aws

Guia do usuário

Amazon CloudWatch

Amazon CloudWatch: Guia do usuário

Table of Contents

O que é o Amazon CloudWatch?

O Amazon CloudWatch monitora os recursos da Amazon Web Services (AWS) e as aplicações executadas na AWS em tempo real. Você pode usar o CloudWatch para coletar e monitorar métricas, que são as variáveis que é possível medir para avaliar seus recursos e suas aplicações.

A página inicial do CloudWatch exibe automaticamente métricas sobre cada produto da AWS usado. Também crie painéis personalizados para exibir métricas sobre os aplicativos personalizados e exibir coleções personalizadas de métricas escolhidas.

É possível criar alarmes que observem métricas e enviem notificações ou façam alterações automaticamente nos recursos que você está monitorando quando um limite é violado. Por exemplo, você pode monitorar o uso de CPU e de leituras e gravações de disco de suas instâncias do Amazon EC2 e usar esses dados para determinar se deve iniciar instâncias adicionais para lidar com o aumento de carga. Você também pode usar esses dados para interromper instâncias subutilizados para economizar dinheiro.

Com o CloudWatch, você obtém visibilidade no âmbito do sistema da utilização de recursos, da performance das aplicações e da integridade operacional.

Acessar o CloudWatch

Você pode acessar o CloudWatch usando qualquer um destes métodos:

- Console do Amazon CloudWatch: https://console.aws.amazon.com/cloudwatch/
- AWS CLI: para obter mais informações, consulte Configurar a AWS Command Line Interface no Manual do usuário do AWS Command Line Interface.
- API do CloudWatch: ara mais informações, consulte a Referência da API do Amazon CloudWatch.
- SDKs da AWS: para mais informações, consulte Ferramentas para a Amazon Web Services.

Serviços relacionados da AWS

Os seguintes serviços são usados com o Amazon CloudWatch:

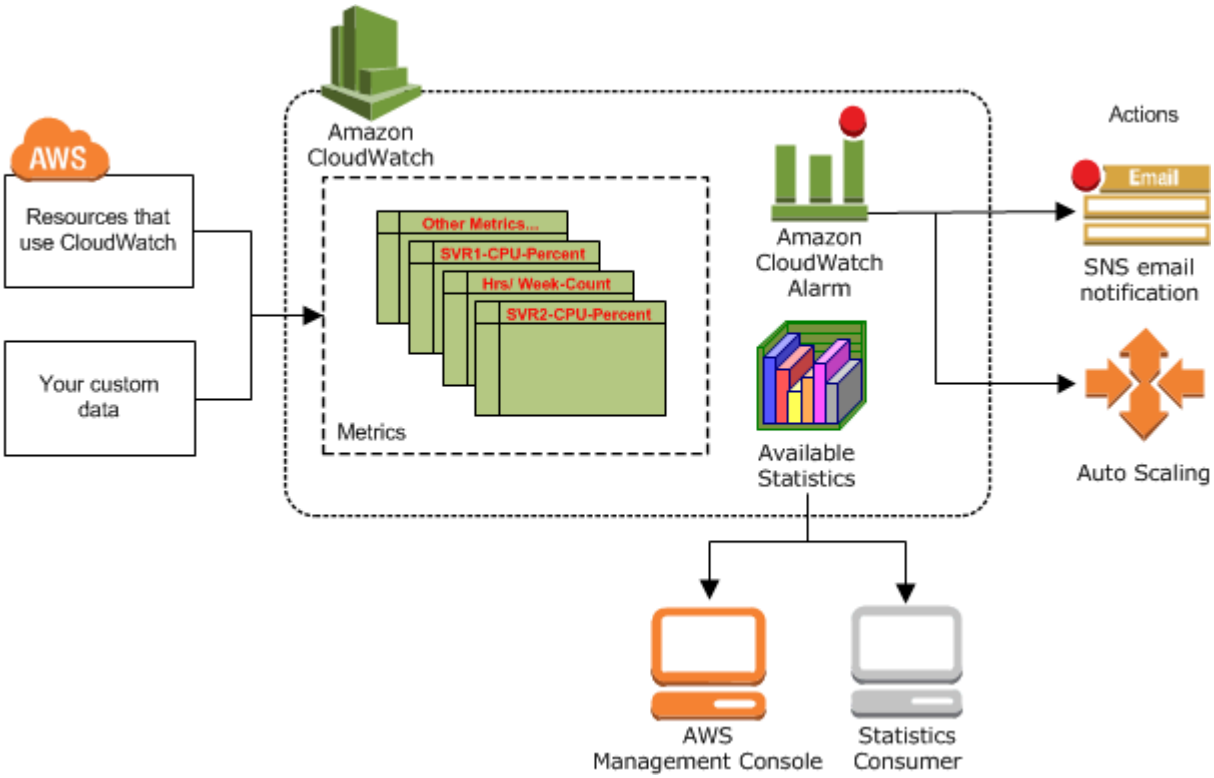
- O Amazon Simple Notification Service (Amazon SNS) é um serviço da Web que coordena e gerencia a entrega ou o envio de mensagens a endpoints ou clientes inscritos. Use o Amazon SNS

