



Aplicando a AWS Well-Architected estrutura do Amazon Timestream para InfluxDB

AWS Recomendações



AWS Recomendações: Aplicando a AWS Well-Architected estrutura do Amazon Timestream para InfluxDB

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

Introdução	1
Público-alvo	2
Objetivos	2
Excelência operacional	3
Automatize a implantação usando uma abordagem de IaC	3
Fazer alterações frequentes, pequenas e reversíveis	4
Antecipar falha	4
Aprenda com todas as falhas operacionais	5
Use recursos de registro em log para monitorar atividades não autorizadas ou anômalas	5
Segurança	7
Implementar segurança de dados	7
Proteger suas redes	8
Implementar autorização e autenticação	8
Confiabilidade	10
Arquitetura de workload, incluindo cotas de serviço e padrões de implantação	10
Padrões de implantação	10
Gerencie e escale o Timestream para o InfluxDB	11
Eficiência de desempenho	13
Modelagem de dados de influxo e otimização de consultas	13
Otimize gravações	15
Otimização de custos	17
Entendendo os requisitos e custos do seu caso de uso	17
Seleção de recursos com atenção ao custo	17
Escalabilidade para atender às necessidades do negócio sem gastar em excesso	18
Dimensionamento correto do armazenamento e da transferência de dados	19
Sustentabilidade	20
Região da AWS seleção	20
Baseie o consumo de recursos nos padrões de comportamento do usuário	21
Otimizar os padrões de desenvolvimento e arquitetura de software	21
Recursos	23
Referências	23
Publicações no blog	23
Histórico do documento	24
.....	xxv

Aplicação do AWS Well-Architected Framework para Amazon Timestream para InfluxDB

Balwanth Bobilli, Amazon Web Services

Janeiro de 2025 ([histórico do documento](#))

Você pode criar time-series-based soluções na Amazon Web Services (AWS) usando o [Amazon Timestream](#). O Amazon Timestream oferece mecanismos de banco de dados de séries temporais totalmente gerenciados e personalizados para cargas de trabalho, desde consultas de baixa latência até ingestão de dados em grande escala. Com o [Amazon Timestream for InfluxDB](#), você pode executar bancos de dados InfluxDB de código aberto em aplicativos de séries temporais, como alertas AWS em tempo real e monitoramento da confiabilidade da infraestrutura, com tempos de resposta de milissegundos. O Timestream for InfluxDB fornece disponibilidade de até 99,9%.

Este guia fornece orientação prescritiva para aplicar os princípios do [AWS Well-Architected Framework](#) ao planejar sua implantação do Timestream para o InfluxDB. 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 seguintes seis pilares:

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

Este guia fornece informações dos pilares de design do AWS Well-Architected Framework. Considere usar essas melhores práticas ao implantar o Amazon Timestream para InfluxDB no AWS.

 Note

Alguns Serviços da AWS não estão disponíveis em todas as Regiões da AWS. Para saber a disponibilidade da região, consulte a página de [endpoints e cotas do serviço](#) na AWS documentação e escolha o link para o serviço.

Público-alvo

Este guia é destinado a engenheiros de dados, arquitetos de soluções e analistas de dados que projetam e implementam soluções para dados de séries temporais em AWS.

Objetivos

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

- Escolha entre as opções de implantação suportadas, realize gravações otimizadas e implemente acesso refinado.
- Siga os padrões de design do AWS Well-Architected que ajudam a melhorar a resiliência e a segurança.
- Crie suas consultas para obter um desempenho ideal.
- Aprenda a ser operacionalmente eficiente ao gerenciar sua instância Timestream for InfluxDB 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 para agregar valor comercial. O pilar da excelência operacional inclui a capacidade de apoiar o desenvolvimento e executar cargas de trabalho de forma eficaz, além de obter informações sobre suas operações.

