



Aplicação do AWS Well-Architected Framework para o Amazon Neptune

AWS Orientação prescritiva



AWS Orientação prescritiva: Aplicação do AWS Well-Architected Framework para o Amazon Neptune

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 desprestigie 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
Público-alvo	1
Objetivos	2
Excelência operacional	3
Automatize a implantação usando uma abordagem de IaC	3
Faça mudanças frequentes, pequenas e reversíveis	4
Antecipe a falha	4
Aprenda com todas as falhas operacionais	5
Use recursos de registro para monitorar atividades não autorizadas ou anômalas	5
Segurança	7
Implemente a segurança de dados	8
Proteja suas redes	9
Implemente autenticação e autorização	9
Confiabilidade	11
Entenda as cotas de serviço do Neptune	11
Entenda os padrões de implantação do Neptune	12
Gerencie e escale os clusters do Neptune	13
Gerencie backups e eventos de failover	14
Eficiência de desempenho	15
Entenda a modelagem gráfica	15
Otimizar consultas	16
Clusters do tamanho certo	18
Otimize gravações	19
Otimização de custo	21
Entenda os padrões de uso e os serviços necessários	21
Selecione recursos com atenção ao custo	22
Escolha a melhor configuração de instância do Neptune para sua carga de trabalho	23
Armazenamento e transferência de dados do tamanho certo	25
Sustentabilidade	26
Seleção de região da AWS	26
Consumo com base nos padrões de comportamento do usuário	27
Otimize os padrões de desenvolvimento e arquitetura de software	27
Recursos	29
Colaboradores	30

Histórico do documento	31
Glossário	32
#	32
A	33
B	36
C	38
D	41
E	45
F	47
G	49
H	50
eu	52
L	54
M	55
O	60
P	62
Q	65
R	66
S	69
T	73
U	74
V	75
W	75
Z	76
.....	lxxviii

Aplicação do AWS Well-Architected Framework para o Amazon Neptune

Amazon Web Services ([colaboradores](#))

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

Você pode criar soluções baseadas em gráficos na Amazon Web Services (AWS) usando o [Amazon Neptune](#). Este guia fornece orientação prescritiva para aplicar os princípios do [AWS Well-Architected Framework](#) ao planejar sua implantação do Neptune. [A aplicação do AWS Well-Architected Framework para o Amazon Neptune Analytics abrange o mesmo tópico para o mecanismo de análise gráfica Neptune](#).

O AWS Well-Architected Framework ajuda você a criar infraestruturas seguras, de alto desempenho, resilientes e eficientes para uma variedade de aplicativos e cargas de trabalho. Ele também fornece uma abordagem consistente para você avaliar arquiteturas e implementar projetos escaláveis.

O AWS Well-Architected Framework é construído em torno dos seis pilares a seguir:

- Excelência operacional
- Segurança
- Confiabilidade
- Eficiência de desempenho
- Otimização de custo
- Sustentabilidade

Este guia fornece informações sobre os pilares de design e as melhores práticas do Well-Architected Framework, além de considerações que você deve ter em mente ao implantar o Neptune no AWS.

Público-alvo

Este guia é destinado a engenheiros de dados, arquitetos de soluções e analistas de dados que projetam e implementam soluções que usam gráficos em AWS.

Objetivos

Este guia pode ajudar você e sua organização a fazer o seguinte:

- Escolha entre as opções de implantação e linguagens de consulta suportadas, com base em seu caso de uso e padrões de consulta.
- Siga os padrões de design do AWS Well-Architected que ajudarão a melhorar a resiliência e a segurança.
- Crie suas consultas para otimizar o desempenho e a economia de custos.
- Saiba como ser operacionalmente eficiente ao gerenciar seu cluster Neptune em produção.

Pilar Excelência operacional

O pilar de [excelência operacional](#) do AWS Well-Architected Framework se concentra na execução e monitoramento de sistemas e na melhoria contínua de processos e procedimentos. Isso inclui a capacidade de apoiar o desenvolvimento e executar cargas de trabalho de forma eficaz, obter informações sobre suas operações e melhorar continuamente os processos e procedimentos de suporte para agregar valor comercial. Você pode reduzir a complexidade operacional por meio de cargas de trabalho de autorrecuperação, que detectam e solucionam a maioria dos problemas sem intervenção humana. Você pode trabalhar para atingir essa meta seguindo as melhores práticas descritas nesta seção. Use as métricas APIs e os mecanismos do Amazon Neptune para responder adequadamente quando sua carga de trabalho se desviar do comportamento esperado.

Essa discussão sobre o pilar de excelência operacional se concentra nas seguintes áreas principais:

- Infraestrutura como código (IaC)
- Gerenciamento de alterações
- Estratégias de resiliência
- Gerenciamento de incidentes
- Relatórios de auditoria para conformidade
- Registro em log e monitoramento

Automatize a implantação usando uma abordagem de IaC

As melhores práticas para automatizar a implantação no Neptune usando o IaC incluem o seguinte:

- Aplique a infraestrutura como código (IaC) para implantar clusters do Neptune sempre que possível. Para uma configuração consistente do ambiente, use um [AWS CloudFormation](#) modelo ou o [HashiCorp Terraform](#) para criar todos os recursos necessários para seu cluster. [AWS Cloud Development Kit \(AWS CDK\)](#)
- Automatize os procedimentos operacionais do Neptune, como redimensionar instâncias, adicionar ou remover réplicas de leitura ou realizar failovers manuais em tabelas globais, sempre que possível.
- Armazene cadeias de conexão externamente do seu cliente. Use processos de extração, transformação e carregamento (ETL) para facilitar estratégias de implantação azul/verde, recuperação de desastres (DR) e migrações com tempo de inatividade quase zero para novos

clusters. As cadeias de conexão podem ser armazenadas no [AWS Secrets Manager](#) ou em qualquer local onde possam ser alteradas dinamicamente.

- Use tags para adicionar metadados aos seus recursos do Neptune e monitore o uso com base em tags. Para obter mais informações, consulte Como [marcar os recursos do Amazon Neptune](#).

Faça mudanças frequentes, pequenas e reversíveis

As recomendações a seguir se concentram em mudanças pequenas e reversíveis para minimizar a complexidade e reduzir a probabilidade de interrupção da carga de trabalho:

- Armazene modelos e scripts de IaC em um serviço de controle de origem, como GitHub ou GitLab.

Importante: não armazene AWS credenciais no controle de origem.

- Exija que as implantações de IaC usem um serviço de integração contínua e entrega contínua (CI/CD), como ou. [AWS CodeDeploy](#) [AWS CodeBuild](#) [Esses serviços compilam, testam e implantam código em um ambiente de não produção contendo um cluster efêmero do Neptune antes de impactar seu cluster Amazon Neptune de produção.](#)
- Teste as consultas de infraestrutura e aplicativos em um ambiente inferior antes de implantá-las na produção. Isso minimizará a probabilidade de uma interrupção e ajudará a garantir que eles funcionem bem com sua carga de trabalho e escala.

Antecipe a falha

Uma infraestrutura de autorrecuperação exemplifica a excelência operacional ao antecipar falhas e tentar resolver quaisquer problemas sem intervenção. As recomendações a seguir ajudam você a atingir essa maturidade com o Neptune:

- Crie um plano de monitoramento que use CloudWatch as métricas da Amazon para monitorar o uso da CPU e da memória da sua instância de banco de dados e entender os padrões de uso. Crie CloudWatch painéis e alarmes para as principais métricas e as respostas do cliente Neptune encontradas nos registros do seu aplicativo. Para obter mais informações sobre indicadores de alta ou baixa utilização da CPU, consulte [Usando CloudWatch para monitorar o desempenho da instância de banco de dados no Neptune na documentação](#) do Neptune.