com o CloudWatch para enviar mensagens quando um limite de alarme for atingido. Para ter mais informações, consulte Configurar notificações do Amazon SNS.

- O Amazon EC2 Auto Scaling permite iniciar ou terminar automaticamente instâncias do Amazon EC2 com base em políticas definidas pelo usuário, verificações de status de integridade e cronogramas. É possível usar um alarme do CloudWatch com o Amazon EC2 Auto Scaling para escalar suas instâncias do EC2 sob demanda. Para obter mais informações, consulte Escalabilidade dinâmica no Manual do usuário do Amazon EC2 Auto Scaling.
- O AWS CloudTrail permite monitorar chamadas feitas para a API do Amazon CloudWatch para a sua conta, incluindo as chamadas feitas pelo AWS Management Console, pela AWS CLI e por outros serviços. Quando o registro do CloudTrail é ativado, o CloudWatch grava os arquivos de log no bucket do Amazon S3 especificado quando você configurou o CloudTrail. Para ter mais informações, consulte Registrar chamadas de API do Amazon CloudWatch com o AWS CloudTrail.
- O AWS Identity and Access Management (IAM) é um serviço da Web que ajuda a controlar seguramente o acesso de seus usuários aos recursos da AWS. Use o IAM para controlar quem pode usar os recursos da AWS (autenticação) e quais recursos os usuários podem usar e de que maneira (autorização). Para ter mais informações, consulte Gerenciamento de Identidade e Acesso para o Amazon CloudWatch.

Como o Amazon CloudWatch funciona

O Amazon CloudWatch é basicamente um repositório de métricas. Um produto da AWS, como o Amazon EC2, coloca métricas no repositório, e você recupera as estatísticas com base nessas métricas. Se você colocar suas próprias métricas personalizadas no repositório, poderá recuperar as estatísticas sobre essas métricas.



Você pode usar as métricas para calcular estatísticas e apresentar os dados graficamente no console do CloudWatch. Para obter mais informações sobre os outros recursos da AWS que geram e enviam métricas ao CloudWatch, consulte Produtos da AWS que publicam métricas do CloudWatch.

Você pode configurar ações de alarmes para interromper, iniciar ou terminar uma instância do Amazon EC2 quando determinados critérios são atendidos. Além disso, é possível criar alarmes que iniciam ações do Amazon EC2 Auto Scaling e do Amazon Simple Notification Service (Amazon SNS) em seu nome. Para obter mais informações sobre como criar alarmes do CloudWatch, consulte Alarmes.

Os recursos de computação em nuvem da AWS são abrigados em instalações de datacenters de alta disponibilidade. Para fornecer escalabilidade e confiabilidade adicionais, cada instalação de datacenter está localizada em uma determinada área geográfica, conhecida como Região. Cada região é projetada para ser completamente isolada das outras regiões, a fim de atingir o máximo possível de isolamento de falha e estabilidade. As métricas são armazenadas separadamente nas regiões, mas é possível usar a funcionalidade entre regiões do CloudWatch para agregar estatísticas de regiões diferentes. Para obter mais informações, consulte Console do CloudWatch entre contas e entre regiões e Regiões e endpoints no Referência geral da Amazon Web Services.

Conceitos do Amazon CloudWatch

Os seguintes conceitos e terminologia são fundamentais para o entendimento e uso do Amazon CloudWatch::

- Namespaces
- Indicadores
- Dimensões
- Resolução
- Estatísticas
- Percentis
- Alarmes

Para obter informações sobre as cotas de serviço para métricas, alarmes, solicitações de API e notificações por e-mail de alarmes do CloudWatch, consulte Cotas de serviço do CloudWatch.

Namespaces

Namespace é um contêiner para as métricas do CloudWatch. As métricas em namespaces diferentes são isoladas umas das outras, portanto, as métricas de aplicativos diferentes não são agregadas por engano nas mesmas estatísticas.