Você pode reduzir a complexidade operacional por meio de workloads de autorrecuperação, que detectam e solucionam a maioria dos problemas sem intervenção humana. Para atingir esse objetivo, siga as melhores práticas descritas nesta seção. Use CloudWatch métricas [da Amazon](#) para Amazon Timestream para InfluxDB, o endpoint APIs de métricas nativas do InfluxDB, e mecanismos para responder quando sua carga de trabalho se desvia 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
- Registro e monitoramento para fins de auditoria

Automatize a implantação usando uma abordagem de IaC

As melhores práticas para automatizar a implantação no Timestream for InfluxDB usando o IaC incluem o seguinte:

- Aplique o IaC para implantar o Timestream para o InfluxDB 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 sua instância. [AWS Cloud Development Kit \(AWS CDK\)](#)
- Automatize os procedimentos operacionais do Timestream para o InfluxDB, como redimensionar instâncias.
- Use tags para adicionar metadados aos recursos do Timestream for InfluxDB e acompanhe o uso com base em tags. Para obter mais informações, consulte Como [marcar o Amazon Timestream](#) para o InfluxDB.

Fazer alterações frequentes, pequenas e reversíveis

As recomendações a seguir focam as mudanças pequenas e reversíveis para minimizar a complexidade e reduzir a probabilidade de interrupção da workload:

- Armazene modelos e scripts de IaC em um serviço de controle de origem, como ou. GitHub GitLab. Não armazene AWS credenciais no controle de origem.
- Exija que as implantações de IaC usem um serviço de integração e entrega contínuas (CI/CD), como o [AWS CodeDeploy](#) ou [AWS CodeBuild](#). Esses serviços compilam, testam e implantam código em um ambiente de não produção que contém uma instância efêmera do InfluxDB antes de afetar sua instância de produção do InfluxDB.
- Teste as consultas de infraestrutura e aplicações em um ambiente inferior antes de implantá-las na produção. Isso minimiza a probabilidade de uma interrupção e ajuda a garantir que eles funcionem bem com sua carga de trabalho e escala.

Antecipar 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 Timestream for InfluxDB:

- Use métricas para monitorar sua memória, CPU e uso de armazenamento. O CloudWatch pode ser configurado para lhe notificar quando os padrões de uso mudam ou quando você se aproxima da capacidade de implantação. Dessa maneira, é possível manter a disponibilidade e a performance do sistema.
- Amplie sua instância de banco de dados quando estiver se aproximando do limite de recursos. É preciso ter algum buffer de armazenamento e memória para acomodar aumentos imprevistos na demanda de seus aplicativos.
- Se a carga de trabalho do seu banco de dados exigir I/O mais do que você provisionou, a recuperação após um failover ou falha no banco de dados será lenta. Para aumentar a I/O capacity of a DB instance, migrate to a different DB instance that has higher I/O capacidade.
- Se seu aplicativo cliente estiver armazenando em cache os dados DNS de suas instâncias de banco de dados, defina um valor time-to-live (TTL) de menos de 30 segundos. O endereço IP subjacente de uma instância de banco de dados pode ser alterado após um failover. Armazenar

os dados do DNS em cache por um longo período pode levar a falhas de conexão. Sua aplicação pode tentar se conectar a um endereço IP que não está mais em serviço.

- Se seu aplicativo precisar sobreviver a uma Região da AWS paralisação completa, considere configurar a replicação ou gravar em uma região diferente como parte de seus planos de recuperação de desastres (DR). Entenda as limitações ao configurar a replicação. Para obter mais informações sobre replicação, consulte a documentação do [InfluxDB](#).

Aprenda com todas as falhas operacionais

Uma infraestrutura de autorrecuperação é um esforço de longo prazo que você desenvolve em iterações quando ocorrem problemas raros ou quando as respostas não são tão eficazes quanto você deseja. Para se concentrar em alcançar uma infraestrutura de autorrecuperação, adote as seguintes práticas:

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

Use recursos de registro em log para monitorar atividades não autorizadas ou anômalas

Para observar padrões anômalos de desempenho e atividade, considere as seguintes práticas:

- Habilite a [entrega de logs](#) para armazenar logs do InfluxDB no [Amazon Simple Storage Service \(Amazon S3\)](#). O InfluxDB registra informações de registro que podem ajudar a verificar o seguinte:
 - [Eventos da API do plano de dados](#)
 - Tempos de resposta
 - Detalhes de compactação
 - Quaisquer erros ou avisos críticos encontrados pelo sistema

Analise os registros em busca de acesso não autorizado ou anomalias. No geral, o registro fornece informações de diagnóstico para solução de problemas.