Se você costuma receber out-of-memory exceções em suas consultas enquanto `FreeableMemory` estiver baixo, considere usar uma instância da família X2.

- Defina notificações para monitorar a integridade do cluster Neptune. Por exemplo, `BufferCacheHitRatio` deve ser constantemente alto (maior que 99,9%), enquanto `MainRequestQueuePendingRequests` deve ser constantemente baixo (idealmente 0, mas depende de seus requisitos e tolerância de latência).
- Considere usar réplicas de leitura para obter alta disponibilidade no Neptune. Você deve ter pelo menos duas réplicas de leitura em zonas de disponibilidade diferentes da instância do gravador para garantir que uma instância esteja sempre disponível para atender consultas de leitura durante um evento de failover.
- Dimensione automaticamente as réplicas de leitura com base nas métricas de utilização. Para obter mais informações, consulte [Escalabilidade automática do número de réplicas em um cluster de banco de dados Amazon Neptune](#).
- Teste o failover da instância do banco de dados para entender quanto tempo o processo leva para seu caso de uso.
- Se seu aplicativo precisar sobreviver a uma Região da AWS paralisação completa, considere o uso de [bancos de dados globais](#) como parte de seus planos de recuperação de desastres.

Aprenda com todas as falhas operacionais

Uma infraestrutura de autorrecuperação é um esforço de longo prazo que se desenvolve em iterações à medida que problemas raros ocorrem ou as respostas não são tão eficazes quanto o desejado. A adoção das seguintes práticas impulsiona o foco em direção a essa meta:

- Promova a melhoria aprendendo com todas as falhas.
- Compartilhe o que foi aprendido entre as equipes e a organização. Se várias equipes de uma organização usarem o Neptune, crie uma sala de bate-papo ou um grupo de usuários comum para compartilhar aprendizados e melhores práticas.

Use recursos de registro para monitorar atividades não autorizadas ou anômalas

Para observar padrões anômalos de desempenho e atividade, armazene os registros no Amazon CloudWatch Logs. Considere as seguintes práticas recomendadas:

- Ative o [registro de consultas lentas](#). Analise regularmente o registro e diagnostique por que certas consultas são lentas. Use os endpoints de explicação e perfil do Neptune [para Gremlin, SPARQL ou OpenCypher para obter](#) informações sobre por que essas consultas são lentas.
- [Ative os registros de auditoria do Neptune](#) e revise-os regularmente em busca de acesso não autorizado ou anomalias.
- Se você estiver usando registros de consultas lentas ou registros de auditoria, habilite a CloudWatch publicação no Logs. Isso ajudará você a evitar a falta de espaço em disco nas instâncias. As instâncias do Neptune têm capacidade limitada de armazenamento de registros e substituirão os arquivos de log antigos quando o espaço de log for excedido. CloudWatch Os registros oferecem suporte à retenção de registros a longo prazo. Os recursos aprimorados de monitoramento do CloudWatch Logs melhorarão sua capacidade de consultar registros e diagnosticar problemas.
- Para facilitar melhores ferramentas de análise para seus registros de auditoria, você pode configurar um cluster de banco de dados Neptune para publicar dados do registro de auditoria em um grupo de registros no Logs. CloudWatch Com o CloudWatch Logs, você pode realizar análises em tempo real dos dados de log, usar CloudWatch para criar alarmes e visualizar métricas, e usar o CloudWatch Logs para armazenar seus registros de log em um armazenamento altamente durável. Para obter mais informações, consulte [Publicação de registros do Neptune no Amazon Logs](#). CloudWatch
- Neptune suporta o registro de ações do plano de controle usando. AWS CloudTrail Para obter mais informações, consulte [Registrar chamadas de API do Amazon Neptune](#) com. AWS CloudTrail

Pilar Segurança

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve a segurança da nuvem e a segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que é executada Serviços da AWS no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia da AWS segurança como parte dos [programas de AWS conformidade](#). Para saber mais sobre os programas de conformidade que se aplicam ao Amazon Neptune, consulte [Serviços da AWS no escopo por programa de conformidade](#).
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS service (Serviço da AWS) que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade dos dados, os requisitos da empresa e as leis e regulamentos aplicáveis. Para obter mais informações sobre a privacidade de dados, consulte as [Perguntas frequentes sobre privacidade de dados](#). Para obter informações sobre proteção de dados na Europa, consulte o [Modelo de Responsabilidade AWS Compartilhada e a postagem no blog do GDPR](#).

O [pilar de segurança](#) ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Neptune. Os tópicos a seguir mostram como configurar o Neptune para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros Serviços da AWS que o ajudem a monitorar e proteger seus recursos do Neptune.

O pilar de segurança inclui as seguintes áreas principais de foco:

- Segurança de dados
- Segurança de rede
- Autenticação e autorização

Implemente a segurança de dados

Vazamentos e violações de dados colocam seus clientes em risco e podem causar um impacto negativo substancial em sua empresa. As práticas recomendadas a seguir ajudam a proteger os dados de seus clientes contra exposição inadvertida e maliciosa:

- Nomes de clusters, tags, grupos de parâmetros, funções AWS Identity and Access Management (IAM) e outros metadados não devem conter informações confidenciais ou sigilosas, pois esses dados podem aparecer em registros de faturamento ou diagnóstico.
- URIs ou links para servidores externos armazenados como dados no Neptune não devem conter informações de credenciais para validar solicitações.
- As instâncias criptografadas do Neptune fornecem uma camada adicional de proteção de dados, ajudando a proteger os dados contra acesso não autorizado ao armazenamento subjacente. É possível usar a criptografia do Neptune para aumentar a segurança dos dados das aplicações implantadas na nuvem. Você também pode usar a criptografia Neptune para atender aos requisitos de conformidade para dados em repouso.

Para habilitar a criptografia para uma nova instância de banco de dados Neptune, escolha Sim na seção Habilitar criptografia no console do Neptune (selecionada por padrão) ou defina a propriedade em [AWS::Neptune::DBCluster::StorageEncrypted](#) AWS CloudFormation. [Se a criptografia estiver habilitada, o Neptune usará a chave gerenciada da AWS do Amazon Relational Database Service \(Amazon RDS\) por padrão, ou você poderá criar uma chave gerenciada pelo cliente.](#) Para obter informações sobre como criar uma instância de banco de dados Neptune, consulte [Criação de um novo cluster de banco de dados Neptune](#). Para obter mais detalhes, consulte [Criptografando recursos do Neptune](#) em repouso. Seus instantâneos automatizados e manuais usam a mesma criptografia que você selecionou para seu cluster Neptune.

- Ao usar as linguagens SPARQL e OpenCypher, pratique técnicas adequadas de validação e parametrização de entrada para evitar a injeção de SQL e outras formas de ataques. Evite criar consultas que usem concatenação de strings com entrada fornecida pelo usuário. Use consultas parametrizadas ou instruções preparadas para passar com segurança os parâmetros de entrada para o banco de dados gráfico. [Para obter mais informações, consulte Exemplos de consultas parametrizadas do OpenCypher e Defesa de Injeção SPARQL.](#)
- Para a linguagem Gremlin, use [Gremlin Language Variants](#) em vez de passar diretamente scripts Gremlin baseados em strings para evitar possíveis problemas de injeção.