Não há um namespace padrão. Você deve especificar um namespace para cada ponto de dados que publicar no CloudWatch. Você pode especificar um nome de namespace ao criar uma métrica. Esses nomes devem conter caracteres ASCII válidos e ter 255 caracteres ou menos. Os caracteres passíveis de uso são: caracteres alfanuméricos (0-9, A-Z, a-z), ponto (.), hífen (-), sublinhado (_), barra (/), hash (#) e dois pontos (:) e o caractere de espaço. Um namespace deve conter pelo menos um caractere que não seja espaço em branco.

Os namespaces da AWS usam a seguinte convenção de nomenclatura: `AWS/service`. Por exemplo, o Amazon EC2 usa o namespace `AWS/EC2`. Para obter uma lista de namespaces da AWS, consulte Produtos da AWS que publicam métricas do CloudWatch.

Indicadores

As métricas são um conceito fundamental do CloudWatch. Uma métrica representa um conjunto de pontos de dados ordenados ao longo do tempo que são publicados no CloudWatch. Considere

uma métrica como variável a ser monitorada, e os pontos de dados representando os valores dessa variável ao longo do tempo. Por exemplo, o uso de CPU de determinada instância do EC2 é uma métrica fornecida pelo Amazon EC2. Os pontos de dados em si podem ser provenientes de qualquer aplicativo ou atividade de negócios da qual você coleta dados.

Por padrão, muitos produtos da AWS fornecem métricas gratuitas para recursos (como instâncias do Amazon EC2, volumes do Amazon EBS e instâncias de banco de dados do Amazon RDS). Por uma taxa, você também pode habilitar o monitoramento detalhado de alguns recursos, como instâncias do Amazon EC2 ou publicar suas próprias métricas de aplicações. Para métricas personalizadas é possível adicionar os pontos de dados em qualquer ordem e em qualquer taxa que você escolher. É possível recuperar estatísticas sobre os pontos de dados como um conjunto ordenado de dados da série de tempo.

As métricas existem somente na Região em que são criadas. Não é possível excluir métricas, mas elas expirarão automaticamente depois de 15 meses se novos dados não forem publicados nelas. Os pontos de dados com mais de 15 meses expiram de forma contínua; à medida que novos pontos de dados são adicionados, os dados com mais de 15 meses são descartados.

As métricas são definidas exclusivamente por um nome, um namespace e zero ou mais dimensões. Cada ponto de dados em uma métrica tem um time stamp e (opcionalmente) uma unidade de medida. É possível recuperar estatísticas do CloudWatch de qualquer métrica.

Para obter mais informações, consulte Visualizar métricas disponíveis e Publicar métricas personalizadas do .

Carimbos de data/hora

Cada ponto de dados de métrica deve ser associado a um time stamp. O time stamp pode ser de até duas semanas no passado e até duas horas no futuro. Se você não fornecer um carimbo de data/hora, o CloudWatch criará um carimbo de data/hora para você com base no momento em que o ponto de dados foi recebido.

Os time stamps são objetos `dateTime` com a data completa além de horas, minutos e segundos (por exemplo, 2016-10-31T23:59:59 Z). Para obter mais informações, consulte dateTime. Embora não seja necessário, recomendamos que você use o Tempo Universal Coordenado (UTC). Quando você recupera estatísticas a partir do CloudWatch, todos os horários são exibidos em UTC.

Os alarmes do CloudWatch verificam as métricas com base na hora atual em UTC. As métricas personalizadas enviadas ao CloudWatch com carimbos de data/hora diferentes do horário UTC atual

podem fazer com que os alarmes exibam o estado Dados insuficientes ou resultem em alarmes atrasados.

Retenção de métricas

O CloudWatch mantém os dados de métrica da seguinte forma:

- Pontos de dados com um período inferior a 60 segundos ficam disponíveis por 3 horas. Estes pontos de dados são métricas personalizadas de alta resolução.
- Pontos de dados com um período de 60 segundos (1 minuto) ficam disponíveis por 15 dias
- Pontos de dados com um período de 300 segundos (5 minutos) ficam disponíveis por 63 dias
- Pontos de dados com um período de 3.600 segundos (1 hora) ficam disponíveis por 455 dias (15 meses)

Os pontos de dados que inicialmente são publicados com um período menor são agregados para um armazenamento de longo prazo. Por exemplo, se você coletar dados usando um período de 1 minuto, os dados permanecerão disponíveis por 15 dias com resolução de 1 minuto. Depois de 15 dias estes dados ainda estarão disponíveis, mas estarão agregados e poderão ser recuperados apenas com uma resolução de 5 minutos. Depois de 63 dias, os dados estarão ainda mais agregados e disponíveis com uma resolução de 1 hora.