- O Timestream for InfluxDB suporta o registro de ações do plano de controle usando. AWS CloudTrail Para obter mais informações, consulte [Logging Timestream para chamadas de API do InfluxDB](#) com. AWS CloudTrail
- Você pode monitorar e **DiskUtilization** métricas **CPUUtilization** de **MemoryUtilization** Timestream/InfluxDB > < Namespace > in. CloudWatch

Para obter mais informações, consulte a documentação do [Timestream for InfluxDB](#).

Pilar Segurança

O [pilar de segurança](#) ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Timestream para o InfluxDB.

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

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

As seções a seguir mostram como configurar o Timestream para o InfluxDB para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outros Serviços da AWS que o ajudem a monitorar e proteger seu Timestream para recursos do InfluxDB.

Implementar segurança de dados

Violações e contaminações de dados colocam seus clientes em risco e podem causar um impacto negativo substancial em sua empresa. O [modelo de responsabilidade AWS compartilhada se aplica](#) à proteção de dados no Timestream for InfluxDB. As práticas a seguir ajudam a proteger os dados de seus clientes contra exposição inadvertida e maliciosa:

- Não inclua informações confidenciais em nomes de instâncias, tags, grupos de parâmetros, funções AWS Identity and Access Management (IAM) e outros metadados. Esses dados podem aparecer nos registros de faturamento ou diagnóstico.
- Use a criptografia para aumentar a proteção dos dados de seus aplicativos implantados na nuvem. As instâncias criptografadas fornecem uma camada adicional de proteção de dados, ajudando a proteger seus dados contra acesso não autorizado ao armazenamento subjacente. O Timestream para criptografia do InfluxDB está habilitado por padrão.
- Use a parametrização APIs sempre que possível.

Para obter mais informações sobre proteção e criptografia de dados, consulte a documentação do [Timestream for InfluxDB](#).

Proteger suas redes

Você pode criar um Timestream para a instância do InfluxDB somente em uma nuvem privada virtual (VPC) ativada. AWS Para proteger ainda mais sua instância, faça o seguinte:

- Restrinja o acesso público ao endpoint da instância fazendo o seguinte:
 - Escolha Não acessível publicamente na página Criar banco de dados do InfluxDB (selecionada por padrão).
 - Defina a [PubliclyAccessible](#) propriedade como `false` in AWS CloudFormation (`AWS::Timestream::InfluxDBInstance`).

Quando você escolhe Não acessível publicamente ou definido como `false`, `PubliclyAccessible` os endpoints da instância são acessíveis somente dentro da VPC. Os endpoints geralmente são acessados de uma instância do [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) em execução na mesma VPC da instância Timestream for InfluxDB.

- Use grupos de segurança para proteger ainda mais seu acesso à rede ao Timestream for InfluxDB dentro da VPC.
- Use SSL/TLS para se comunicar com AWS os recursos. O Timestream para InfluxDB requer TLS 1.2 e recomendamos o TLS 1.3.
- Conecte-se com segurança à sua instância seguindo as instruções em [Criação e conexão com uma instância Timestream for InfluxDB](#).
- [Estabeleça uma conexão privada entre sua VPC e o Amazon Timestream para endpoints de API do plano de controle do InfluxDB criando uma interface VPC endpoint.](#)

Para obter mais informações sobre segurança, consulte a documentação do [Timestream for InfluxDB](#).

Implementar autorização e autenticação

Use as credenciais do IAM para controlar as ações de gerenciamento no Timestream para instâncias do InfluxDB. Quando você se conecta ao Timestream for InfluxDB 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 Timestream para o InfluxDB. Certifique-se de seguir o princípio do privilégio mínimo, concedendo somente as permissões necessárias para concluir uma tarefa.

Para obter mais informações, consulte [Identity and Access Management for Amazon Timestream](#) for InfluxDB.

Você pode usar as ações do [Amazon Timestream for InfluxDB na política do IAM para](#) controlar quem pode gerenciar sua instância do Timestream for InfluxDB.

[Para fornecer controle de acesso refinado para seus dados armazenados em sua instância Amazon Timestream for InfluxDB, forneça aos usuários tokens de API do InfluxDB.](#)

Para interagir com outras pessoas Serviços da AWS, o Amazon Timestream para InfluxDB usa funções vinculadas ao serviço IAM. Uma função vinculada ao serviço é um tipo exclusivo de função do IAM que está vinculada diretamente ao Timestream for InfluxDB. As funções vinculadas ao serviço são predefinidas pelo Timestream for InfluxDB 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 a serviços para Amazon Timestream para](#) InfluxDB.

