



O teste CI/CD decisivo: seu pipeline é totalmente CI/CD?

AWS Orientação prescritiva



AWS Orientação prescritiva: O teste CI/CD decisivo: seu pipeline é totalmente CI/CD?

Copyright © 2025 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 os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

Introdução	1
Objetivos	1
Entendendo o CI/CD	3
Sobre a integração contínua	4
Sobre a entrega contínua	4
Testes	5
Métricas	6
Diferenças nos processos de CI/CD	8
Abordagem Gitflow	8
Abordagem baseada em troncos	10
Integridade do ambiente	11
Versões	12
Segurança	13
Teste decisivo para tubulações de CI/CD	15
Práticas recomendadas	17
Perguntas frequentes	19
Quais são alguns indicadores-chave de que meu processo de implantação não é totalmente CI/CD?	19
E se eu quiser usar um processo totalmente de CI/CD, mas ainda quiser programar lançamentos de determinados recursos para momentos específicos?	19
E se algumas etapas do meu processo de implantação não puderem ser automatizadas?	19
E se minha equipe técnica estiver mais confortável com fluxos de trabalho antigos do que com um processo totalmente de CI/CD?	20
E se meus ambientes estiverem em várias contas? Ainda posso usar um processo totalmente de CI/CD?	20
Próximas etapas	21
Recursos	22
AWS documentação e referências	22
Serviços e ferramentas	22
Histórico do documento	23
Glossário	24
#	24
A	25
B	28

C	30
D	33
E	37
F	39
G	41
H	42
eu	44
L	46
M	47
O	52
P	54
Q	57
R	58
S	61
T	65
U	66
V	67
W	67
Z	68
.....	lxx

A ferramenta CI/CD litmus test: Is your pipeline fully CI/CD?

Steven Guggenheimer e Ananya Koduri, da Amazon Web Services (AWS)

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

Seu pipeline é automatizado? É uma pergunta simples, mas muitas organizações abordam a resposta de forma muito simples. A resposta é muito mais complicada do que sim ou não.

As inovações em tecnologia estão acontecendo constantemente e, às vezes, pode ser difícil para as organizações acompanharem o ritmo. Essa coisa nova é uma moda passageira ou é a próxima grande novidade? Devo revisar minhas práticas atuais ou devo esperar? Frequentemente, quando fica claro que algo é de fato a próxima grande novidade, você pode se pegar jogando catchup. A integração contínua e a entrega contínua (CI/CD) chegaram para ficar, mas nem sempre foi assim. Muitas pessoas demoraram muito para serem convencidas, e algumas ainda precisam ser mais convencidas.

CI/CD is the process of automating the source, build, test, staging, and production stages of the software release process, and it is commonly described as a pipeline. Today, the cost savings and speed of CI/CD automações convenceram a maioria das organizações de seu valor. Mas fazer a transição para essa nova abordagem não é uma tarefa fácil. Você precisa garantir que sua equipe tenha o treinamento certo, atualizar alguns recursos e, em seguida, testar, testar e testar. Há muito o que fazer. Na maioria dos casos, você deseja fazer essas alterações gradualmente para ajudar sua organização a se adaptar.

O objetivo deste documento é definir o que significa ter um processo completo de CI/CD. Ele fornece uma ferramenta para avaliar seus próprios processos e apresenta um caminho a seguir para processos que ainda não existem. Esse caminho a seguir raramente é uma conversão da noite para o dia. Esses processos são complexos e dependem de muitos fatores, incluindo os atuais conjuntos de habilidades dos funcionários e as demandas atuais de infraestrutura. Recomendamos que você priorize e faça pequenas mudanças incrementais.

Objetivos

A seguir estão os possíveis benefícios da implementação das recomendações deste guia:

- Eficiência — Um processo de implantação totalmente CI/CD pode reduzir a complexidade, as cargas de trabalho e as incontáveis horas gastas na depuração, execução de processos manuais

e manutenção. Para obter mais informações, consulte [Benefícios da entrega contínua](#). De acordo com uma [postagem no TechAhead blog](#), a implementação do processo de CI/CD pode levar a uma economia estimada de 20% em tempo, esforço e recursos.

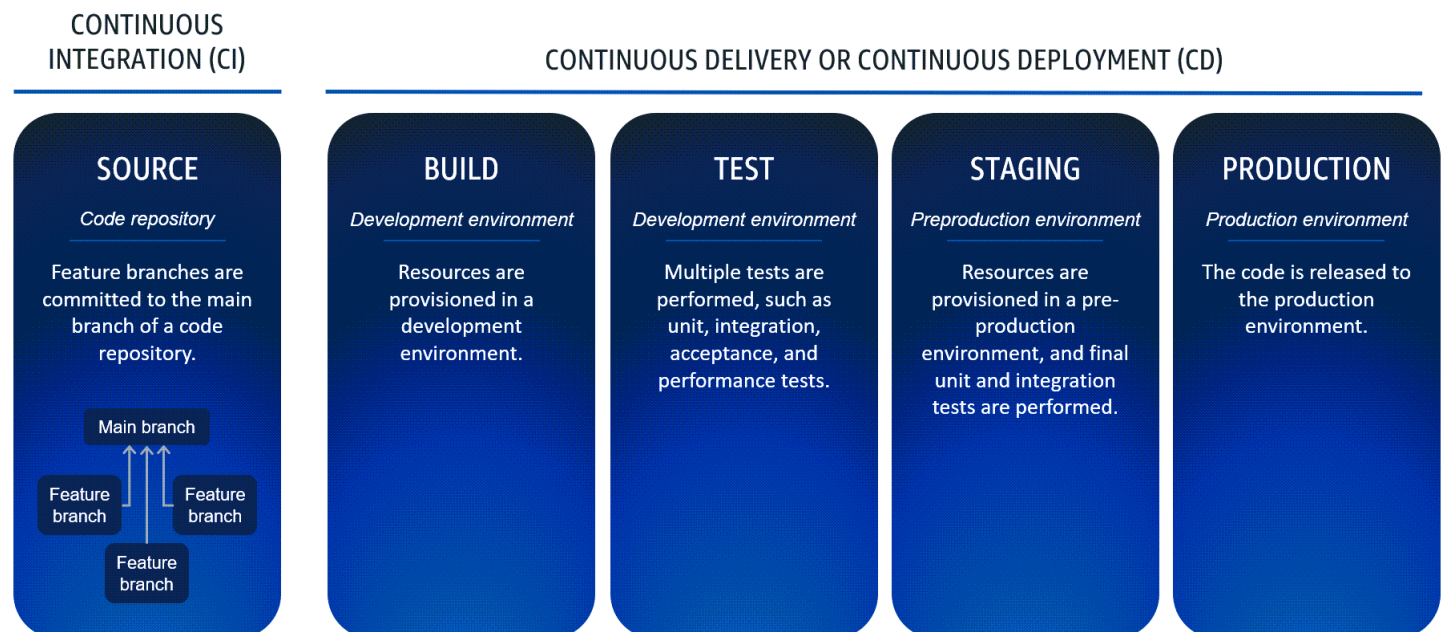
- Redução de custos — De acordo com um [relatório da Forbes Insight](#), “Três em cada quatro executivos concordam que a quantidade de tempo, dinheiro e recursos gastos em manutenção e gerenciamento contínuos — versus desenvolvimento de novos projetos ou novas iniciativas — está afetando a competitividade geral de sua organização”. Quanto menor o ciclo de desenvolvimento, maiores são as chances de sua organização atingir time-to-market metas ambiciosas e aproveitar as oportunidades certas no momento certo.
- Velocidade — Normalmente, um CI/CD pipeline is able to release software changes to customers within a few hours. Especially in cases with quick fault isolations and small patch pushes, the CI/CD pipeline completo ajuda a melhorar o tempo médio de recuperação (MTTR). Para obter mais informações, consulte [Reduzindo o MTTR](#).
- Segurança — CI/CD pipelines also secure the release process by reducing the possible entry points for attacks and reducing the risk of human error. The security gains that come with fully automated CI/CD pipelines completos ajudam a evitar as consequências dispendiosas de violações de dados, interrupções no serviço e muito mais.
- Redução do desgaste — Os desenvolvedores ficam mais satisfeitos quando podem passar mais tempo criando ótimos recursos e menos tempo atolados em um ciclo interminável de manutenção e depuração. Para as organizações, isso significa adquirir e reter os melhores talentos por longos períodos de tempo.
- Código de qualidade superior — Os desenvolvedores liberam o código em um repositório compartilhado em pequenos lotes, o que lhes permite [realizar testes paralelos](#) (postagem BrowserStack no blog). Em vez de trabalhar isoladamente, eles compartilham suas construções com a equipe com frequência e colaboram para identificar bugs críticos. Isso fornece suporte para desenvolvedores, o que ajuda a evitar que códigos incorretos cheguem à produção. Support de colegas desenvolvedores contribui para lançamentos de alta qualidade e impulsiona o crescimento organizacional.
- Manutenção — A manutenção e as atualizações são uma parte crucial da criação de um ótimo produto. No entanto, não desligue o sistema durante os horários de pico do tráfego. Você pode usar um pipeline de CI/CD para realizar a manutenção durante horas de baixo uso, minimizando o tempo de inatividade e os impactos no desempenho.