> **Note**
>
> AS métricas que não tiverem novos pontos de dados nas últimas duas semanas não serão exibidas no console. Elas também não serão exibidas quando você digitar o nome da métrica ou os nomes de dimensão na caixa de pesquisa na guia Todas as métricas do console e não serão retornadas nos resultados de um comando list-metrics. A melhor maneira de recuperar essas métricas é com os comandos get-metric-data ou get-metric-statistics na AWS CLI.

Dimensões

Uma dimensão é um par de nome/valor que faz parte da identidade de uma métrica. Você pode atribuir até 30 dimensões a uma métrica.

Cada métrica tem características específicas que a descrevem, e você pode considerar dimensões como categorias para essas características. Dimensões ajudam a projetar uma estrutura para seu plano de estatísticas. Como as dimensões fazem parte do identificador exclusivo de uma métrica,

sempre que você adicionar um par de nome/valor exclusivo a uma de suas métricas, estará criando uma nova variação daquela métrica.

Produtos da AWS que enviam dados ao CloudWatch anexam dimensões a cada métrica. Você pode usar dimensões para filtrar os resultados que o CloudWatch retorna. Por exemplo, você pode obter estatísticas para uma determinada instância do EC2, especificando a `InstanceId` dimensão ao procurar métricas.

Para métricas produzidas por determinados produtos da AWS, como o Amazon EC2, o CloudWatch pode agregar dados entre dimensões. Por exemplo, se você procurar por métricas no namespace `AWS/EC2`, mas não especificar as dimensões, o CloudWatch agregará todos os dados da métrica especificada para criar a estatística que você solicitou. O CloudWatch não agrega suas métricas personalizadas entre as dimensões.

Combinações de dimensões

O CloudWatch trata cada combinação única de dimensões como uma métrica distinta, mesmo que as métricas tenham o mesmo nome. Você só pode recuperar estatísticas usando combinações de dimensões que publicou especificamente. Quando você recuperar estatísticas, especifique os mesmos valores para o namespace, nome da métrica e parâmetros de dimensão que foram usados quando as métricas foram criadas. Também é possível especificar os horários de início e de término para o CloudWatch usar na agregação.

Por exemplo, suponha que você publique quatro métricas distintas denominadas ServerStats no namespace DataCenterMetric com as seguintes propriedades:

```
Dimensions: Server=Prod, Domain=Frankfurt, Unit: Count, Timestamp:
 2016-10-31T12:30:00Z, Value: 105
Dimensions: Server=Beta, Domain=Frankfurt, Unit: Count, Timestamp:
 2016-10-31T12:31:00Z, Value: 115
Dimensions: Server=Prod, Domain=Rio,       Unit: Count, Timestamp:
 2016-10-31T12:32:00Z, Value: 95
Dimensions: Server=Beta, Domain=Rio,       Unit: Count, Timestamp:
 2016-10-31T12:33:00Z, Value: 97
```

Se você publicar somente quatro métricas, poderá recuperar as estatísticas para estas combinações de dimensões:

- `Server=Prod,Domain=Frankfurt`
- `Server=Prod,Domain=Rio`

- `Server=Beta,Domain=Frankfurt`
- `Server=Beta,Domain=Rio`

Não é possível recuperar estatísticas para as dimensões a seguir ou se você não especificar dimensões. (A exceção é usar a função matemática de métrica SEARCH que pode recuperar estatísticas para várias métricas. Para mais informações, consulte Usar expressões de pesquisa em gráficos.)

- `Server=Prod`
- `Server=Beta`
- `Domain=Frankfurt`
- `Domain=Rio`

Resolução

Cada métrica é um dos seguintes:

- Resolução padrão, com dados de granularidade de um minuto
- Resolução alta, com dados de granularidade de um segundo

Por padrão, as métricas produzidas por serviços da AWS têm resolução padrão. Quando você publica uma métrica personalizada, pode defini-la com resolução padrão ou alta. Quando você publica uma métrica de alta resolução, o CloudWatch a armazena com uma resolução de 1 segundo. Você pode ler e recuperar essa métrica no período de 1 segundo, 5 segundos, 10 segundos, 30 segundos ou em qualquer múltiplo de 60 segundos.

As métricas de alta resolução podem também dar a você insight mais imediato da atividade de subminuto da seu aplicativo. Lembre-se de que cada chamada `PutMetricData` de uma métrica personalizada é cobrada. Portanto, chamar `PutMetricData` com mais frequência em uma métrica de alta resolução pode resultar em tarifas mais altas. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

Se você definir um alarme em uma métrica de alta resolução, pode especificar um alarme de alta resolução com um período de 10 ou 30 segundos ou pode definir um alarme regular com um período de qualquer múltiplo de 60 segundos. Há uma tarifa maior para alarmes de alta resolução com um período de 10 ou 30 segundos.