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 carga de trabalho durante todo o seu ciclo de vida.

A configuração de uma carga de trabalho confiável começa com decisões antecipadas de design para software e infraestrutura. Suas decisões de arquitetura afetarão o comportamento da workload em todos os pilares do Well-Architected. Para obter confiabilidade, você deve seguir padrões específicos.

O pilar de confiabilidade foca as seguintes áreas principais:

- Arquitetura de workload, incluindo cotas de serviço e padrões de implantação
- Gerenciando e escalando instâncias do InfluxDB

Arquitetura de workload, incluindo cotas de serviço e padrões de implantação

Cada uma das Contas da AWS tem cotas para os recursos oferecidos em cada uma das Regiões da AWS. Por exemplo, cada região tem uma [cota de Timestream para instâncias do InfluxDB, independentemente do tamanho da instância](#). Depois de atingir o número máximo de instâncias em uma região, as chamadas adicionais para criar instâncias falham com uma exceção. Um volume de armazenamento da instância Timestream for InfluxDB pode crescer até um tamanho máximo de 16 terabytes (TB) em todas as regiões da AWS.

Padrões de implantação

Para [alta disponibilidade](#) e suporte de failover para instâncias Timestream for InfluxDB, você pode usar implantações Multi-AZ com uma única instância de banco de dados em espera. Esse tipo de implantação é chamado de implantação de instância de banco de dados multi-AZ. O Amazon Timestream para InfluxDB usa a tecnologia de failover da Amazon. Em uma implantação de instância de banco de dados multi-AZ, o Amazon Timestream provisiona e mantém automaticamente uma réplica em espera síncrona em outra zona de disponibilidade. Para fornecer redundância de dados, a instância de banco de dados primária é replicada de forma síncrona entre as zonas de disponibilidade para a réplica em espera.

A execução de uma instância de banco de dados com alta disponibilidade pode fornecer disponibilidade durante falhas na instância de banco de dados ou interrupção da zona de disponibilidade. Se uma interrupção não planejada de sua instância de banco de dados resultar de um defeito na infraestrutura, o Amazon Timestream for InfluxDB mudará automaticamente para a réplica em espera. O tempo de conclusão do failover depende da atividade do banco de dados e de outras condições no momento em que a instância de banco de dados primária se tornou indisponível.

Em geral, os tempos de failover variam de 60 a 120 segundos. No entanto, grandes transações com dados de alta cardinalidade ou um longo processo de recuperação com requisitos de pré-aquecimento podem aumentar o tempo de failover. Após a conclusão do failover, talvez seja necessário mais tempo antes que o console do Timestream reflita a nova zona de disponibilidade.

Se seu aplicativo precisar permanecer disponível durante uma Região da AWS paralisação completa, considere configurar a replicação ou a gravação em uma região diferente como parte de seus planos de recuperação de desastres (DR). No entanto, antes de configurar a replicação, certifique-se de compreender as limitações. Para obter mais informações, consulte a documentação do [InfluxDB](#).

O Amazon Timestream para InfluxDB faz backups internos periodicamente e os retém por 24 horas para oferecer suporte à disponibilidade e durabilidade. Os instantâneos são tirados durante as exclusões e mantidos por 30 dias para apoiar as restaurações. Para acessá-los ou usá-los, crie um caso em [AWS Support](#).

Gerencie e escale o Timestream para o InfluxDB

O Timestream for InfluxDB suporta classes de instância que são ideais para executar cargas de trabalho com uso intenso de memória em bancos de dados InfluxDB de código aberto. As diferentes [classes de instância db.influx](#) têm limites em vCPUs, memória, armazenamento e largura de banda de rede. Para escolher a classe de instância que atende aos requisitos de latência de gravação e consulta do seu aplicativo, observe as `DiskUtilization` métricas da Amazon CloudWatch `CPUUtilization` e durante o teste. `MemoryUtilization` Você pode [escalar suas instâncias para cima e para baixo](#) com base em seus requisitos de carga de trabalho. O Timestream for InfluxDB fornece vários níveis de armazenamento pré-configurados com IOPS e taxa de transferência ideais para diferentes tipos de cargas de trabalho. Escolha o que funciona melhor para sua carga de trabalho com base em seus requisitos.