Proteja suas redes

Um cluster de banco de dados Amazon Neptune só pode ser criado em uma nuvem privada virtual (VPC) ativada. AWS Os endpoints do cluster de banco de dados Neptune são acessíveis somente dentro dessa VPC, geralmente a partir de [uma instância do Amazon Elastic Compute Cloud \(EC2Amazon\)](#) executada nessa VPC. Você pode proteger seus dados do Neptune limitando o acesso à VPC em que seu cluster de banco de dados Neptune está localizado. Para obter mais informações, consulte [Conectando-se ao seu gráfico do Amazon Neptune](#).

Para proteger seus dados em trânsito, o Neptune impõe conexões SSL por meio de HTTPS a qualquer instância ou [endpoint](#) de cluster usando protocolos e cifras seguros. O Neptune fornece certificados SSL para suas instâncias de banco de dados Neptune. Os certificados SSL Neptune são compatíveis somente com nomes de host de endpoint de cluster, endpoint de leitor e endpoint de instância. Se você estiver usando um balanceador de carga ou um servidor proxy (como [HAProxy](#)), deverá usar a terminação SSL e ter seu próprio certificado SSL no servidor proxy. A passagem SSL não funciona porque os certificados SSL fornecidos não correspondem ao nome do host do servidor de proxy. Para obter mais informações sobre como se conectar aos endpoints do Neptune com SSL, [consulte Usando o endpoint HTTP REST para se conectar a](#) uma instância de banco de dados do Neptune.

Implemente autenticação e autorização

Para controlar quem pode realizar ações de gerenciamento do Neptune nos clusters e instâncias de banco de dados do Neptune, use as credenciais do IAM. Quando você se conecta AWS usando credenciais do IAM, sua função do IAM deve ter políticas do IAM que concedam as permissões necessárias para realizar as operações de gerenciamento do Neptune. Certifique-se de seguir [o princípio do privilégio mínimo](#), concedendo somente as permissões necessárias para concluir uma tarefa. Para obter mais informações, consulte [Uso de diferentes tipos de políticas do IAM para controlar o acesso ao Neptune e Autenticação do IAM](#) usando credenciais temporárias.

Para controlar quem pode se conectar a um cluster do Neptune e consultar os dados, você pode usar o IAM para se autenticar na sua instância de banco de dados ou cluster de banco de dados Neptune. Se você habilitar a autenticação do IAM em um cluster de banco de dados Neptune, qualquer pessoa que acesse o cluster de banco de dados deverá primeiro ser autenticada. Para obter mais informações, consulte [Habilitar a autenticação do banco de dados do IAM no Neptune](#) para ver as etapas para habilitar a autenticação do IAM.

Quando a autenticação de banco de dados do IAM está habilitada, cada solicitação deve ser assinada usando o AWS Signature versão 4. Para entender como enviar solicitações assinadas para todos os endpoints do Neptune com a autenticação IAM ativada, [consulte Conexão e assinatura AWS](#) com o Signature versão 4. Muitas bibliotecas e ferramentas, como [awscurl](#), já oferecem suporte à AWS Signature Version 4.

[Para interagir com outras pessoas Serviços da AWS, o Amazon Neptune usa funções vinculadas ao serviço do IAM.](#) A função vinculada ao serviço é um tipo exclusivo de perfil do IAM vinculado diretamente ao Neptune. As funções vinculadas ao serviço são predefinidas pelo Neptune e incluem todas as permissões que o serviço exige para ligar para outras pessoas em seu nome. Serviços da AWS Para obter mais informações, consulte [Usando funções vinculadas ao serviço para o Neptune](#).

Pilar Confiabilidade

O [pilar de confiabilidade](#) engloba a capacidade de uma carga de trabalho realizar a função pretendida de forma correta e consistente quando se espera que isso aconteça. Isso inclui a capacidade de operar e testar a workload durante todo o ciclo de vida dela.

Uma workload confiável começa com as decisões iniciais de projeto que envolvem tanto o software quanto a infraestrutura. Suas decisões de arquitetura afetarão o comportamento da workload em todos os pilares do Well-Architected. Para atingir a confiabilidade, há padrões específicos que devem ser seguidos.

O pilar de confiabilidade se concentra nas seguintes áreas principais:

- Arquitetura de carga de trabalho, incluindo cotas de serviço e padrões de implantação
- Gerenciamento de alterações
- Gerenciamento de falhas

Entenda as cotas de serviço do Neptune

Um volume de [cluster do Neptune](#) pode crescer até um tamanho máximo de 128 tebibytes (TiB) em todos os países com Regiões da AWS suporte, exceto na GovCloud China, onde a cota é de 64 TiB.

A cota de 128 TiB é suficiente para armazenar aproximadamente 200-400 bilhões de objetos no gráfico. Em um gráfico de propriedades rotulado (LPG), um [objeto](#) é um nó, uma borda ou uma propriedade em um nó ou borda. Em um gráfico do Resource Description Framework (RDF), um objeto é um [quad](#).

Para qualquer cluster [Neptune Serverless](#), você define o número mínimo e máximo de Neptune Capacity Units (NCUs). Cada NCU consiste em 2 gibibytes (GiB) de memória e na vCPU e na rede associadas. Os valores mínimo e máximo de NCU se aplicam a qualquer instância sem servidor no cluster. O valor máximo de NCU mais alto que você pode definir é 128,0 NCUs e o mínimo mais baixo é 1,0. NCUs Otimize a faixa de NCU que funciona melhor para sua aplicação observando as CloudWatch métricas da Amazon ServerlessDatabaseCapacity e capturando a faixa em que você normalmente se NCUUtilization depara e correlacionando comportamentos ou custos indesejados dentro dessa faixa. Se você achar que sua carga de trabalho não é dimensionada com rapidez suficiente, aumente o mínimo NCUs para fornecer processamento suficiente para o aumento inicial durante a escalada.

Cada uma Conta da AWS tem cotas para cada região no número de recursos de banco de dados que você pode criar. Esses recursos incluem instâncias de banco de dados e clusters de banco de dados. Depois de atingir o limite de um recurso, as chamadas adicionais para criá-lo falham, com uma exceção. Algumas cotas são cotas flexíveis que podem ser aumentadas mediante solicitação. [Para obter uma lista de cotas compartilhadas entre o Amazon Neptune e o Amazon RDS, o Amazon Aurora e o Amazon DocumentDB \(com compatibilidade com o MongoDB\), além de links para solicitar aumentos de cotas quando disponíveis, consulte Cotas no Amazon RDS.](#)

Entenda os padrões de implantação do Neptune

Nos clusters de banco de dados Neptune, há uma instância de banco de dados primária e até 15 réplicas do Neptune. A instância de banco de dados primária suporta operações de leitura e gravação e realiza todas as modificações de dados no volume do cluster. As réplicas do Neptune se conectam ao mesmo volume de armazenamento da instância de banco de dados primária e oferecem suporte somente às operações de leitura. As réplicas do Neptune podem descarregar cargas de trabalho de leitura da instância de banco de dados primária.