Entendendo o CI/CD

Integração e entrega contínuas (CI/CD) são o processo de automatizar o ciclo de vida do lançamento do software. Em alguns casos, o D em CI/CD também pode significar implantação. A diferença entre entrega contínua e implantação contínua ocorre quando você libera uma alteração no ambiente de produção. Com a entrega contínua, é necessária uma aprovação manual antes de promover mudanças na produção. A implantação contínua apresenta um fluxo ininterrupto em todo o pipeline, e nenhuma aprovação explícita é necessária. Como essa estratégia discute conceitos gerais de CI/CD, as recomendações e as informações fornecidas são aplicáveis às abordagens de entrega contínua e implantação contínua.

CI/CD automates much or all of the manual processes traditionally required to get new code from a commit into production. A CI/CD pipeline encompasses the source, build, test, staging, and production stages. In each stage, the CI/CD pipelines provisions any infrastructure that is needed to deploy or test the code. By using a CI/CD pipeline, as equipes de desenvolvimento podem fazer alterações no código que são automaticamente testadas e enviadas para implantação.

Vamos analisar os CI/CD process before discussing some of the ways that you can, knowingly or unknowingly, deviate from being fully CI/CD. The following diagram shows the CI/CD estágios e atividades básicos em cada estágio.



Sobre a integração contínua

A integração contínua ocorre em um repositório de código, como um repositório Git no GitHub. Você trata uma única ramificação principal como a fonte confiável da base de código e cria ramificações de curta duração para o desenvolvimento de recursos. Você integra uma ramificação de recursos à ramificação principal quando estiver pronto para implantar o recurso em ambientes superiores. As ramificações de recursos nunca são implantadas diretamente nos ambientes superiores. Para obter mais informações, consulte [Abordagem baseada em troncos](#) neste guia.

Processo de integração contínua

1. O desenvolvedor cria uma nova ramificação a partir da ramificação principal.
2. O desenvolvedor faz alterações, cria e testa localmente.
3. Quando as alterações estiverem prontas, o desenvolvedor cria uma [pull request](#) (GitHub documentação) com a ramificação principal como destino.
4. O código é revisado.
5. Quando o código é aprovado, ele é incorporado à ramificação principal.

Sobre a entrega contínua

A entrega contínua ocorre em ambientes isolados, como ambientes de desenvolvimento e ambientes de produção. As ações que ocorrem em cada ambiente podem variar. Muitas vezes, um dos primeiros estágios é usado para fazer atualizações no próprio pipeline antes de continuar. O resultado final da implantação é que cada ambiente é atualizado com as alterações mais recentes. O número de ambientes de desenvolvimento para criação e teste também varia, mas recomendamos que você use pelo menos dois. No pipeline, cada ambiente é atualizado em ordem de importância, terminando com o ambiente mais importante, o ambiente de produção.

Processo de entrega contínua

A parte de entrega contínua do pipeline começa retirando o código da ramificação principal do repositório de origem e passando-o para o estágio de construção. O documento de infraestrutura como código (IaC) do repositório descreve as tarefas que são executadas em cada estágio. Embora o uso de um documento do IaC não seja obrigatório, um serviço ou ferramenta do IaC, como [AWS CloudFormation](#) ou [AWS Cloud Development Kit \(AWS CDK\)](#), é altamente recomendado. As etapas mais comuns incluem:

1. Testes unitários
2. Construção de código
3. Provisionamento de recursos
4. Testes de integração

Se algum erro ocorrer ou algum teste falhar em qualquer estágio do pipeline, o estágio atual voltará ao estado anterior e o pipeline será encerrado. As alterações subsequentes devem começar no repositório de código e passar por todo o processo de CI/CD.

Testes para pipelines de CI/CD

Os dois tipos de testes automatizados comumente mencionados nos pipelines de implantação são testes unitários e testes de integração. No entanto, há muitos tipos de testes que você pode executar em uma base de código e no ambiente de desenvolvimento. A [arquitetura de referência do pipeline de AWS implantação](#) define os seguintes tipos de testes:

- Teste unitário — Esses testes criam e executam o código do aplicativo para verificar se ele está funcionando de acordo com as expectativas. Eles simulam todas as dependências externas usadas na base de código. Exemplos de ferramentas de teste unitário incluem [JUnitJest](#) e [pytest](#).
- Teste de integração — Esses testes verificam se o aplicativo atende aos requisitos técnicos por meio de testes em um ambiente de teste provisionado. Exemplos de ferramentas de teste de integração incluem [Cucumber](#), [vRest NG](#) e [integ-tests](#) (for). AWS CDK
- Teste de aceitação — Esses testes verificam se o aplicativo satisfaz os requisitos do usuário por meio de testes em um ambiente de teste provisionado. Exemplos de ferramentas de teste de aceitação incluem [Cypress](#) e [Selenium](#).
- Teste sintético — Esses testes são executados continuamente em segundo plano para gerar tráfego e verificar se o sistema está íntegro. Exemplos de ferramentas de teste sintético incluem [Amazon CloudWatch Synthetics](#) e [Dynatrace Synthetic](#) Monitoring.
- Teste de desempenho — Esses testes simulam a capacidade de produção. Eles determinam se o aplicativo atende aos requisitos de desempenho e comparam as métricas com o desempenho anterior. [Exemplos de ferramentas de teste de desempenho incluem Apache JMeter, Locust e Gatling](#).
- Teste de resiliência — Também conhecido como teste de caos, esses testes injetam falhas nos ambientes para identificar áreas de risco. Os períodos em que as falhas são injetadas são então

comparados aos períodos sem falhas. Exemplos de ferramentas de teste de resiliência incluem [AWS Fault Injection Service](#) e [Gremlin](#).

- Teste estático de segurança de aplicativos (SAST) — Esses testes analisam o código em busca de violações de segurança, como [injeção de SQL](#) ou [cross-site scripting \(XSS\)](#). Exemplos de ferramentas SAST incluem [Amazon CodeGuru](#) e [Checkmarx](#). [SonarQube](#)
- Teste dinâmico de segurança de aplicativos (DAST) — Esses testes também são conhecidos como testes de penetração ou teste de caneta. Eles identificam vulnerabilidades, como injeção de SQL ou XSS em um ambiente de teste provisionado. [Exemplos de ferramentas do DAST incluem o Zed Attack Proxy \(ZAP\) e o HCL. AppScan](#) Para obter mais informações, consulte [Teste de penetração](#).

Nem todos os pipelines totalmente de CI/CD executam todos esses testes. No entanto, no mínimo, um pipeline deve executar testes unitários e testes SAST na base de código, bem como testes de integração e aceitação em um ambiente de teste.

Métricas para pipelines de CI/CD

De acordo com a [arquitetura de referência do pipeline de AWS implantação](#), você deve, no mínimo, rastrear as quatro métricas a seguir para pipelines de CI/CD:

- Prazo de entrega — O tempo médio necessário para que um único compromisso entre em produção. Recomendamos definir um lead time entre 1 hora e 1 dia, conforme apropriado para seu caso de uso.
- Frequência de implantação — O número de implantações de produção em um determinado período de tempo. Recomendamos definir as frequências de implantação entre várias vezes por dia e duas vezes por semana, conforme apropriado para seu caso de uso.
- Tempo médio entre falhas (MTBF) — A quantidade média de tempo entre o início de uma tubulação bem-sucedida e o início de uma tubulação com falha. Recomendamos atingir um MTBF que seja o mais alto possível. Para obter mais informações, consulte [Aumentando o MTBF](#).
- Tempo médio de recuperação (MTTR) — O tempo médio entre o início de uma tubulação com falha e o início da próxima tubulação bem-sucedida. Recomendamos atingir um MTTR que seja o mais baixo possível. Para obter mais informações, consulte [Reduzindo o MTTR](#).

Essas métricas ajudam as equipes a acompanhar seu progresso para se tornarem totalmente CI/CD. As equipes devem ter discussões abertas com as partes interessadas da organização sobre

quais devem ser as metas ideais. As situações e necessidades variam muito de organização para organização e até mesmo de equipe para equipe.