Se suas necessidades de escalabilidade mudarem em horários previsíveis, você pode usar uma [AWS Lambda função](#) ou um agendador personalizado e executar uma API ou SDK para aumentar e diminuir a escala com algum tempo de buffer.

Você gerencia sua configuração do InfluxDB no Timestream for InfluxDB usando parâmetros em um grupo de parâmetros. Os grupos de parâmetros atuam como um contêiner para as opções de configuração do InfluxDB que são aplicadas a uma ou mais instâncias de banco de dados. Ao modificar parâmetros em grupos de parâmetros, entenda a diferença entre parâmetros estáticos e dinâmicos e como e quando eles são aplicados. Para ver a configuração aplicada atual, use a ação [GetDbParameterGroup](#) da API.

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 de baixo desempenho
- Ajustando os componentes para atender às necessidades de sua empresa

O pilar de eficiência de desempenho fornece diretrizes que podem ajudar você a escolher um modelo de dados de alto desempenho. O pilar de eficiência de desempenho inclui as melhores práticas de otimização de consulta e gravação.

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

- Modelagem de dados de influxo e otimização de consultas
- Otimização de gravação

Modelagem de dados de influxo e otimização de consultas

Projetar um esquema eficaz é crucial para otimizar o desempenho e os recursos de consulta de dados de séries temporais no InfluxDB. Comece escolhendo as tags e os campos corretos. O InfluxDB indexa as tags, portanto, o mecanismo de consulta não precisa escanear todos os registros em uma medição para localizar o valor da tag. Isso significa que consultar tags é mais eficiente do que consultar campos. Para compactar e armazenar dados, o mecanismo de armazenamento agrupa valores de campo por chave de série e, em seguida, ordena esses valores de campo por hora. Uma chave de série é definida por medição, chave e valor de tag e chave de campo. Para obter mais informações sobre design de dados, consulte a documentação do [InfluxDB](#).

O mecanismo de armazenamento usa um formato de dados Time-Structured Merge Tree (TSM). Para obter mais informações sobre o formato de dados TSM, consulte a documentação do [InfluxDB](#).

Imagine que você está coletando dados

(timestamp,host_id,region,cpu,memory,network_in_bytes,network_out_bytes,disk_io)

como parte de um caso de DevOps uso. As tags, incluindo a data e hora do registro, fornecem contexto para ajudar a identificar quem, o quê, quando e onde de um registro. As tags são usadas para organizar e categorizar dados e filtrar dados como parte de uma consulta.

As `region` tags `host_id` e são etiquetas ideais para organizar e categorizar o caso de DevOps uso. Essas colunas ajudam a filtrar os dados de um determinado host ou a executar análises com base na coluna da região.

As medidas fornecem a base para realizar cálculos matemáticos (como calcular totais, médias e diferenças na taxa de variação) e análises quantitativas em seus dados. Portanto, `cpu`, `memory`, `network_in_bytes`, `network_out_bytes`, e `disk_io` capture métricas importantes relacionadas ao DevOps que estão mudando com o tempo. Você pode usar essas métricas para realizar várias análises, como calcular a CPU e a memória em diferentes hosts. Você pode usar esses valores métricos para tomar decisões baseadas em dados que ajudam a evitar interrupções na produção e a realizar o planejamento da infraestrutura.

A cardinalidade é a combinação de valores de tag exclusivos. Procure manter a cardinalidade o mais baixa possível. Se seu aplicativo exigir um identificador exclusivo para cada ponto de dados, use valores de campo em vez de valores de tag. Isso resultará em uma latência de consulta significativamente melhor. Um bom design de esquema pode evitar uma alta cardinalidade de séries, resultando em consultas com melhor desempenho. Se você notar que as leituras e gravações de dados estão diminuindo ou se quiser saber como a cardinalidade afeta o desempenho, consulte a documentação do [Timestream](#) for InfluxDB.