Para obter alta disponibilidade, use réplicas de leitura. Ter uma ou mais instâncias de réplica de leitura disponíveis em diferentes zonas de disponibilidade pode aumentar a disponibilidade, pois as réplicas de leitura servem como destinos de failover para a instância primária. Se a instância do gravador falhar, o Neptune promoverá uma instância de réplica de leitura para se tornar a instância primária. Quando isso acontece, há uma breve interrupção (geralmente menos de 30 segundos) enquanto a instância promovida é reinicializada, durante a qual as solicitações de leitura e gravação feitas à instância primária falham com uma exceção. Para maior confiabilidade, considere duas réplicas de leitura em diferentes zonas de disponibilidade. Se a instância primária na Zona de Disponibilidade 1 ficar off-line, a instância na Zona de Disponibilidade 2 será promovida para primária, mas não poderá lidar com consultas enquanto isso acontece. Portanto, é necessária uma instância na Zona de Disponibilidade 3 para lidar com consultas de leitura durante a transição.

Se você estiver usando o Neptune Serverless, as instâncias de leitura e gravação em todas as zonas de disponibilidade aumentarão e diminuirão, independentemente umas das outras, dependendo da carga do banco de dados. Você pode definir o nível de promoção de uma instância do leitor como 0 ou 1 para que ela aumente e diminua junto com a capacidade da instância do gravador. Isso o prepara para assumir a carga de trabalho atual a qualquer momento.

Gerencie e escale os clusters do Neptune

Você pode usar o auto-scaling do [Neptune](#) para ajustar automaticamente o número de réplicas do Neptune em um cluster de banco de dados para atender aos requisitos de conectividade e carga de trabalho com base nos limites de utilização da CPU. Com o auto-scaling, seu cluster de banco de dados Neptune pode lidar com aumentos repentinos na carga de trabalho. Quando a carga de trabalho diminui, o auto-scaling remove réplicas desnecessárias para que você não pague pela capacidade não utilizada. Lembre-se de que a inicialização de uma nova instância pode levar até 15 minutos, portanto, o auto-scaling por si só não é uma solução suficiente para mudanças rápidas na demanda.

[Você pode usar o auto-scaling somente com um cluster de banco de dados Neptune que já tenha uma instância de gravador principal e pelo menos uma instância de réplica de leitura \(consulte Clusters e instâncias de banco de dados do Amazon Neptune\)](#). Além disso, todas as instâncias de réplica de leitura no cluster devem estar em um estado disponível. Se alguma réplica de leitura estiver em um estado diferente de disponível, o auto-scaling do Neptune não fará nada até que todas as réplicas de leitura no cluster estejam disponíveis.

Se você tiver mudanças rápidas na demanda, considere usar instâncias sem servidor. As instâncias sem servidor podem ser escaladas verticalmente em períodos curtos, enquanto o escalonamento automático é dimensionado horizontalmente em períodos mais longos. Essa configuração fornece escalabilidade ideal porque as instâncias sem servidor são dimensionadas verticalmente, enquanto o auto-scaling instancia novas réplicas de leitura para lidar com a carga de trabalho além da capacidade máxima de uma única instância sem servidor. Para obter mais informações sobre o escalonamento de capacidade do Amazon Neptune Serverless, [consulte Escalabilidade de capacidade em um cluster de banco de dados Neptune Serverless](#).

Se suas necessidades de escalabilidade mudarem em horários previsíveis, você poderá [programar mudanças nas](#) instâncias mínimas, máximas e limites para lidar melhor com essas necessidades variáveis. Lembre-se de agendar eventos de expansão com pelo menos 15 minutos de antecedência para permitir que essas instâncias fiquem on-line quando necessário.

Gerencia a configuração de banco de dados no Amazon Neptune usando [parâmetros](#) em um grupo de parâmetros. Grupos de parâmetros atuam como contêineres de valores de configuração do mecanismo que são aplicados a uma ou mais instâncias de bancos de dados. Ao modificar parâmetros de cluster em grupos de parâmetros, entenda a diferença entre parâmetros estáticos e dinâmicos e como e quando eles são aplicados. Use o endpoint de [status](#) para ver a configuração aplicada atual.

Gerencie backups e eventos de failover

O Neptune faz backup automático do volume do cluster e retém os dados de backup durante o período de retenção do backup. Os backups do Neptune são contínuos e incrementais para que você possa restaurar rapidamente em qualquer momento do período de retenção de backup. Você pode especificar um período de retenção de backup de 1 a 35 dias ao criar ou modificar um cluster de banco de dados.

Para reter um backup além do período de retenção do backup, você também pode tirar um instantâneo dos dados no volume do cluster. Armazenar snapshots gera taxas de armazenamento padrão do Neptune.

Quando você cria um snapshot do Amazon Neptune de um cluster de banco de dados, o Neptune cria um snapshot do volume de armazenamento do cluster, fazendo backup de todos os seus dados, não apenas de instâncias individuais. É possível criar um cluster de banco de dados restaurando pelo snapshot desse cluster de banco de dados. Ao restaurar o cluster de banco de dados, você fornece o nome do snapshot do cluster de banco de dados a partir do qual restaurar e, em seguida, fornece um nome para o novo cluster de banco de dados criado pela restauração.

Teste como seu sistema responde aos eventos de failover. Use a API Neptune [para forçar](#) um evento de failover. A [reinicialização com failover](#) é vantajosa quando você deseja simular uma falha de uma instância de banco de dados para testar ou restaurar as operações na zona de disponibilidade original após a ocorrência de um failover. Para obter mais informações, consulte [Configurando e gerenciando uma implantação Multi-AZ](#). Quando você reinicializa uma instância de gravador de banco de dados, ela passa para a réplica em espera. A reinicialização de uma réplica do Neptune não inicia um failover.

Projete seus clientes para garantir a confiabilidade. Teste seu comportamento durante eventos de failover. Implemente a lógica de repetição em seu cliente com a lógica de recuo exponencial. Exemplos de código que implementam essa lógica podem ser encontrados em [exemplos de AWS Lambda funções do Amazon Neptune](#).

Considere usar [AWS Backup](#) se você tiver um conjunto comum de requisitos de backup que você aplica em vários mecanismos de banco de dados.

Pilar Eficiência de performance

O [pilar de eficiência de desempenho](#) do AWS Well-Architected Framework se concentra em como otimizar o desempenho ao ingerir ou consultar dados. A otimização do desempenho é um processo incremental e contínuo do seguinte:

- Confirmando os requisitos de negócios
- Medindo o desempenho da carga de trabalho
- Identificação de componentes com baixo desempenho
- Ajustando os componentes para atender às necessidades de sua empresa

O pilar de eficiência de desempenho fornece diretrizes específicas para casos de uso que podem ajudar a identificar o modelo de dados gráficos e as linguagens de consulta corretos a serem usados. Também inclui as melhores práticas a serem seguidas ao ingerir e consumir dados do Amazon Neptune.

O pilar de eficiência de desempenho se concentra nas seguintes áreas principais:

- Modelagem gráfica
- Otimização de consultas
- Dimensionamento correto do cluster
- Otimização de gravação

Entenda a modelagem gráfica

Entenda a diferença entre os modelos Labeled Property Graph (LPG) e Resource Description Framework (RDF). Na maioria dos casos, é uma questão de preferência. No entanto, existem vários casos de uso em que um modelo é mais adequado do que o outro. Se você precisar de conhecimento do caminho que conecta dois nós em seu gráfico, escolha LPG. Se você quiser federar dados em clusters do Neptune ou em outros armazenamentos gráficos triplos, escolha RDF.

