Organizations (consulte a 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. Para obter mais informações, consulte [Habilitar a persistência de dados em microsserviços](#).

## 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 WHERE cláusula.

## pressão de predicados

Uma técnica de otimização de consulta 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 o desempenho 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.

## entidade principal

Entidade na AWS que pode executar ações e acessar recursos. Essa entidade geralmente é um usuário raiz de uma Conta da AWS, um perfil 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 engenharia.

### 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](#) projetado para impedir a implantação de recursos não compatíveis. Esses controles examinam 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.

### ambiente de produção

Veja o [ambiente](#).

### 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.

## 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, responsável, consultado, informado \(RACI\)](#).

### 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, responsável, consultado, informado \(RACI\)](#).

### RCAC

Veja o [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.

### rearquiteta

Veja [7 Rs](#).

### objetivo de ponto de recuperação (RPO)

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

### refatorar

Veja [7 Rs](#).



## região

Uma coleção de recursos da AWS em uma área geográfica. Cada Região da AWS é isolada e independente das demais para fornecer tolerância a falhas, estabilidade e resiliência. Para obter mais informações, consulte [Gerenciar Regiões da AWS](#) na Referência geral da AWS.

## 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](#).

## 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](#).

## aposentar-se

Veja [7 Rs](#).

## rotação

O processo de atualizar periodicamente um [segredo](#) para dificultar o acesso das credenciais por um invasor.

## 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 o [objetivo do ponto de recuperação](#).

## RTO

Veja o [objetivo do 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 a autenticação única (SSO) federada para que os usuários possam fazer login no AWS

Management Console ou chamar as operações de API da AWS 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.

## SCP

Veja a [política de controle de serviços](#).

## secret

Em AWS Secrets Manager, informações confidenciais ou restritas, como uma senha ou credenciais de usuário, que você armazena de forma criptografada. Ele 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 [Secret](#) na documentação do Secrets Manager.

## 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 principais de controles de segurança: preventivos, detectivos, responsivos e proativos.](#)

## fortalecimento 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 correção de uma instância do Amazon EC2 ou a rotação de credenciais.

## Criptografia do lado do servidor

A criptografia dos dados no destino pelo AWS service (Serviço da AWS) que 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 serviços](#) na documentação do AWS Organizations.

## service endpoint (endpoint de serviço)

O URL do ponto de entrada de 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 medida de um aspecto de desempenho de um serviço, como taxa de erro, disponibilidade ou taxa de transferência.

## objetivo de nível de serviço (SLO)

Uma métrica alvo que representa a integridade de um serviço, conforme medida por um indicador de [nível de serviço](#).

## modelo de responsabilidade compartilhada

Um modelo que descreve a responsabilidade que você compartilha com a AWS em questões de segurança e conformidade na nuvem. A 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](#).

## SIEM

Veja [informações de segurança e sistema de gerenciamento de eventos](#).

## ponto único de falha (SPOF)

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

## SLA

Veja o contrato [de nível de serviço](#).

## ESGUIO

Veja o indicador [de nível de serviço](#).

## SLO

Veja o objetivo do [nível de serviço](#).

## split-and-seed modelo

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 [Abordagem em fases para modernizar aplicativos no](#) Nuvem AWS

## CUSPE

Veja [um único ponto de falha](#).

## esquema de estrelas

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 uso 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 [Modernizar incrementalmente os serviços Web herdados do Microsoft ASP.NET \(ASMX\) usando contêineres e o Amazon API Gateway](#).

## sub-rede

Um intervalo de endereços IP na VPC. Uma sub-rede deve residir em uma única zona de disponibilidade.

## 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 o desempenho. Você pode usar o [Amazon CloudWatch Synthetics](#) para criar esses testes.

# T

## tags

Pares de valor chave que atuam como metadados para organizar seus recursos do AWS. As tags podem ajudar você a gerenciar, identificar, organizar, pesquisar e filtrar recursos. 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 o [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.

### 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 documentação do AWS Transit Gateway.

### 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 no 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 [Como usar o AWS Organizations com outros serviços da AWS](#) na documentação do AWS Organizations.

### 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. Para obter mais informações, consulte o guia [Como quantificar a incerteza em sistemas de aprendizado profundo](#).

## 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 o [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 back-end.

## 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.

## MINHOCA

Veja [escrever uma vez, ler muitas](#).

## WQF

Consulte o [AWS Workload Qualification Framework](#).

## escreva uma vez, leia muitas (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, geralmente malware, que tira proveito de uma vulnerabilidade de [dia zero](#).

## 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.

## 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.