É muito importante lembrar que mudanças rápidas e drásticas geralmente aumentam o risco de surgirem problemas. Estabeleça metas para buscar pequenas melhorias incrementais. Um prazo de entrega ideal comum para tubulações totalmente CI/CD é inferior a 3 horas. Uma equipe que começa com um prazo de entrega de 5,2 dias deve ter como meta uma redução de um dia a cada poucas semanas. Depois que essa equipe atingir um prazo de entrega de um dia ou menos, ela poderá permanecer lá por vários meses e passar para um lead time mais agressivo somente se as partes interessadas da equipe e da organização considerarem necessário.

Como os processos de CI/CD são totalmente diferentes

Os pipelines de CI/CD usam um fluxo de trabalho moderno baseado em troncos, no qual os desenvolvedores mesclam atualizações pequenas e frequentes em uma ramificação principal (ou tronco) que é criada e testada por meio da parte de CD do pipeline de CI/CD. Esse fluxo de trabalho substituiu o fluxo de trabalho do Gitflow, no qual as ramificações de desenvolvimento e lançamento são separadas por um cronograma de lançamento. Em muitas organizações, o Gitflow ainda é um método popular de controle de versão e implantação. No entanto, agora é considerado legado e pode ser difícil integrá-lo a um pipeline de CI/CD.

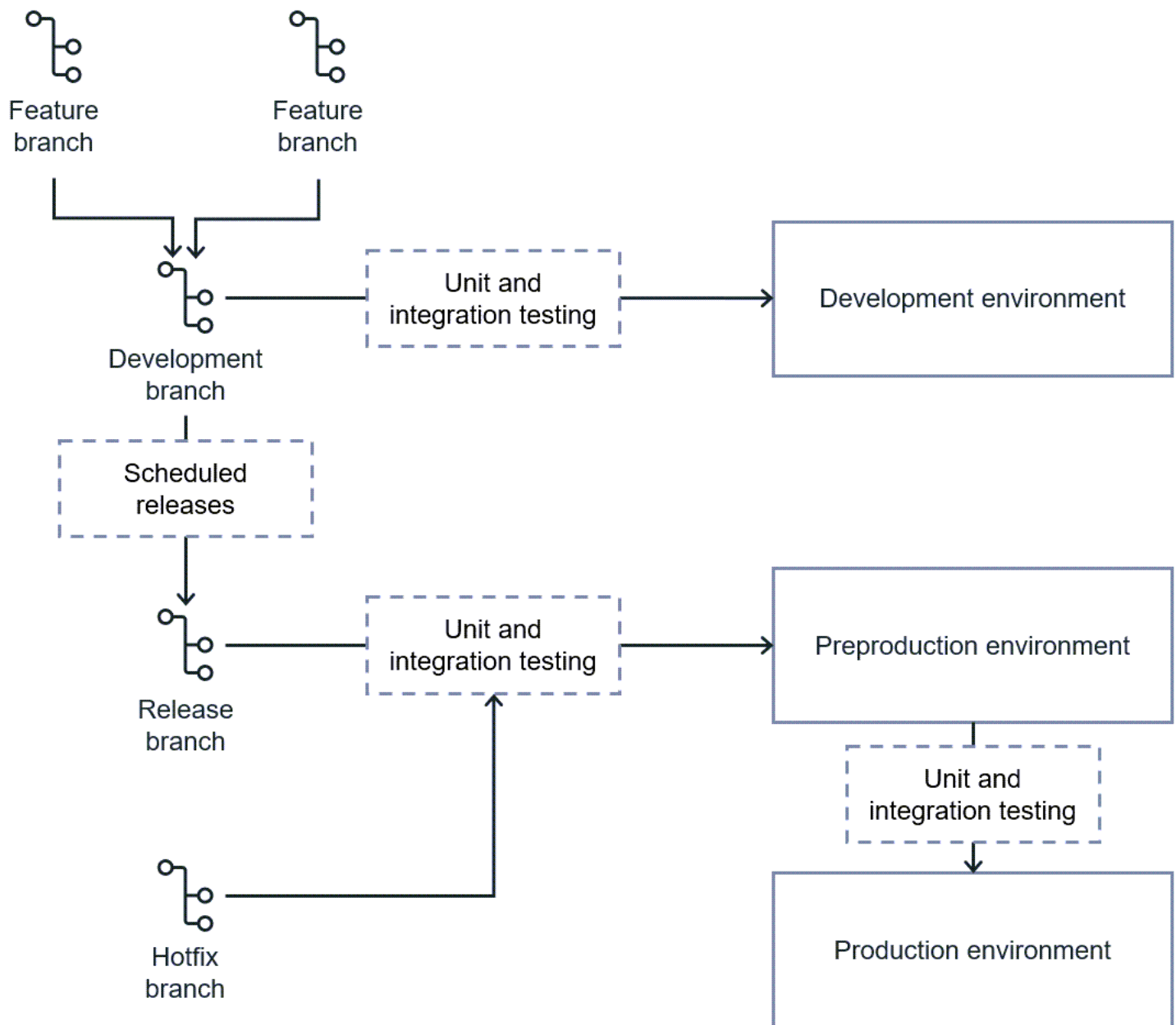
Para muitas organizações, a transição de um fluxo de trabalho do Gitflow para um fluxo de trabalho baseado em troncos foi incompleta, e o resultado é que elas ficam presas em algum lugar ao longo do caminho e nunca migram totalmente para o CI/CD. De alguma forma, seus pipelines acabam se agarrando a certos resquícios do fluxo de trabalho legado, presos em um estado de transição entre o passado e o presente. Analise as diferenças nos fluxos de trabalho do Git e, em seguida, saiba como o uso de um fluxo de trabalho legado pode afetar o seguinte:

- [Integridade do ambiente](#)
- [Versões](#)
- [Segurança](#)

[Para facilitar a identificação dos remanescentes de um fluxo de trabalho antigo do Git em uma configuração moderna, vamos comparar o Gitflow com a abordagem moderna baseada em troncos.](#)

Abordagem Gitflow

A imagem a seguir mostra um fluxo de trabalho do Gitflow. A abordagem do Gitflow usa várias ramificações para rastrear várias versões diferentes do código ao mesmo tempo. Você agenda lançamentos de atualizações em um aplicativo para algum momento no futuro, enquanto os desenvolvedores ainda trabalham na versão atual do código. Repositórios baseados em troncos podem usar sinalizadores de recursos para fazer isso, mas eles são incorporados ao Gitflow por padrão.



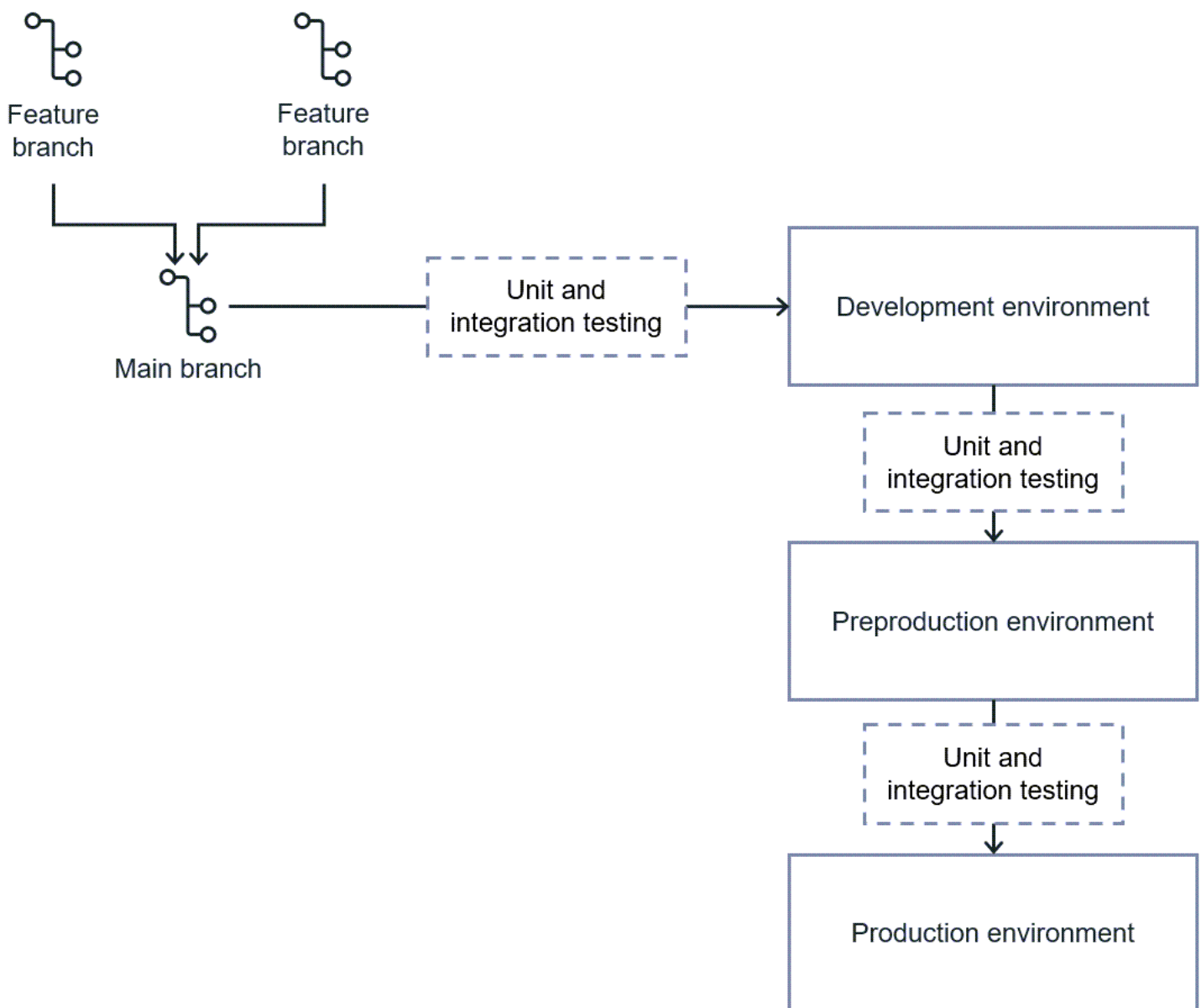
Um resultado da abordagem do Gitflow é que os ambientes de aplicativos geralmente estão fora de sincronia. Em uma implementação padrão do Gitflow, os ambientes de desenvolvimento refletem o estado atual do código, enquanto os ambientes de pré-produção e produção permanecem congelados no estado da base de código desde a versão mais recente.