Se seu aplicativo emitir objetos JSON, converta-os em colunas individuais (tags ou campos) e carregue as colunas no InfluxDB. O InfluxDB foi projetado para dados de séries temporais, portanto, organizar seus dados com colunas individuais é a melhor prática para aproveitar ao máximo os recursos do serviço.

Uma única instância OSS do InfluxDB v2.7 suporta aproximadamente 20 buckets do InfluxDB que estão sendo ativamente gravados ou consultados em todas as organizações. Mais de 20 buckets podem afetar adversamente o desempenho. Existem limites em algumas opções de configuração do InfluxDB e algumas opções que você pode configurar com base no seu caso de uso. Valide a configuração com base na carga de trabalho do aplicativo durante a fase de teste. As retenções de dados são configuradas no nível do bucket, portanto, dados com diferentes requisitos de retenção de dados devem ser armazenados em diferentes buckets. Para obter mais informações sobre as opções de configuração, consulte a documentação do [Timestream for InfluxDB](#).

Armazene dados em valores de tag ou valores de campo, não em chaves de tag, chaves de campo ou medidas. Se você criar seu esquema para armazenar dados em valores de tag e campo, suas consultas serão mais fáceis de escrever e mais eficientes. Para obter mais práticas recomendadas sobre modelagem de dados, consulte [Design para desempenho](#).

Use [as tarefas do InfluxDB](#) para pré-agregar dados, carregar os dados em diferentes medidas ou compartimentos e gerar dados para painéis e visualizações a partir deles.

O InfluxDB OSS expõe um `/metrics` [endpoint](#) que retorna métricas de desempenho, recursos e uso formatadas no formato de exposição de texto simples do Prometheus. Use os modelos do InfluxDB para configurar o [monitoramento e os alertas para detectar](#) problemas de forma proativa, como alta latência de consulta, degradação da taxa de transferência de gravação ou picos de uso de recursos.

O Timestream for InfluxDB fornece armazenamento incluído no Influx IO. Selecionar o tamanho de IOPS apropriado pode acelerar significativamente a execução da consulta. Isso é especialmente útil para consultas que precisam digitalizar grandes quantidades de dados ou lidar com uma grande variedade de solicitações. Em algumas situações, uma combinação de escalar a instância e aprimorar o IOPS pode ser necessária para alcançar as melhorias de desempenho desejadas.

Recomendamos combinar os ambientes de desenvolvimento e produção (classe de instância, tipo de armazenamento, configurações). Teste as mudanças no ambiente inferior para cada versão antes de passar para a produção. Nos volumes de armazenamento incluídos no Influx IO, o Timestream for InfluxDB fornece três níveis de armazenamento pré-configurados com IOPS ideais (3.000, 12.000, 16.000) e taxa de transferência necessária para diferentes tipos de cargas de trabalho. A maioria dos casos de uso exige menos de 3.000 IOPS. Escolha 12.000 ou 16.000 somente se o teste de desempenho indicar a necessidade de alto IOPS. Para obter mais informações, consulte a [seção Configuração](#) na documentação do Timestream for InfluxDB.

Otimize gravações

Para otimizar as gravações no InfluxDB, recomendamos gravar dados em lotes de 5.000 linhas de protocolo de linha por solicitação para minimizar a sobrecarga da rede. Para um melhor desempenho, classifique as tags por chave em ordem lexicográfica antes de gravar os pontos de dados. Usar a maior precisão de tempo possível para registros de data e hora, em vez de nanossegundos, também pode melhorar o desempenho. Ativar a compactação gzip é outra forma de acelerar as gravações e reduzir a largura de banda da rede. Na configuração do plug-in de `influxdb_v2` saída em seu `telegraf.conf` arquivo, defina a `content_encoding` opção

comogzip. A implementação dessas otimizações pode melhorar significativamente o desempenho e a eficiência da gravação de dados no InfluxDB. Para obter mais práticas recomendadas de gravação do InfluxDB, consulte [Otimizar gravações](#) no InfluxDB.

O desempenho de gravação do InfluxDB geralmente está intimamente ligado ao IOPS disponível. Ao gravar dados, o InfluxDB precisa realizar um número significativo de operações de E/S para armazenar os dados. Quando você aumenta o IOPS, o InfluxDB pode processar mais gravações por segundo.