Se você estiver criando um aplicativo de software como serviço (SaaS) ou um aplicativo que exija multilocação, considere incorporar a separação lógica de locatários em seu modelo de dados em vez de ter um inquilino para cada cluster. Para obter esse tipo de design, você pode usar gráficos nomeados e estratégias de rotulagem do SPARQL, como prefixar identificadores de clientes em rótulos ou adicionar pares de valores-chave de propriedades representando identificadores de

inquilinos. Certifique-se de que sua camada de cliente injete esses valores para manter essa separação lógica.

O desempenho de suas consultas depende do número de objetos gráficos (nós, bordas, propriedades) que precisam ser avaliados no processamento de sua consulta. Dessa forma, o modelo gráfico pode ter um impacto significativo no desempenho do seu aplicativo. Use rótulos granulares sempre que possível e armazene somente as propriedades necessárias para obter a determinação ou a filtragem do caminho. Para obter maior desempenho, considere pré-calcular partes do gráfico, como criar nós de resumo ou bordas mais diretas conectando caminhos comuns.

Tente evitar navegar por nós que tenham um número anormalmente alto de bordas com o mesmo rótulo. Esses nós geralmente têm milhares de bordas (onde a maioria dos nós tem contagens de arestas na casa das dezenas). O resultado é uma complexidade muito maior de computação e dados. Esses nós podem não ser problemáticos em alguns padrões de consulta, mas recomendamos modelar seus dados de forma diferente para evitá-los, especialmente se você navegar pelo nó como uma etapa intermediária. Você pode usar [registros de consultas lentas](#) para ajudar a identificar consultas que navegam por esses nós. Você provavelmente observará métricas de latência e acesso a dados muito maiores do que seus padrões de consulta comuns, especialmente se usar o modo de [depuração](#).

Use um nó determinístico IDs para nós e bordas se seu caso de uso oferecer suporte a isso, em vez de usar o Neptune para atribuir valores de GUID aleatórios para. IDs Acessar nós por ID é o método mais eficiente.

Otimizar consultas

As linguagens OpenCypher e Gremlin podem ser usadas de forma intercambiável em modelos de GLP. Se o desempenho for uma das principais preocupações, considere usar as duas linguagens de forma intercambiável, pois uma pode ter um desempenho melhor do que a outra para padrões de consulta específicos.

O Neptune está em processo de conversão para seu [mecanismo de consulta alternativo \(DFE\)](#). [O OpenCypher é executado somente no DFE](#), mas as consultas Gremlin e SPARQL podem ser configuradas opcionalmente para execução no DFE usando anotações de consulta. Considere testar suas consultas com o DFE ativado e comparar o desempenho do seu padrão de consulta quando não estiver usando o DFE.

O Neptune é otimizado para consultas do tipo transacional que começam em um único nó ou conjunto de nós e se espalham a partir daí, em vez de consultas analíticas que avaliam o gráfico

inteiro. [Para suas cargas de trabalho de consultas analíticas, considere usar o AWS SDK para Pandas ou usar o neptune-export combinado com o Amazon EMR. AWS Glue](#)

Para identificar ineficiências e gargalos em seus modelos e consultas, use o `profile` e `explain` APIs para cada linguagem de consulta para obter explicações detalhadas sobre o plano de consulta e as métricas de consulta. [Para obter mais informações, consulte Perfil do Gremlin, explicação do OpenCypher e explicação do SPARQL.](#)

Entenda seus padrões de consulta. Se o número de arestas distintas em um gráfico se tornar grande, a estratégia de acesso padrão do Neptune pode se tornar ineficiente. As consultas a seguir podem se tornar bastante ineficientes:

- Consultas que navegam para trás pelas bordas quando nenhum rótulo de borda é fornecido.
- Cláusulas que usam esse mesmo padrão internamente, como `.both()` no Gremlin, ou cláusulas que eliminam nós em qualquer linguagem (o que exige a eliminação das bordas de entrada sem o conhecimento dos rótulos).
- Consultas que acessam valores de propriedades sem especificar rótulos de propriedades. Essas consultas podem se tornar bastante ineficientes. Se isso corresponder ao seu padrão de uso, considere ativar o [índice OSGP](#) (objeto, assunto, gráfico, predicado).

Use o [registro de consultas lentas](#) para identificar consultas lentas. Consultas lentas podem ser causadas por planos de consulta não otimizados ou por um número desnecessariamente grande de pesquisas de índice, o que pode aumentar os custos de E/S. A explicação e o perfil dos endpoints do Neptune [para](#) Gremlin, SPARQL [ou](#) OpenCypher podem ajudar você a entender por que essas consultas são lentas. As causas podem incluir o seguinte:

- Os nós com um número anormalmente alto de bordas em comparação com o nó médio no gráfico (por exemplo, milhares em comparação com dezenas) podem adicionar complexidade computacional e, portanto, maior latência e maior consumo de recursos. Determine se esses nós estão modelados corretamente ou se os padrões de acesso podem ser aprimorados para reduzir o número de bordas que devem ser percorridas.
- As consultas não otimizadas conterão um aviso de que etapas específicas não estão otimizadas. Reescrever essas consultas para usar etapas otimizadas pode melhorar o desempenho.
- Filtros redundantes podem causar pesquisas de índice desnecessárias. Da mesma forma, padrões redundantes podem causar pesquisas de índice duplicadas que podem ser otimizadas melhorando a consulta (veja `Index Operations - Duplication ratio` na saída do perfil).

- Alguns idiomas, como o Gremlin, não têm valores numéricos fortemente digitados e, em vez disso, usam a promoção de tipos. Por exemplo, se o valor for 55, Neptune procurará valores que sejam inteiros, longos, flutuantes e outros tipos numéricos equivalentes a 55. Isso resulta em operações adicionais. Se você souber que seus tipos coincidem com antecedência, você pode evitar isso usando uma [dica de consulta](#).
- Seu modelo gráfico pode impactar muito o desempenho. Considere reduzir o número de objetos que precisam ser avaliados usando rótulos mais granulares ou pré-calculando atalhos para caminhos lineares de vários saltos.

Se a otimização de consultas por si só não permitir que você atinja seus requisitos de desempenho, considere usar uma variedade de [técnicas de armazenamento em cache](#) com o Neptune para atingir esses requisitos.

Clusters do tamanho certo

Dimensione seu cluster de acordo com seus requisitos de simultaneidade e taxa de transferência. O número de consultas simultâneas que podem ser processadas por cada instância no cluster é igual a duas vezes o número de virtual CPUs (vCPUs) nessa instância. Consultas adicionais que chegam enquanto todos os segmentos de trabalho estão ocupados são colocadas em uma fila do lado do [servidor](#). Essas consultas são tratadas first-in-first-out (FIFO) quando os threads de trabalho são disponibilizados. A CloudWatch métrica da `MainRequestQueuePendingRequests` Amazon mostra a profundidade atual da fila para cada instância. Se esse valor estiver frequentemente acima de zero, considere [escolher uma instância](#) com mais CPUs v. Se a profundidade da fila exceder 8.192, o Neptune retornará um erro. `ThrottlingException`

Aproximadamente 65 por cento da RAM de cada instância é reservada para o cache de buffer. O cache do buffer contém o conjunto de dados de trabalho (não o gráfico inteiro; apenas os dados que estão sendo consultados). Para determinar qual porcentagem de dados está sendo obtida do cache de buffer em vez do armazenamento, monitore a CloudWatch métrica. `BufferCacheHitRatio` Se essa métrica geralmente cai abaixo de 99,9%, considere tentar uma instância com mais memória para determinar se ela diminui a latência e os custos de E/S.