Isso complica as coisas quando um defeito aparece no ambiente de produção porque a base de código na qual os desenvolvedores trabalham não pode ser mesclada à produção sem expor recursos não lançados. A forma como o Gitflow lida com essa situação é usando um hotfix. Uma ramificação de hotfix é criada a partir da ramificação de lançamento e depois implantada

diretamente nos ambientes superiores. A ramificação do hotfix é então mesclada à ramificação de desenvolvimento para manter o código atualizado.

Abordagem baseada em troncos

A imagem a seguir mostra um fluxo de trabalho baseado em troncos. Em um fluxo de trabalho baseado em troncos, 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. Os testes unitários e de integração ocorrem entre cada ambiente.



Usando esse fluxo de trabalho, todos os ambientes estão operando a mesma base de código. Não há necessidade de uma ramificação de hotfix para os ambientes superiores porque você pode implementar alterações na ramificação principal sem expor recursos não lançados. A ramificação principal é sempre considerada estável, livre de defeitos e pronta para ser liberada. Isso ajuda você a integrá-lo como fonte para um pipeline de CI/CD, que pode testar e implantar automaticamente sua base de código em todos os ambientes do pipeline.

Benefícios de integridade ambiental de uma abordagem baseada em troncos

Como muitos desenvolvedores sabem, uma mudança no código às vezes pode criar um [efeito borboleta](#) (artigo da American Scientist), em que um pequeno desvio aparentemente não relacionado desencadeia uma reação em cadeia que causa resultados inesperados. Os desenvolvedores devem então investigar completamente para descobrir a causa raiz.

Quando os cientistas conduzem um experimento, eles separam as cobaias em dois grupos: o grupo experimental e o grupo controle. A intenção é tornar o grupo experimental e o grupo controle completamente idênticos, exceto pelo que está sendo testado no experimento. Quando algo acontece no grupo experimental que não acontece no grupo controle, a única causa pode ser a coisa que está sendo testada.

Pense nas mudanças em uma implantação como um grupo experimental e pense em cada ambiente como grupos de controle separados. Os resultados dos testes em um ambiente inferior só são confiáveis quando os controles são iguais aos de um ambiente superior. Quanto mais os ambientes se desviam, maior a chance de descobrir defeitos nos ambientes superiores. Em outras palavras, se as alterações no código falharem na produção, preferimos que elas falhem primeiro na versão beta para que nunca cheguem à produção. É por isso que todos os esforços devem ser feitos para manter cada ambiente, desde o ambiente de teste mais baixo até a própria produção, em sincronia. Isso é chamado de integridade do ambiente.

O objetivo de qualquer processo completo de CI/CD é descobrir problemas o mais cedo possível. Preservar a integridade do ambiente usando uma abordagem baseada em troncos pode praticamente eliminar a necessidade de hotfixes. Em um fluxo de trabalho baseado em troncos, é raro que um problema apareça pela primeira vez no ambiente de produção.

Em uma abordagem Gitflow, depois que um hotfix é implantado diretamente nos ambientes superiores, ele é adicionado à ramificação de desenvolvimento. Isso preserva a correção para futuros lançamentos. No entanto, o hotfix foi desenvolvido e testado diretamente do estado atual

do aplicativo. Mesmo que o hotfix funcione perfeitamente na produção, existe a possibilidade de surgirem problemas ao interagir com os recursos mais novos no ramo de desenvolvimento. Como a implantação de um hotfix para um hotfix geralmente não é desejável, isso faz com que os desenvolvedores gastem mais tempo tentando adaptar o hotfix ao ambiente de desenvolvimento. Em muitos casos, isso pode levar a uma dívida técnica significativa e reduzir a estabilidade geral do ambiente de desenvolvimento.

Quando ocorre uma falha em um ambiente, todas as alterações são revertidas para que o ambiente retorne ao estado anterior. Qualquer alteração em uma base de código deve reiniciar o pipeline desde o primeiro estágio. Quando surge um problema no ambiente de produção, a correção também deve passar por todo o pipeline. O tempo extra necessário para percorrer os ambientes mais baixos geralmente é insignificante em comparação com os problemas que são evitados com o uso dessa abordagem. Como todo o propósito dos ambientes inferiores é detectar erros antes que eles cheguem à produção, contornar esses ambientes por meio de uma abordagem Gitflow é um risco ineficiente e desnecessário.

Libere os benefícios de uma abordagem baseada em troncos

Uma das coisas que geralmente torna necessário um hotfix é que, em um fluxo de trabalho legado, o estado do aplicativo no qual os desenvolvedores estão trabalhando pode conter vários recursos inéditos que ainda não estão em produção. O ambiente de produção e o ambiente de desenvolvimento só ficam sincronizados quando ocorre um lançamento programado e, em seguida, eles imediatamente começam a divergir novamente até o próximo lançamento programado.

Ter lançamentos agendados é possível em um CI/CD process. You can delay the release of code to production by using feature flags. However, a fully CI/CD processo completo, permitindo mais flexibilidade, tornando desnecessários os lançamentos agendados. Afinal, contínuo é uma palavra-chave em CI/CD, e isso sugere que as mudanças são lançadas à medida que ficam prontas. Evite manter um ambiente de lançamento separado que esteja quase sempre fora de sincronia com os ambientes de teste inferiores.

Se uma tubulação não for totalmente CI/CD, a divergência entre os ambientes superior e inferior geralmente ocorre no nível da ramificação. Os desenvolvedores trabalham em uma ramificação de desenvolvimento e mantêm uma ramificação de lançamento separada que é atualizada somente quando chega a hora de um lançamento agendado. À medida que o ramo de liberação e o ramo de desenvolvimento divergem, outras complicações podem surgir.

Além de os ambientes estarem fora de sincronia, à medida que os desenvolvedores trabalham na ramificação de desenvolvimento e se acostumam com um estado do aplicativo muito à frente do

que está em produção, eles precisam se reajustar ao estado de produção sempre que surge um problema. O estado do ramo de desenvolvimento pode estar em muitos aspectos antes da produção. Quando os desenvolvedores trabalham nessa filial todos os dias, é difícil lembrar o que foi e o que não foi lançado para produção. Isso aumenta o risco de que novos bugs sejam introduzidos durante o processo de correção de outros bugs. Esse resultado é um ciclo aparentemente interminável de correções que estendem os prazos e atrasam o lançamento de recursos por semanas, meses ou até anos.

Benefícios de segurança de uma abordagem baseada em troncos

Um processo totalmente CI/CD fornece uma abordagem de implantação totalmente automatizada e única de fonte confiável. O gasoduto tem um único ponto de entrada. As atualizações de software entram no pipeline no início e são passadas como estão de um ambiente para o outro. Se um problema for descoberto em qualquer estágio do pipeline, as alterações de código que o corrigem devem passar pelo mesmo processo e começar no primeiro estágio. Reduzir os pontos de entrada em um pipeline também reduz as possíveis maneiras pelas quais as vulnerabilidades podem ser introduzidas no pipeline.