Pilar Otimização de custos

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

O pilar de otimização de custos foca as seguintes áreas principais:

- Entendendo os requisitos e custos do seu caso de uso
- Seleção de recursos com atenção ao custo
- Escalabilidade para atender às necessidades do negócio sem gastar em excesso
- Dimensionamento correto do armazenamento e da transferência de dados

Entendendo os requisitos e custos do seu caso de uso

Recomendamos não usar o Timestream para InfluxDB nos seguintes casos de uso:

- Se seu modelo de dados tem dados relacionais, o Timestream for InfluxDB não é a solução certa.
- Se você não puder usar filtros de tempo em suas consultas, o Influx examinará todas as séries, o que é ineficiente.

Seleção de recursos com atenção ao custo

Os custos da [instância do InfluxDB](#) são baseados em uma taxa horária para as horas em que sua instância é executada. As instâncias representam, em média, 85% do custo geral de execução de um banco de dados AWS, portanto, o dimensionamento correto pode ter implicações de custo significativas. A melhor maneira de dimensionar corretamente as instâncias é testar o desempenho do aplicativo:

- O `CPUUtilization` e `MemoryUtilization` estão constantemente altos ou baixos?
- Qual é o equilíbrio entre preço e desempenho?

Os custos das instâncias são escalonados linearmente. O custo por hora da `db.influx.2xlarge` instância é o dobro do da `db.influx.xlarge` instância, embora também tenha o dobro da

alocação de recursos. A `db.influx.16xlarge` instância é 16 vezes o custo por hora da `db.influx.xlarge` instância.

Estime o número de gravações e leituras da sua carga de trabalho para um período específico (segundo, minuto, hora ou dia). O Timestream para instâncias do InfluxDB suporta 50.000 a mais de 500.000 gravações por segundo e de 10 a 100 consultas por segundo (QPS) com base no tipo de instância. Por exemplo, `db.influx.2xlarge` normalmente suporta até 150.000 gravações por segundo e aproximadamente 25 QPS. Com um modelo de dados eficiente e consultas eficientes, ele pode superar esse desempenho. Se seus requisitos variarem de acordo com a hora do dia, da semana ou do mês, você pode programar o aumento e a redução da escala fazendo o seguinte:

- [Crie e agende uma AWS Lambda função](#).
- Use um agendador personalizado e execute uma API ou um SDK para [aumentar e diminuir a escala](#) com algum tempo de buffer.

Escalabilidade para atender às necessidades do negócio sem gastar em excesso

Para experimentação básica com o Timestream for InfluxDB, você pode usar e.

`db.influx.medium` `db.influx.large` Essas instâncias são grandes o suficiente para que você tenha experiência com o Timestream for InfluxDB antes de investir em instâncias maiores.

As `db.influx.large` instâncias `db.influx.medium` e são boas para ambientes de desenvolvimento de baixo custo. No entanto, eles têm uma RAM menor (8 GiB e 16 GiB), menos v (CPUs 1 vCPU e 2 vCPUs) e desempenho de rede de até apenas 10 GB. Nem todas as cargas de trabalho são adequadas para essas classes de instância. Monitore `CPUUtilization` `MemoryUtilization` e aumente ou diminua a escala conforme necessário. Geralmente, há uma proporção consistente entre a memória e a vCPU. A classe de instância `db.influx` tem uma `memory-to-vCPU` proporção semelhante à classe de instância `r7g` do Amazon EC2. É altamente recomendável executar testes de end-to-end desempenho ou carga antes de começar a produção.

A modelagem eficiente de dados, a gravação em lote e as consultas otimizadas exigem menos uso de memória e computação. Quando menos recursos são necessários, você pode potencialmente usar instâncias menores.

Dimensionamento correto do armazenamento e da transferência de dados

Para armazenar dados, use as seguintes práticas recomendadas:

- Armazene somente dados de séries temporais no Timestream for InfluxDB.
- Defina a retenção apropriada no bucket do InfluxDB para que os dados anteriores à retenção sejam excluídos e os fragmentos sejam periodicamente compactados automaticamente. Para obter mais informações, consulte a documentação do [InfluxDB](#).
- Otimize o uso do disco para futuras gravações.
- Exclua todos os buckets do InfluxDB que não sejam necessários para suas cargas de trabalho. O InfluxDB suporta exclusões. Você pode realizar limpezas programadas se isso for adequado ao seu caso de uso.