As réplicas de leitura não precisam ter o mesmo tamanho da sua instância de gravação. No entanto, cargas de trabalho pesadas de gravação podem fazer com que réplicas menores fiquem atrasadas e sejam reinicializadas, pois não conseguem acompanhar a replicação. Portanto, recomendamos criar réplicas iguais ou maiores que a instância do gravador.

Ao usar o auto-scaling para suas réplicas de leitura, lembre-se de que pode levar até 15 minutos para colocar uma nova réplica de leitura on-line. Quando o tráfego do cliente aumenta de forma rápida, mas previsível, considere usar a [escalabilidade programada](#) para definir um número mínimo de réplicas de leitura mais alto para considerar esse tempo de inicialização.

As instâncias sem servidor oferecem suporte a vários casos de uso e cargas de trabalho diferentes. Considere instâncias provisionadas sem servidor para os seguintes cenários:

- Sua carga de trabalho flutua com frequência ao longo do dia.
- Você criou um novo aplicativo e não tem certeza de qual será o tamanho da carga de trabalho.
- Você está realizando desenvolvimento e testes.

É importante observar que as instâncias sem servidor são mais caras do que as instâncias provisionadas equivalentes com base em um dólar por GB de RAM. Cada instância sem servidor consiste em 2 GB de RAM junto com a vCPU e a rede associadas. Faça uma análise de custos entre suas opções para evitar contas inesperadas. Em geral, você economizará custos sem servidor somente quando sua carga de trabalho for muito pesada por apenas algumas horas por dia e quase zero no resto do dia ou se sua carga de trabalho flutuar significativamente ao longo do dia.

Otimize gravações

Para otimizar as gravações, considere o seguinte:

- O [Neptune Bulk](#) Loader é a maneira ideal de carregar inicialmente seu banco de dados ou anexar dados existentes. O carregador Neptune não é transacional e não pode excluir dados, portanto, não o use se esses forem seus requisitos.
- As atualizações transacionais podem ser feitas usando os idiomas de consulta compatíveis. Para otimizar as operações de E/S de gravação, grave dados em lotes de 50 a 100 objetos por confirmação. Um objeto é um nó, uma borda ou uma propriedade em um nó ou borda no GLP, ou um armazenamento triplo ou um quádruplo no RDF..
- Todas as operações de gravação do Neptune são de thread único para cada conexão. Ao enviar uma grande quantidade de dados para Neptune, considere ter várias conexões paralelas, cada uma gravando dados. Quando você escolhe uma instância provisionada pelo Neptune, o tamanho da instância é associado a um número de v. CPUs O Neptune cria dois threads de banco de dados para cada vCPU na instância, então comece com o dobro do número de CPUs v ao testar a paralelização ideal. As instâncias sem servidor escalam o número de v CPUs a uma taxa de aproximadamente uma para cada 4. NCUs

- Planeje e gerencie com [ConcurrentModificationException](#) eficiência todos os processos de gravação, mesmo que apenas uma única conexão esteja gravando dados a qualquer momento. Projete seus clientes para que sejam confiáveis quando `ConcurrentModificationExceptions` ocorrerem.
- Se você quiser excluir todos os seus dados, considere usar a [API de reinicialização rápida](#) em vez de emitir consultas de exclusão simultâneas. O último levará muito mais tempo e terá um custo substancial de E/S em comparação com o anterior.
- Se você quiser excluir a maioria dos seus dados, considere exportar os dados que deseja manter usando o [neptune-export](#) para carregar os dados em um novo cluster. Em seguida, exclua o cluster original.

Pilar Otimização de custos

O [pilar de otimização de custos](#) do AWS Well-Architected Framework se concentra em evitar custos desnecessários. As recomendações a seguir podem ajudá-lo a atender aos princípios de design de otimização de custos e às melhores práticas arquitetônicas do Amazon Neptune.

O pilar de otimização de custos se concentra nas seguintes áreas principais:

- Compreender os gastos ao longo do tempo e controlar a alocação de fundos
- Seleção de recursos do tipo e quantidade corretos
- Dimensionamento para atender às necessidades de negócios sem gastar demais

Entenda os padrões de uso e os serviços necessários

O Neptune é uma boa opção para sua carga de trabalho se seu modelo de dados tiver uma estrutura gráfica perceptível e suas consultas precisarem explorar relacionamentos e percorrer vários saltos. Um banco de dados gráfico não é adequado para os seguintes padrões:

- Principalmente consultas de salto único (considere se seus dados podem ser melhor representados como atributos de um objeto)
- Dados JSON ou BLOB armazenados como propriedades
- Consultas que se agregam em um conjunto de dados, como calcular a soma de uma propriedade numérica em um grande número de nós

Considere se o uso de vários bancos de dados específicos em conjunto para padrões de acesso específicos pode atender a todas as suas necessidades. Por exemplo:

- Uma API que exija navegações gráficas complexas menos frequentes junto com a recuperação altamente simultânea de propriedades para um único nó pode ser melhor apresentada usando um ou mais do Neptune, DynamoDB ou Amazon DocumentDB.
- Os bancos de dados relacionais podem coexistir com o Neptune para manter sua funcionalidade existente, mas use o Neptune somente para travessias de vários saltos que não funcionam nem escalam bem em bancos de dados relacionais.

Entenda os custos associados aos serviços que interagem e complementam o Neptune, incluindo o seguinte:

- Custos de armazenamento do Amazon Simple Storage Service (Amazon S3) para arquivos de dados sendo carregados em massa no Neptune
- Funções Lambda usadas para inserir ou atualizar consultas, consultas de leitura e processamento de fluxos do Neptune
- A camada de API criada no Neptune para interagir com o aplicativo cliente (em vez de ter conexões diretas com o banco de dados) em ou AWS AppSync
- AWS Glue trabalhos usados para transferir dados de e para Neptune
- Instâncias do Amazon Kinesis ou Amazon Managed Streaming for Apache Kafka (Amazon MSK) recebendo dados de streaming para ingestão quase em tempo real no Neptune.
- AWS Database Migration Service para migração de dados relacionais para Neptune
- Custos do Amazon SageMaker Runtime para notebooks Jupyter e modelos de aprendizado de máquina da Deep Graph Library

Selecione recursos com atenção ao custo

Os preços do [Neptune](#) são baseados no custo por hora da instância (ou nas unidades de computação do Neptune consumidas sem servidor), na E/S de dados e no uso do armazenamento. As instâncias representam, em média, 85% do custo total, portanto, o dimensionamento correto pode ter implicações de custo significativas. A melhor maneira de dimensionar corretamente as instâncias é testar o desempenho do aplicativo em várias instâncias e comparar os seguintes fatores:

- A `MainRequestQueuePendingRequests` CloudWatch métrica permanece em um número consistentemente baixo próximo de zero?
- A `BufferCacheHitRatio` CloudWatch métrica permanece igual ou superior a 99,9% na maioria das vezes?
- Quais são as curvas de custo e desempenho para custos de instância e custos associados de E/S de dados? Os custos de leitura de dados podem aumentar significativamente com uma instância subdimensionada que exige troca frequente de cache de buffer com armazenamento. `BufferCacheHitRatio` cairão com frequência nesses cenários.

Os custos das instâncias escalam linearmente com o tamanho dentro da mesma família de instâncias. O custo por hora da `db.r6i.2xlarge` instância é o dobro do da `db.r6i.xlarge`

instância e também tem o dobro da alocação de recursos. A `db.r6i.24xlarge` instância é 24 vezes o custo por hora da `db.r6i.xlarge` instância.