Além disso, como o ponto de entrada é o ponto mais distante possível do ambiente de produção, isso reduz drasticamente a probabilidade de vulnerabilidades atingirem a produção. Se você implementar um processo de aprovação manual em um pipeline totalmente de CI/CD, ainda poderá permitir a tomada de decisão contínua ou não sobre se as mudanças serão promovidas para o próximo ambiente. O tomador de decisão não é necessariamente a mesma pessoa que implementa as mudanças. Isso separa as responsabilidades do implantador das alterações de código e do aprovador dessas alterações. Também torna mais viável que um líder de organização menos técnico desempenhe a função de aprovador.

Por fim, o único ponto de entrada ajuda a limitar o acesso de gravação ao console da interface de usuário (UI) do ambiente de produção a alguns ou até mesmo zero usuários. Ao reduzir o número de usuários que podem fazer alterações manuais no console, você reduz o risco de eventos de segurança. A capacidade de gerenciar manualmente o console no ambiente de produção é muito mais necessária em fluxos de trabalho legados do que em uma abordagem automatizada de CI/CD. Essas alterações manuais são mais difíceis de monitorar, revisar e testar. Eles geralmente são executados para economizar tempo, mas, a longo prazo, acrescentam uma dívida técnica significativa ao projeto.

Os problemas de segurança do console não são necessariamente causados por agentes mal-intencionados. Muitos dos problemas que ocorrem no console são acidentais. A exposição acidental

à segurança é muito comum e levou ao surgimento do modelo de segurança de confiança zero. Esse modelo postula, em parte, que acidentes de segurança são menos prováveis quando até mesmo a equipe interna tem o mínimo de acesso possível, também conhecido como permissões de privilégio mínimo. Preservar a integridade do ambiente de produção ao restringir todos os processos a um pipeline automatizado praticamente elimina o risco de problemas de segurança relacionados ao console.

Teste decisivo para tubulações de CI/CD

Em química, o papel de tornassol é uma tira fina de papel tratada com um corante vermelho ou azul especial usado para determinar a acidez de uma substância. Um ácido torna o papel de tornassol azul vermelho, uma base torna o papel de tornassol vermelho azul e substâncias neutras não afetam em nada a cor do papel.

A forma como o papel de tornassol determina a acidez é medindo o nível de pH de uma substância. Se um nível de pH for maior que 8, é ácido; se estiver abaixo de 5, é básico; e se estiver entre 5 e 8, é neutro. Da mesma forma, o [teste decisivo de CI/CD](#) ajuda você a medir o nível de CI/CD do seu pipeline.

Para testar se seu pipeline é totalmente CI/CD

1. Comece com uma pontuação de 0.
2. Responda a cada uma das perguntas a seguir e adicione 1 à sua pontuação sempre que responder sim:
 - Cada um de nossos repositórios tem exatamente uma ramificação principal usada para implantação em ambientes?
 - Enviamos código para a ramificação principal com frequência e evitamos ramificações de recursos de longa duração?
 - Nosso pipeline tem um único ponto de entrada? Em outras palavras, nosso pipeline extrai o código de cada repositório exatamente uma vez?
 - Temos mais de um ambiente de implantação?
 - Quando o pipeline não está funcionando, nossos ambientes superior e inferior geralmente estão sincronizados?
 - Executamos testes no código antes da implantação?
 - Executamos testes em um ambiente antes de passar para o próximo ambiente?
 - Nosso pipeline faz uma reversão completa e sai após uma falha?
 - Nosso pipeline é reiniciado desde a primeira etapa ao se recuperar de uma falha?
 - Seguimos o mesmo processo para corrigir bugs na produção que fazemos para liberar recursos para produção?
 - Usamos alguma forma de infraestrutura como modelo de código (IaC) para implantar código?

3. Responda a cada uma das perguntas a seguir e adicione 1 à sua pontuação para cada vez que você responder não:
- Já implantamos diretamente em um ambiente de implantação a partir de filiais que não sejam a filial principal?
 - Já implantamos diretamente de qualquer filial para um ambiente superior ou de produção?
 - Costumamos encontrar bugs em ambientes superiores que não estavam presentes em ambientes inferiores?
 - Alguma vez contornamos ambientes inferiores durante uma implantação?
 - Esperamos até um horário de lançamento programado para implantar na produção?
 - Fazemos atualizações regularmente no console do ambiente de produção?
 - Há alguma etapa de implantação manual que deve ser realizada no console do ambiente de produção para concluir a implantação?
 - Mais de uma pessoa tem acesso de gravação ao ambiente de produção?
 - Mais de cinco pessoas têm acesso de gravação ao ambiente de produção?
4. Divida sua pontuação por 2. Essa é a pontuação de CI/CD do seu funil.
5. Compare o CI/CD score to the following table to determine your pipeline's CI/CD nível do seu funil.

Pontuação de CI/CD	Nível de CI/CD
9,5 ou superior	Totalmente CI/CD
8—9	Principalmente CI/CD
5—7	Neutro
Abaixo de 5	Não é CI/CD

Se você pontuou abaixo de 8, recomendamos que você defina uma meta para avançar gradualmente para o próximo nível. Quando essa meta for alcançada, as partes interessadas do produto devem avaliar se e quando uma nova meta deve ser definida. A intenção deste exercício não é necessariamente defender uma mudança em seu pipeline, mas sim conscientizar sobre a aparência de um processo completo de implantação de CI/CD e onde seus pipelines estão atualmente nesse espectro.

Melhores práticas para pipelines de CI/CD

A seguir estão as melhores práticas para pipelines totalmente de CI/CD:

- **Proteja o ambiente de produção** — Como é possível realizar praticamente tudo o que é necessário para a manutenção da conta e do ambiente usando o IaC, é importante fazer todos os esforços para proteger o ambiente de produção limitando o acesso programático e ao console. Recomendamos limitar o acesso a apenas alguns usuários, ou mesmo a zero. Quando você implanta o IaC por meio de AWS CloudFormation, o usuário precisa de permissões limitadas. A maioria das permissões é atribuída ao CloudFormation serviço por meio de uma função de serviço. Para obter mais informações, consulte [Função de serviço](#) na CloudFormation documentação e [Implementação de políticas para permissões de privilégios mínimos](#) para AWS CloudFormation.
- **Crie contas separadas para cada ambiente** — Ao dedicar uma conta separada a cada ambiente, você pode simplificar o processo de implantação e criar controles de acesso refinados no nível da conta. Quando vários ambientes compartilham recursos, isso reduz a integridade do ambiente como uma unidade isolada. É melhor manter os ambientes sincronizados e distintos. Isso é ainda mais importante para o ambiente de produção porque tudo nessa conta deve ser tratado como um recurso de produção.
- **Restrinja as informações de identificação pessoal (PII) ao ambiente de produção** — Tanto para segurança quanto para proteção contra riscos de responsabilidade, proteja as PII o máximo possível. Quando possível em ambientes inferiores, use dados anônimos ou de amostra em vez de copiar dados potencialmente confidenciais do ambiente de produção.
- **Revise o código em repositórios** — Um processo totalmente de CI/CD reduz os pontos de entrada de um pipeline a um único ponto, e esse ponto único deve ser protegido. Por esse motivo, é recomendável que você exija várias revisões de código antes de mesclar as ramificações de recursos na ramificação principal. Essas análises de código podem ser conduzidas por qualquer membro qualificado da equipe, mas pelo menos um membro sênior deve revisá-las. O código deve ser testado rigorosamente pelo revisor. Afinal, a melhor maneira de corrigir problemas em um pipeline é evitar introduzi-los nele. Além disso, é importante resolver todos os comentários feitos por qualquer revisor antes da fusão. Essa resolução pode ser simplesmente uma explicação de por que nenhuma mudança é necessária, mas abordar todos os comentários é uma verificação extra importante para ajudar a evitar a introdução de problemas no pipeline.
- **Faça fusões pequenas e frequentes** — Para aproveitar ao máximo a integração contínua, é uma boa ideia inserir mudanças locais no pipeline de forma contínua. Afinal, é muito mais benéfico que

os ambientes de desenvolvimento permaneçam sincronizados se os ambientes locais também os acompanharem.

Para obter mais práticas recomendadas para pipelines de CI/CD, consulte [Resumo das melhores práticas](#) na prática da integração contínua e da entrega contínua em. AWS

Perguntas frequentes

Quais são alguns indicadores-chave de que meu processo de implantação não é totalmente CI/CD?