Para transferência de dados, recomendamos implantar seu aplicativo da mesma Região da AWS forma que sua instância de banco de dados Timestream for InfluxDB para evitar sobrecarga de rede entre regiões. Também pode haver taxas de transferência de dados. Para obter mais informações sobre transferência de dados, consulte a [página de preços](#).

Pilar Sustentabilidade

O [pilar de sustentabilidade](#) se concentra em minimizar os impactos ambientais da execução de cargas de trabalho na nuvem. O pilar de sustentabilidade contém as seguintes áreas de foco principais:

- Entendendo seu impacto
- Objetivos de sustentabilidade
- Maximizando o uso para minimizar os recursos
- Antecipação e adoção de ofertas de software e hardware novos e mais eficientes
- Usando serviços gerenciados
- Reduzindo o impacto a jusante

Este guia se concentra em entender 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. Para aumentar a sustentabilidade da sua carga de trabalho, faça o seguinte:

- Escolha Regiões da AWS aqueles com menor intensidade de carbono.
- Dimensione seus recursos para refletir as necessidades reais da carga de trabalho, em vez de maximizar o tempo de atividade e a durabilidade.
- Otimize seu modelo de dados e maximize o uso de recursos computacionais.

As próximas seções discutem as práticas que você pode adotar para reduzir o impacto ambiental no projeto da carga de trabalho e nas operações contínuas.

Região da AWS seleção

Alguns Regiões da AWS estão perto de projetos de energia renovável da Amazon ou localizados onde a intensidade de carbono publicada pela rede é menor do que outras redes. Avalie as regiões com base em suas [metas de sustentabilidade](#) e seus requisitos de carga de trabalho. Em seguida, faça referência cruzada à sua lista de regiões viáveis com as regiões em que o [Timestream for InfluxDB](#) está disponível.

Baseie o consumo de recursos nos padrões de comportamento do usuário

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

- Monitore CloudWatch as métricas da Amazon, como `CPUUtilization` e `MemoryUtilization` para determinar quando sua demanda é maior e menor. Certifique-se de que os recursos da sua instância estejam do tamanho certo durante esses períodos.
- Considere alinhar seus acordos de serviço às metas de sustentabilidade, além das metas de continuidade dos negócios. A redução de requisitos, como recuperação de desastres em várias regiões, alta disponibilidade ou retenção de backup a longo prazo, pode reduzir a quantidade de recursos necessários para atingir essas metas. Ambientes que não são de produção e cargas de trabalho não essenciais oferecem oportunidades para reduzir os requisitos.

Otimizar os padrões de desenvolvimento e arquitetura de software

Para evitar desperdício, otimize seu modelo de dados e suas consultas. Compartilhe recursos de computação para usar todos os recursos disponíveis na instância Timestream for InfluxDB. Recomendamos implementar as seguintes melhores práticas:

- Incentive as equipes de desenvolvedores a compartilhar a pilha Timestream for InfluxDB para melhor uso dos recursos sempre que possível.
- Implemente padrões que maximizem o uso de recursos e minimizem o tempo ocioso. Exemplos de padrões incluem o uso de threads paralelos para carregar dados e agrupar registros em lotes em uma transação maior.
- Otimize suas consultas e o modelo de dados do InfluxDB para minimizar os recursos necessários para calcular os resultados.
- Use [as tarefas do InfluxDB](#) para pré-agregar os dados e reduzir a digitalização dos mesmos dados brutos por diferentes usuários para visualização ou painel.
- Mantenha seus ambientes Timestream for InfluxDB atualizados. As versões mais recentes do Timestream for InfluxDB suportam as instâncias EC2 mais recentes, como Graviton, que são mais eficientes. As versões mais recentes do banco de dados também incluem 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](#)
- [Documentação do Amazon Timestream para InfluxDB](#)
- [Documentação do InfluxDB OSS v2](#)

Publicações no blog

- [Use o script de migração do AWS InfluxDB para migrar seus dados do InfluxDB OSS 2.x para o Amazon Timestream for InfluxDB](#)
- [Execute e gerencie bancos de dados InfluxDB de código aberto com o Amazon Timestream](#)

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	—	28 de janeiro de 2025

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.