Estime o número de consultas simultâneas que você deve suportar. Você pode ter entre zero e quinze réplicas de leitura para processar consultas somente para leitura. Se seus requisitos variarem de acordo com a hora do dia, da semana ou do mês, você poderá usar várias instâncias menores para escalar de acordo com uma programação. Cada vCPU em uma instância fornece dois segmentos para lidar com consultas simultâneas. Três réplicas de `db.r6i.xlarge` leitura, com 4 vCPUs cada, podem lidar com 24 consultas simultâneas.

Se, em vez disso, seu volume de tráfego for medido em consultas por segundo (QPS), você deverá experimentar para determinar a latência média de suas consultas. O número de consultas por segundo que um cluster do Neptune pode suportar é igual a $vCPU \times 2 \times (1 \text{ second} / \text{average query latency})$. Por exemplo, se você tiver 4 vCPUs e uma latência de consulta de 100 milissegundos (0,1 segundo), $QPS = 4 \times 2 \times (1s / 0.1s) = 80 \text{ queries per second}$

As instâncias provisionadas são mais baratas do que as sem servidor para cargas de trabalho contínuas, estáveis e previsíveis. A tecnologia sem servidor oferece oportunidades para otimizar custos quando você tem uma carga de trabalho que exige um uso muito alto por apenas algumas horas por dia (por exemplo `db.r6i.4xlarge`) e, em seguida, quase nenhum tráfego pelo restante do dia (por exemplo, 1 unidade de computação Neptune). Uma instância sem servidor que aumenta a escala por algumas horas e depois diminui será mais barata do que usar uma instância provisionada `db.r6i.4xlarge` o dia todo.

Escolha a melhor configuração de instância do Neptune para sua carga de trabalho

[Para experiências de nível básico com o Neptune, você pode usar o nível gratuito da AWS.](#) As 750 horas gratuitas de uso `db.t3.medium` de uma `db.t4g.medium` instância são suficientes para que você tenha uma boa compreensão de Neptune em baixa escala. Seu cluster permanecerá após o término do período de teste gratuito, embora você seja cobrado pelo uso a partir desse momento.

As `db.t4g.medium` instâncias `db.t3.medium` and são boas para ambientes de desenvolvimento de baixo custo, mas lembre-se de que elas têm uma proporção menor de RAM para vCPU (2:1) do que as instâncias da família R (8:1) ou as instâncias da família X (16:1). Os perfis de desempenho podem ser diferentes dessas classes, especialmente em relação `OutOfMemoryExceptions` e quando as consultas navegam por uma parte significativa do gráfico. Para determinar se a última condição pode ser afetada, verifique a `BufferCacheHitRatio` CloudWatch métrica.

É altamente desaconselhável fazer qualquer teste de desempenho ou carga com instâncias da família T, pois você pode ter resultados inconsistentes que não são indicativos de um ambiente de produção.

As instâncias provisionadas oferecem a melhor combinação de custo e desempenho quando sua carga de trabalho é razoavelmente estável e previsível. Escolha o tamanho da instância com base na simultaneidade de solicitações necessária e na complexidade da consulta. Uma maior simultaneidade requer mais v. CPUs Uma maior complexidade de consulta requer mais RAM. Use a `MainRequestQueuePendingRequests` CloudWatch métrica para determinar o impacto da primeira (maior que zero representa mais solicitações simultâneas do que podem ser tratadas). Use a `BufferCacheHitRatio` CloudWatch métrica para determinar o impacto da última. Uma proporção que geralmente cai abaixo de 99,9% sugere que não há RAM suficiente para conter a parte funcional do gráfico que está sendo avaliada, o que resulta em trocas de cache mais frequentes. Se a família R de instâncias fornecer simultaneidade suficiente, mas não memória RAM suficiente, considere experimentar a família de instâncias X.

[Os casos de uso ideais para instâncias sem servidor estão descritos na documentação do Neptune.](#)

Se você não tiver certeza se provisionado ou sem servidor é o melhor para você, e o custo é sua principal preocupação, teste sua carga de trabalho sem servidor para determinar o número de NCUs usados e comparar o custo de provisioned () com serverless (). $N \text{ hours} \times \text{hourly provisioned cost} \text{ sum of NCUs} \times \text{hourly cost per NCU}$ Se você não tiver certeza sobre a instância de provisionamento de tamanho equivalente, uma NCU equivale a aproximadamente 2 GB de RAM e a vCPU e a rede associadas. Se sua instância provisionada for da família R6i, a proporção será de 1 vCPU por 8 GB de RAM, ou 4 NCUs, junto com a rede associada.

Ao usar a tecnologia sem servidor para instâncias primárias e de réplica, lembre-se de que as réplicas de leitura nos níveis de promoção 0 e 1 serão escaladas de acordo com a instância do gravador para que sejam escaladas adequadamente se ocorrer um evento de failover. NCUs Defina seus limites de NCU para essas instâncias com base em quais de suas instâncias, gravadoras ou leitoras, recebem mais tráfego.

Em ambientes em que o cluster não é necessário 24 horas por dia, 7 dias por semana, considere escrever scripts que desativarão as instâncias do Neptune quando não estiverem em uso e as iniciarão novamente antes de serem usadas. As instâncias do Neptune serão reiniciadas automaticamente a cada 7 dias para garantir que as atualizações de manutenção necessárias sejam aplicadas. Se você pretende deixar as instâncias desativadas por longos períodos, use um script semanal para desligá-las novamente.

Armazenamento e transferência de dados do tamanho certo

Consultas mais eficientes (por exemplo, consultas que precisam tocar menos nós, bordas e propriedades no gráfico) exigem menos transferência de E/S e, potencialmente, podem usar instâncias menores porque é necessário menos cache de buffer. Use o perfil ou explique os endpoints da sua linguagem de consulta para otimizar sua consulta e considere otimizar seu modelo gráfico para o desempenho da consulta.

O Neptune usa codificação de dicionário em cadeias grandes, e esse dicionário é otimizado para desempenho, não eficiência. Se você tiver cadeias de caracteres grandes BLOBs, JSON ou que mudam com frequência, considere armazená-las fora do Neptune no Amazon S3, no Amazon DynamoDB ou no Amazon DocumentDB e armazene somente uma referência dentro do nó do Neptune.

Em alguns casos, escolher um tamanho de instância maior pode ser mais barato. Se seus custos de E/S forem muito altos devido a uma baixa `BufferCacheHitRatio`, é possível que o cache de buffer maior reduza significativamente esse custo. Isso porque todos os dados caberiam no cache em vez de serem frequentemente trocados do armazenamento e incorrerem na taxa de transferência de E/S.

Neptune usa clonagem `copy-on-write`. Ao clonar para dividir um gráfico em vários fragmentos, pode ser mais eficiente não excluir os dados indesejados no cluster clonado, pois isso envolverá a criação de novas páginas de dados, resultando em maiores custos de armazenamento. Os dados que não foram alterados antes do evento de clonagem existirão em uma única página de dados compartilhada entre os dois clusters e serão cobrados somente por essa cópia única.

Não habilite o índice OSGP nem use instâncias R5d, a menos que você tenha testado para confirmar que elas fazem uma diferença substancial em sua carga de trabalho. Ambos foram projetados para cenários que ocorrem raramente e podem aumentar seus custos com ganhos mínimos ou inexistentes.