O indicador mais comum é quando há várias ramificações do repositório que representam ambientes separados em um pipeline. Os repositórios em um processo totalmente de CI/CD usam um fluxo de trabalho baseado em troncos, no qual uma ramificação atua como a única fonte confiável para as implantações desse repositório. Para obter mais informações, consulte [Abordagem baseada em troncos](#). Outros indicadores incluem etapas de implantação manual, além de decisões simples de usar ou não, o uso de hotfixes e lançamentos programados.

E se eu quiser usar um processo totalmente de CI/CD, mas ainda quiser programar lançamentos de determinados recursos para momentos específicos?

Isso geralmente é feito com sinalizadores de recursos. Nesse processo, as implantações ainda são feitas continuamente, mas certos recursos são ocultados pelo uso de fechamentos condicionais no código até a hora de lançá-los.

E se algumas etapas do meu processo de implantação não puderem ser automatizadas?

Um dos objetivos de um pipeline totalmente de CI/CD é minimizar a necessidade de processos manuais, mas certamente existem possíveis casos de uso em que processos manuais podem ser necessários. Na verdade, processos somente para leitura, como consultar registros de aplicativos, geralmente podem ser realizados em ambientes de produção com risco mínimo. No entanto, é altamente recomendável que você trate as ações de gravação manual na produção como um último recurso absoluto.

E se minha equipe técnica estiver mais confortável com fluxos de trabalho antigos do que com um processo totalmente de CI/CD?

É comum que a equipe técnica seja resistente a grandes mudanças, especialmente quando algo que costumava ser uma prática recomendada é substituído por algo mais novo. A tecnologia avança rapidamente e melhorias são descobertas constantemente. Embora um certo grau de ceticismo seja uma boa qualidade para uma equipe técnica, é igualmente importante que ela esteja aberta a mudanças. Não se mova muito rápido com funcionários céticos, pois eles precisam gerenciar as mudanças no sistema antes de serem implementadas. A chave é evitar que os céticos permaneçam estáticos para sempre.

E se meus ambientes estiverem em várias contas? Ainda posso usar um processo totalmente de CI/CD?

Sim, na verdade, é recomendável usar uma conta separada para cada ambiente. Para obter mais informações sobre um pipeline que ativa estágios em diferentes contas, consulte [Criar um pipeline CodePipeline que usa recursos de outra Conta da AWS](#).

Próximas etapas

Use a [Teste decisivo para tubulações de CI/CD](#) seção para avaliar os DevOps processos em sua organização. Determine se seus processos são totalmente CI/CD. If they aren't, decide whether they need improvement to take full advantage of the benefits of CI/CD implantados.

Como você sabe quando terminou? Bem, a resposta é que muitas organizações nunca terminam de fato. Eles param em algum lugar ao longo do caminho, em um local adequado para seu caso de uso. Porém, uma CI/CD pipeline is the best-case scenario, it depends heavily on the organizational situation and the stakeholders behind the decision. Stakeholders must decide which stage of the CI/CD implementação completa funciona melhor para seu caso de uso e para a melhor forma de mapear a progressão para as próximas fases.

Para obter mais informações sobre como projetar e criar pipelines de CI/CD, consulte. [Recursos](#)

Recursos

AWS documentação e referências

- [Arquitetura de referência do pipeline de implantação da AWS](#)
- [O que é entrega contínua?](#)
- [Praticando a integração contínua e a entrega contínua em AWS](#) (AWS whitepaper)
- [Configurar um pipeline de CI/CD em AWS](#)(tutorialAWS prático)
- [Crie um pipeline Regiões da AWS que não suporte AWS CodePipeline](#) (orientação AWS prescritiva)
- [Arquitetura de referência e implementações de referência de pipelines de implantação](#) (AWS postagem no blog)

Serviços e ferramentas

- [O teste CI/CD Litmus](#)
- [AWS Cloud Development Kit \(AWS CDK\)](#)
- [AWS CloudFormation](#)
- [AWS CodePipeline](#)

Histórico do documento

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

Alteração	Descrição	Data
Publicação inicial	—	25 de agosto de 2023

AWS Glossário de orientação prescritiva

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

Números

7 Rs

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

- 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: migre seu banco de dados Oracle local para a edição compatível com o Amazon Aurora 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: Migre seu banco de dados Oracle local para o Amazon Relational Database Service (Amazon RDS) for Oracle no. Nuvem AWS
- Recomprar (drop and shop): mude para um produto diferente, normalmente migrando de uma licença tradicional para um modelo SaaS. Exemplo: migre seu sistema de gerenciamento de relacionamento com o cliente (CRM) para a 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: Migre seu banco de dados Oracle local para o Oracle em uma EC2 instância no. Nuvem AWS
- Realocar (mover o hipervisor sem alterações [hypervisor-level lift-and-shift]): mover a infraestrutura para a nuvem sem comprar novo hardware, reescrever aplicações ou modificar suas operações existentes. Você migra servidores de uma plataforma local para um serviço em nuvem para a mesma plataforma. Exemplo: Migrar um Microsoft Hyper-V aplicativo para o. AWS
- Reter (revisitar): mantenha as aplicações em seu ambiente de origem. Isso pode incluir aplicações que exigem grande refatoração, e você deseja adiar esse trabalho para um

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

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

A

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 AIOps é usado na estratégia de AWS migração, consulte o [guia de integração de operações](#).

criptografia assimétrica

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

atomicidade, consistência, isolamento, durabilidade (ACID)

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

controle de acesso por atributo (ABAC)

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

fonte de dados autorizada

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

Zona de disponibilidade

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

AWS Estrutura de adoção da nuvem (AWS CAF)

Uma estrutura de diretrizes e melhores práticas AWS para ajudar as organizações a desenvolver um plano eficiente e eficaz para migrar com sucesso para a nuvem. AWS O CAF organiza a orientação em seis áreas de foco chamadas perspectivas: negócios, pessoas, governança, plataforma, segurança e operações. As perspectivas de negócios, pessoas e governança têm como foco habilidades e processos de negócios; as perspectivas de plataforma, segurança e operações concentram-se em habilidades e processos técnicos. Por exemplo, a perspectiva das pessoas tem como alvo as partes interessadas que lidam com recursos humanos (RH), funções de pessoal e gerenciamento de pessoal. Nessa perspectiva, o AWS CAF fornece orientação para desenvolvimento, treinamento e comunicação de pessoas para ajudar a preparar a organização para a adoção bem-sucedida da nuvem. Para obter mais informações, consulte o [site da AWS CAF](#) e o [whitepaper da AWS CAF](#).

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

Uma ferramenta que avalia as cargas de trabalho de migração do banco de dados, recomenda estratégias de migração e fornece estimativas de trabalho. AWS O WQF está incluído com AWS

Schema Conversion Tool (AWS SCT). Ela analisa esquemas de banco de dados e objetos de código, código de aplicações, dependências e características de performance, além de fornecer relatórios de avaliação.

B

bot ruim

Um [bot](#) destinado a perturbar ou causar danos a indivíduos ou organizações.

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.

blue/green deployment (implantação azul/verde)

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

bot

Um aplicativo de software que executa tarefas automatizadas pela Internet e simula a atividade ou interação humana. Alguns bots são úteis ou benéficos, como rastreadores da Web que indexam informações na Internet. Alguns outros bots, conhecidos como bots ruins, têm como objetivo perturbar ou causar danos a indivíduos ou organizações.

botnet

Redes de [bots](#) infectadas por [malware](#) e sob o controle de uma única parte, conhecidas como pastor de bots ou operador de bots. As redes de bots são o mecanismo mais conhecido para escalar bots e seu impacto.

ramo

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

acesso 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-Architected](#) AWS .

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](#).

implantação canária

O lançamento lento e incremental de uma versão para usuários finais. Quando estiver confiante, você implanta a nova versão e substituirá a versão atual em sua totalidade.

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

Centro de excelência em 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 CCo E](#) no Blog de Estratégia Nuvem AWS Empresarial.

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 quando migram para o Nuvem AWS:

- Projeto: executar alguns projetos relacionados à nuvem para fins de prova de conceito e aprendizado
- Fundação — Fazer investimentos fundamentais para escalar sua adoção da nuvem (por exemplo, criar uma landing zone, definir um CCo E, 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 postagem do blog [The Journey Toward Cloud-First & the Stages of Adoption](#) no blog de estratégia Nuvem AWS empresarial. Para obter

informações sobre como eles se relacionam com a estratégia de AWS migração, 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 Bitbucket Cloud. Cada versão do código é chamada de ramificação. Em uma estrutura de microsserviços, cada repositório é dedicado a uma única peça de funcionalidade. Um único 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 (CV)

Um campo da [IA](#) que usa aprendizado de máquina para analisar e extrair informações de formatos visuais, como imagens e vídeos digitais. Por exemplo, a Amazon SageMaker AI fornece algoritmos de processamento de imagem para CV.

desvio de configuração

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

banco de dados de gerenciamento de configuração (CMDB)

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

pacote de conformidade

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

integração contínua e entrega contínua (CI/CD)

O processo de automatizar os estágios de origem, criação, teste, preparação e produção do processo de lançamento do software. CI/CD is commonly described as a pipeline. CI/CD pode ajudá-lo a automatizar processos, melhorar a produtividade, melhorar a qualidade do código e entregar com mais rapidez. Para obter mais informações, consulte [Benefícios da entrega contínua](#). CD também pode significar implantação contínua. Para obter mais informações, consulte [Entrega contínua versus implantação contínua](#).

CV

Veja [visão computacional](#).

D

dados em repouso

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

classificação de dados

Um processo para identificar e categorizar os dados em sua rede com base em criticalidade e confidencialidade. É um componente crítico de qualquer estratégia de gerenciamento de riscos de segurança cibernética, pois ajuda a determinar os controles adequados de proteção e retenção para os dados. A classificação de dados é um componente do pilar de segurança 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.

malha de dados

Uma estrutura arquitetônica que fornece propriedade de dados distribuída e descentralizada com gerenciamento e governança centralizados.

minimização de dados

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

perímetro de dados

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

pré-processamento de dados

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

proveniência dos dados

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

titular dos dados

Um indivíduo cujos dados estão sendo coletados e processados.

data warehouse

Um sistema de gerenciamento de dados 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 AWS, você adiciona vários controles em diferentes camadas da AWS Organizations estrutura para ajudar a proteger os recursos. Por exemplo, uma defense-in-depth abordagem pode combinar autenticação multifatorial, segmentação de rede e criptografia.

administrador delegado

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

implantação

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

ambiente de desenvolvimento

Veja 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 (RD)

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 de cargas de trabalho em AWS: Recuperação na nuvem no](#) AWS Well-Architected Framework.

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

Rastreando 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](#).

EDI

Veja [intercâmbio eletrônico de dados](#).

computação de borda

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

intercâmbio eletrônico de dados (EDI)

A troca automatizada de documentos comerciais entre organizações. Para obter mais informações, consulte [O que é intercâmbio eletrônico de dados](#).

Criptografia

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

chave de criptografia

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

endianismo

A ordem na qual os bytes são armazenados na memória do computador. 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. Você pode criar um serviço de endpoint com AWS PrivateLink e conceder permissões a outros diretores Contas da AWS ou a AWS Identity and Access Management (IAM). Essas contas ou entidades principais podem se conectar ao serviço de endpoint de maneira privada criando endpoints da VPC de interface. Para obter mais informações, consulte [Criar um serviço de endpoint](#) na documentação do Amazon Virtual Private Cloud (Amazon VPC).

planejamento de recursos corporativos (ERP)

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

criptografia envelopada

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

ambiente

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

- ambiente de desenvolvimento: uma instância de uma aplicação em execução que está disponível somente para a equipe principal responsável pela manutenção da aplicação. Ambientes de desenvolvimento são usados para testar mudanças antes de promovê-las para ambientes superiores. Esse tipo de ambiente às vezes é chamado de ambiente de teste.
- ambientes inferiores: todos os ambientes de desenvolvimento para uma aplicação, como aqueles usados para compilações e testes iniciais.
- ambiente de produção: uma instância de uma aplicação em execução que os usuários finais podem acessar. Em um 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 épicos de segurança AWS da CAF incluem gerenciamento de identidade e acesso, controles de detetive, segurança de infraestrutura, proteção de dados e resposta a incidentes. Para obter mais informações sobre epics na estratégia de migração da AWS, consulte o [guia de implementação do programa](#).

ERP

Veja o [planejamento de recursos corporativos](#).

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

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

solicitação rápida

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

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.

FM

Veja o [modelo da fundação](#).

modelo de fundação (FM)

Uma grande rede neural de aprendizado profundo que vem treinando em grandes conjuntos de dados generalizados e não rotulados. FMs são capazes de realizar uma ampla variedade de tarefas gerais, como entender a linguagem, gerar texto e imagens e conversar em linguagem natural. Para obter mais informações, consulte [O que são modelos básicos](#).

G

IA generativa

Um subconjunto de modelos de [IA](#) que foram treinados em grandes quantidades de dados e que podem usar uma simples solicitação de texto para criar novos conteúdos e artefatos, como imagens, vídeos, texto e áudio. Para obter mais informações, consulte [O que é IA generativa](#).

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.

imagem dourada

Um instantâneo de um sistema ou software usado como modelo para implantar novas instâncias desse sistema ou software. Por exemplo, na manufatura, uma imagem dourada pode ser usada para provisionar software em vários dispositivos e ajudar a melhorar a velocidade, a escalabilidade e a produtividade nas operações de fabricação de dispositivos.

estratégia greenfield

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

barreira de proteção

Uma regra de alto nível que ajuda a governar recursos, políticas e conformidade em todas as unidades organizacionais (OUs). 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.

dados de retenção

Uma parte dos dados históricos rotulados que são retidos de um conjunto de dados usado para treinar um modelo de aprendizado [de máquina](#). Você pode usar dados de retenção para avaliar o desempenho do modelo comparando as previsões do modelo com os dados de retenção.

migração homogênea de bancos de dados

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

dados quentes

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

hotfix

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

período de hipercuidados

Imediatamente após a substituição, o período em que uma equipe de migração gerencia e monitora as aplicações migradas na nuvem para resolver quaisquer problemas. Normalmente,

a duração desse período é de 1 a 4 dias. No final do período de hipercuidados, a equipe de migração normalmente transfere a responsabilidade pelas aplicações para a equipe de operações de nuvem.

eu

IaC

Veja a [infraestrutura como código](#).

Política baseada em identidade

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

aplicação ociosa

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

IIoT

Veja 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 AWS várias contas, uma VPC que aceita, inspeciona e roteia conexões de rede de fora de um aplicativo. A [Arquitetura de Referência de AWS Segurança](#) recomenda configurar sua conta de rede com entrada, saída e inspeção VPCs para proteger a interface bidirecional entre seu aplicativo e a Internet em geral.

migração incremental

Uma estratégia de substituição na qual você migra a aplicação em pequenas partes, em vez de realizar uma única substituição completa. Por exemplo, é possível mover inicialmente

apenas alguns microsserviços ou usuários para o novo sistema. Depois de verificar se tudo está funcionando corretamente, mova os microsserviços ou usuários adicionais de forma incremental até poder descomissionar seu sistema herdado. Essa estratégia reduz os riscos associados a migrações de grande porte.

Indústria 4.0

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

infraestrutura

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

Infraestrutura como código (IaC)

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

Internet industrial das coisas (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 [Criando uma estratégia de transformação digital industrial da Internet das Coisas \(IIoT\)](#).

VPC de inspeção

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

Internet das Coisas (IoT)

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

interpretabilidade

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

IoT

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

informações sobre zonas de pouso, consulte [Configurar um ambiente da AWS com várias contas seguro e escalável](#).

modelo de linguagem grande (LLM)

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

migração de grande porte

Uma migração de 300 servidores ou mais.

LBAC

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

privilegio mínimo

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

mover sem alterações (lift-and-shift)

Veja [7 Rs](#).

sistema little-endian

Um sistema que armazena o byte menos significativo antes. Veja também [endianness](#).

LLM

Veja [um modelo de linguagem grande](#).

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](#).

malware

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

serviços gerenciados

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

sistema de execução de manufatura (MES)

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

MAP

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 de membro

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

MES

Veja o [sistema de execução de manufatura](#).

Transporte de telemetria de enfileiramento de mensagens (MQTT)

Um protocolo de comunicação leve machine-to-machine (M2M), baseado no padrão de publicação/assinatura, para dispositivos de IoT com recursos limitados.

microserviço

Um serviço pequeno e independente que se comunica de forma bem definida APIs e normalmente é de propriedade de equipes pequenas e independentes. Por exemplo, um sistema de seguradora pode incluir microserviços que mapeiam as capacidades comerciais, como vendas ou marketing, ou subdomínios, como compras, reclamações ou análises. Os benefícios dos microserviços incluem agilidade, escalabilidade flexível, fácil implantação, código reutilizável e resiliência. Para obter mais informações, consulte [Integração de microserviços usando serviços sem AWS servidor](#).

arquitetura de microserviços

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

Programa de Aceleração da Migração (MAP)

Um AWS programa que fornece suporte de consultoria, treinamento e serviços para ajudar as organizações a criar uma base operacional sólida para migrar para a nuvem e ajudar a compensar o custo inicial das migrações. O MAP inclui uma metodologia de migração para executar migrações legadas de forma metódica e um conjunto de ferramentas para automatizar e acelerar cenários comuns de migração.

migração em escala

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

fábrica de migração

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, o grupo de segurança e AWS a conta de destino.

padrão de migração

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

Avaliação de Portfólio para Migração (MPA)

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

Avaliação de Preparação para Migração (MRA)

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

estratégia de migração

A abordagem usada para migrar uma carga de trabalho para o. Nuvem AWS 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](#).