Pilar Sustentabilidade

O [pilar de sustentabilidade](#) se concentra em minimizar os impactos ambientais da execução de cargas de trabalho na nuvem. Os principais tópicos incluem um modelo de responsabilidade compartilhada para sustentabilidade, compreensão do impacto e maximização do uso para minimizar os recursos necessários e reduzir os impactos posteriores.

O pilar de sustentabilidade contém as seguintes áreas de foco principais:

- Seu impacto
- Metas de sustentabilidade
- Uso maximizado
- Antecipando e adotando ofertas de hardware e software novas e mais eficientes
- Uso de serviços gerenciados
- Redução do impacto a jusante

Este guia se concentra em seu impacto. Para obter mais informações sobre os outros princípios de design de sustentabilidade, consulte o [AWS Well-Architected Framework](#).

Suas escolhas e requisitos têm um impacto no meio ambiente. Se você puder escolher uma Região da AWS que tenha menor intensidade de carbono e se seus requisitos refletirem as necessidades reais da carga de trabalho em vez de apenas maximizar o tempo de atividade e a durabilidade, a sustentabilidade da carga de trabalho aumentará. As próximas seções discutem as melhores práticas e considerações ponderadas que terão um impacto ambiental positivo se adotadas em seu projeto de carga de trabalho e operações contínuas.

Seleção de região da AWS

Alguns Regiões da AWS estão perto de projetos de energia renovável da Amazon ou localizados onde a rede tem uma intensidade de carbono publicada que é menor do que outros. Considere o [impacto da sustentabilidade](#) em regiões que podem ser viáveis para sua carga de trabalho e cruze sua lista com as regiões em que o [Netuno está](#) disponível.

Consumo com base nos padrões de comportamento do usuário

O dimensionamento correto do consumo para corresponder ao tráfego e ao comportamento de seus usuários ajuda a AWS minimizar o impacto dos serviços no meio ambiente. Considere as seguintes práticas recomendadas ao projetar sua solução:

- Monitore CloudWatch métricas da `AmazonCPUUtilization`, `mainRequestQueuePendingRequests`, e `TotalRequestsPerSec` para determinar quando sua demanda é maior e menor, e garanta que seus recursos de cluster estejam do tamanho certo durante esses períodos.
- Automatize a interrupção de ambientes que não são de produção durante horas em que eles não estão sendo usados. Para obter mais informações, consulte a postagem do blog [Automatize a parada e a inicialização dos recursos ambientais do Amazon Neptune usando tags de recursos](#).
- Se seus padrões de tráfego variarem com frequência e de forma imprevisível, considere usar instâncias Neptune Serverless que aumentarão e diminuirão de acordo com a demanda, em vez de usar uma instância provisionada para tráfego de pico.
- Considere alinhar seus contratos de nível de serviço às metas de sustentabilidade, além das metas de continuidade dos negócios. A atenuação de requisitos como recuperação de desastres em várias regiões, alta disponibilidade ou retenção de backup de longo prazo, especialmente para ambientes que não sejam de produção ou cargas de trabalho não essenciais, pode reduzir a quantidade de recursos necessários para atingir essas metas.

Otimize os padrões de desenvolvimento e arquitetura de software

Para evitar desperdícios, otimize seus modelos e consultas e compartilhe recursos computacionais para usar todos os recursos disponíveis nas instâncias e clusters do Neptune. As melhores práticas específicas incluem:

- Faça com que os desenvolvedores compartilhem instâncias do Neptune e do aplicativo Jupyter Notebook em vez de cada um criar suas próprias. Forneça a cada desenvolvedor sua própria partição lógica em um único cluster do Neptune por meio do uso de estratégias de [particionamento multilocatário e crie pastas de](#) caderno separadas para cada desenvolvedor em uma única instância do Jupyter.
- Implemente padrões que maximizem o uso de recursos e minimizem o tempo ocioso, como threads paralelos para carregar dados e agrupar registros em lotes em uma transação maior.

-
- Otimize suas consultas e o modelo gráfico para minimizar os recursos necessários para calcular os resultados.
 - Para resultados de consultas do Gremlin, use o recurso de [cache de resultados](#) para minimizar os recursos gastos recalculando consultas paginadas ou frequentes.
 - Mantenha seus ambientes de Neptune atualizados. As versões mais recentes do Neptune oferecem suporte às instâncias EC2 mais recentes, como Graviton, que são mais eficientes. Eles também têm melhorias na otimização de consultas e correções de erros que reduzem a quantidade de recursos necessários para calcular suas consultas.

Recursos

Referências

- [AWS Well-Architected](#)
- [AWS Documentação do Well-Architected Framework](#)
- [Aplicação do AWS Well-Architected Framework para o Amazon Neptune Analytics](#)
- [Atualizações mais recentes do Amazon Neptune](#)
- [Melhores práticas: tirar o máximo proveito de Neptune](#)

Publicações no blog

- [Teste automatizado do acesso aos dados do Amazon Neptune com o Apache Gremlin TinkerPop](#)
- [Automatize a parada e a inicialização dos recursos ambientais do Amazon Neptune usando tags de recursos](#)
- [Controle de acesso refinado para ações do plano de dados do Amazon Neptune](#)
- [Use o raciocínio semântico para inferir novos fatos do seu gráfico RDF por meio da integração com o Amazon Neptune RDFox](#)
- [Crie uma solução de detecção de fraudes em tempo real usando o Amazon Neptune ML](#)

Cursos gratuitos do AWS Skill Builder

- [Conceitos básicos do Amazon Neptune](#)
- [Criação de aplicativos no Amazon Neptune](#)
- [Modelagem de dados para Amazon Neptune](#)

Colaboradores

Os colaboradores deste guia incluem:

- Brian O'Keefe, arquiteto principal da Neptune Solutions, AWS
- Abhishek Mishra, arquiteto sênior da Neptune Solutions, AWS
- Ganesh Sawhney, líder de equipe - Strategic Partner Success Solutions Architect, AWS
- Michael Havey, arquiteto sênior da Neptune Solutions, AWS
- Kevin Phillips, arquiteto da Neptune Solutions, AWS
- Melissa Kwok, arquiteta da Neptune Solutions, AWS
- Sakti Mishra, arquiteta principal de soluções AWS
- Javed Ali, arquiteto sênior de soluções, AWS

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	—	27 de setembro 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
- **Rever (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 descryptografia. É possível compartilhar a chave pública porque ela não é usada na descryptografia, mas o acesso à chave privada deve ser altamente restrito.

atomicidade, consistência, isolamento, durabilidade (ACID)

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

controle de acesso por atributo (ABAC)

A prática de criar permissões minuciosas com base nos atributos do usuário, como departamento, cargo e nome da equipe. Para obter mais informações, consulte [ABAC 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 hiper cuidados, a equipe de migração normalmente transfere a responsabilidade pelas aplicações para a equipe de operações de nuvem.

eu

laC

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.

IloT

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](#).

privilégio mínimo

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

mover sem alterações (lift-and-shift)

Veja [7 Rs](#).

sistema little-endian

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

LLM

Veja [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: rehoste 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 microsserviço com base em padrões de acesso a dados e outros requisitos. Se seus microsserviços tiverem a mesma tecnologia de armazenamento de dados, eles poderão enfrentar desafios de implementação ou apresentar baixa performance. Os microsserviços serão implementados com mais facilidade e alcançarão performance e escalabilidade melhores se usarem o armazenamento de dados mais bem adaptado às suas necessidades. Para obter mais informações, consulte [Habilitar a persistência de dados em microsserviços](#).

avaliação do portfólio

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

predicado

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

pressão de predicados

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

controle preventivo

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

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.