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 aplicativos no Nuvem AWS](#).

avaliação de preparação para modernização

Uma avaliação que ajuda a determinar a preparação para modernização das aplicações de uma organização. Ela identifica benefícios, riscos e dependências e determina o quão bem a organização pode acomodar o estado futuro dessas aplicações. O resultado da avaliação é um esquema da arquitetura de destino, um roteiro que detalha as fases de desenvolvimento e os marcos do processo de modernização e um plano de ação para abordar as lacunas identificadas. Para obter mais informações, consulte [Avaliação da prontidão para modernização de aplicativos no Nuvem AWS](#).

aplicações monolíticas (monólitos)

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

MAPA

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

MQTT

Consulte Transporte de [telemetria de enfileiramento de](#) mensagens.

classificação multiclasse

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

infraestrutura mutável

Um modelo que atualiza e modifica a infraestrutura existente para 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.

OPC-UA

Consulte [Comunicação de processo aberto — Arquitetura unificada](#).

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

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

acordo de nível operacional (OLA)

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

análise de prontidão operacional (ORR)

Uma lista de verificação de perguntas e 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.

tecnologia operacional (OT)

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

integração de operações (OI)

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

trilha organizacional

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

gerenciamento de alterações organizacionais (OCM)

Uma estrutura para gerenciar grandes transformações de negócios disruptivas de uma perspectiva de pessoas, cultura e liderança. O OCM ajuda as organizações a se prepararem e fazerem a transição para novos sistemas e estratégias, acelerando a adoção de alterações, abordando questões de transição e promovendo mudanças culturais e organizacionais. Na estratégia de AWS migração, essa estrutura é chamada de aceleração de pessoas, devido à velocidade de mudança 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

S3 Regiões da AWS, criptografia do lado do servidor com AWS KMS (SSE-KMS) e solicitações dinâmicas ao bucket S3. PUT DELETE

Identidade do acesso de origem (OAI)

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

ORR

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

OT

Veja a [tecnologia operacional](#).

VPC de saída (egresso)

Em uma arquitetura de AWS várias contas, uma VPC que gerencia conexões de rede que são iniciadas de dentro de um aplicativo. A [Arquitetura de Referência de AWS Segurança](#) recomenda configurar sua conta de rede com entrada, saída e inspeção VPCs para proteger a interface bidirecional entre seu aplicativo 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.

PLC

Consulte [controlador lógico programável](#).

AMEIXA

Veja o gerenciamento [do ciclo de vida do produto](#).

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 microserviço com base em padrões de acesso a dados e outros requisitos. Se seus microserviços tiverem a mesma tecnologia de armazenamento de dados, eles poderão enfrentar desafios de implementação ou apresentar baixa performance. Os microserviços 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 microserviços](#).

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.

principal (entidade principal)

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

privacidade por design

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

zonas hospedadas privadas

Um contêiner que contém informações sobre como você deseja que o Amazon Route 53 responda às consultas de DNS para um domínio e seus subdomínios em um 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.

gerenciamento do ciclo de vida do produto (PLM)

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

ambiente de produção

Veja o [ambiente](#).

controlador lógico programável (PLC)

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

encadeamento imediato

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

pseudonimização

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

publish/subscribe (pub/sub)

Um padrão que permite comunicações assíncronas entre microsserviços para melhorar a escalabilidade e a capacidade de resposta. Por exemplo, em um [MES](#) baseado em microsserviços, um microsserviço pode publicar mensagens de eventos em um canal no qual outros microsserviços possam se inscrever. O sistema pode adicionar novos microsserviços sem alterar o serviço de publicação.

Q

plano de consulta

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

regressão de planos de consultas

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

R

Matriz RACI

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

RAG

Consulte [Geração Aumentada de Recuperação](#).

ransomware

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

Matriz RASCI

Veja [responsável, 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 máximo período de tempo aceitável desde o último ponto de recuperação de dados.

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

objetivo de tempo de recuperação (RTO)

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

refatorar

Veja [7 Rs](#).

Região

Uma coleção de AWS recursos em uma área geográfica. Cada um Região da AWS é isolado e independente dos outros para fornecer tolerância a falhas, estabilidade e resiliência. Para obter mais informações, consulte [Especificar o que Regiões da AWS sua conta pode usar](#).

regressão

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

redefinir a hospedagem

Veja [7 Rs](#).

versão

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

realocar

Veja [7 Rs](#).

redefinir a plataforma

Veja [7 Rs](#).

recomprar

Veja [7 Rs](#).

resiliência

A capacidade de um aplicativo de resistir ou se recuperar de interrupções. [Alta disponibilidade](#) e [recuperação de desastres](#) são considerações comuns ao planejar a resiliência no. Nuvem AWS Para obter mais informações, consulte [Nuvem AWS Resiliência](#).

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](#).

Geração Aumentada de Recuperação (RAG)

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

alternância

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 o login único federado (SSO), para que os usuários possam fazer login AWS Management Console ou chamar as operações da AWS API sem que você precise criar um usuário no IAM para todos em sua organização. Para obter mais informações sobre a federação baseada em SAML 2.0, consulte [Sobre a federação baseada em SAML 2.0](#) na documentação do IAM.

SCADA

Veja [controle de supervisão e aquisição de dados](#).

SCP

Veja 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 [O que há em um segredo do Secrets Manager?](#) na documentação do Secrets Manager.

segurança por design

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

controle de segurança

Uma barreira de proteção técnica ou administrativa que impede, detecta ou reduz a capacidade de uma ameaça explorar uma vulnerabilidade de segurança. [Existem quatro tipos 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 EC2 da Amazon ou a rotação de credenciais.

Criptografia do lado do servidor

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

política de controle de serviços (SCP)

Uma política que fornece controle centralizado sobre as permissões de todas as contas em uma organização em AWS Organizations. SCPs defina barreiras ou estabeleça limites nas ações que um administrador pode delegar a usuários ou funções. Você pode usar SCPs como listas de permissão ou listas de negação para especificar quais serviços ou ações são permitidos ou proibidos. Para obter mais informações, consulte [Políticas de controle de serviço](#) na AWS Organizations documentação.

service endpoint (endpoint de serviço)

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

acordo de serviço (SLA)

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

indicador de nível de serviço (SLI)

Uma 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 com a qual você compartilha AWS pela segurança e conformidade na nuvem. AWS é responsável pela segurança da nuvem, enquanto você é responsável pela segurança na nuvem. Para obter mais informações, consulte o [Modelo de responsabilidade compartilhada](#).

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. Cada sub-rede fica alocada em uma única zona de disponibilidade.

controle de supervisão e aquisição de dados (SCADA)

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

symmetric encryption (criptografia simétrica)

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

testes sintéticos

Testar um sistema de forma que simule as interações do usuário para detectar possíveis problemas ou monitorar o desempenho. Você pode usar o [Amazon CloudWatch Synthetics](#) para criar esses testes.

prompt do sistema

Uma técnica para fornecer contexto, instruções ou diretrizes a um [LLM](#) para direcionar seu comportamento. Os prompts do sistema ajudam a definir o contexto e estabelecer regras para interações com os usuários.

T

tags

Pares de valores-chave que atuam como metadados para organizar seus recursos. 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 você pode usar para interconectar sua rede com VPCs a rede local. Para obter mais informações, consulte [O que é um gateway de trânsito](#) na AWS Transit Gateway documentação.

fluxo de trabalho baseado em troncos

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

Acesso confiável

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

tuning (ajustar)

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

equipe de duas pizzas

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

U

incerteza

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

Uma conexão entre duas VPCs que permite rotear o 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 [Estrutura de qualificação AWS da carga de](#) trabalho.

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.

aviso zero-shot

Fornecer a um [LLM](#) instruções para realizar uma tarefa, mas sem exemplos (fotos) que possam ajudar a orientá-la. O LLM deve usar seu conhecimento pré-treinado para lidar com a tarefa. A

eficácia da solicitação zero depende da complexidade da tarefa e da qualidade da solicitação. Veja também a solicitação [de algumas fotos](#).

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.