Estatísticas

Estatísticas são agregações de dados de métrica ao longo de períodos especificados. O CloudWatch fornece estatísticas com base nos pontos de dados de métrica fornecidos por seus dados personalizados ou por outros produtos da AWS para o CloudWatch. As agregações são feitas usando o namespace, o nome da métrica, as dimensões e a unidade de medida do ponto de dados no período especificado.

Para obter as definições detalhadas das estatísticas compatíveis com o CloudWatch, consulte Definições de estatísticas do CloudWatch.

Unidades

Cada estatística tem uma unidade de medida. Exemplo de unidades incluem `Bytes`, `Seconds`, `Count` e `Percent`. Para ver a lista completa das unidades compatíveis com o CloudWatch, consulte o tipo de dados MetricDatum na Referência da API do Amazon CloudWatch.

Você pode especificar uma unidade ao criar uma métrica personalizada. Se você não especificar uma unidade, o CloudWatch usará `None` como a unidade. Unidades ajudam a atribuir significado conceitual aos seus dados. Embora o CloudWatch não vincule nenhum significado a uma unidade internamente, outras aplicações podem extrair informações semânticas com base na unidade.

Os pontos de dados de métrica que especificam uma unidade de medida são agregados separadamente. Quando você obtém estatísticas sem especificar uma unidade, o CloudWatch agrega todos os pontos de dados da mesma unidade. Se você tiver duas métricas idênticas com unidades diferentes, dois streams de dados separados serão retornados, um para cada unidade.

Períodos

Período é o intervalo de tempo associado a uma determinada estatística do Amazon CloudWatch. Cada estatística representa uma agregação de dados de métricas coletados por um período especificado. Os períodos são definidos em número de segundos. Os valores válidos para o período são 1, 5, 10, 30 ou qualquer múltiplo de 60. Por exemplo, para especificar um período de seis minutos, use 360 como o valor do período. Você pode ajustar a forma como os dados são agregados alterando a duração do período. O valor padrão de um período é de 60 segundos. Um período pode ser tão curto quanto um segundo e deve ser um múltiplo de 60 se for maior que o valor padrão de 60 segundos.

Somente métricas personalizadas que você define com uma solução de armazenamento de 1 segundo oferece suporte aos períodos inferiores a um minuto. Embora a opção de definir um período

abaixo de 60 esteja sempre disponível no console, você deve selecionar um período que alinha a forma como a métrica é armazenada. Para obter mais informações sobre as métricas que oferecem suporte a períodos com menos de um minuto, consulte Métricas de alta resolução.

Ao recuperar estatísticas, você pode especificar um período, os horários de início e de término. Esses parâmetros determinam o período geral associado às estatísticas. Os valores padrão para os horários de início e de término obtêm as estatísticas da última hora. Os valores que você especifica para os horários de início e de término determinam quantos períodos o CloudWatch retornará. Por exemplo, ao recuperar as estatísticas usando os valores padrão para o período, os horário de início e de término é retornado um conjunto agregado de estatísticas para cada minuto da hora anterior. Se você preferir estatísticas agregadas em blocos de 10 minutos, especifique um período de 600. Para estatísticas agregadas durante toda a hora, especifique um período de 3600.

Quando as estatísticas são agregadas ao longo de um período temporal, elas são marcadas com a hora correspondente ao início do período. Por exemplo, os dados agregados das 19h para às 20h são marcados como 19h. Além disso, os dados agregados entre 19h e 20h começam a ficar visíveis às 19h e, em seguida, os valores desses dados agregados podem alterar conforme o CloudWatch recolhe mais amostras durante o período.

Períodos também são importantes para os alarmes do CloudWatch. Ao criar um alarme para monitorar uma métrica específica, você solicita que o CloudWatch compare essa métrica ao valor de limite especificado. Você tem extenso controle sobre a forma como o CloudWatch faz essa comparação. Você pode especificar o período no qual a comparação é feita, mas você também pode especificar quantos períodos de avaliação são usados para chegar a uma conclusão. Por exemplo, se você especificar três períodos de avaliação, o CloudWatch comparará uma janela de três pontos de dados. O CloudWatch só notificará você se o ponto de dados mais antigo e os outros estiverem excedendo o limite ou estiverem ausentes.

Agregação

O Amazon CloudWatch agrega estatísticas de acordo com o período que você especifica ao recuperar estatísticas. Você pode publicar quantos pontos de dados quiser com carimbos de data/hora equivalentes ou semelhantes. O CloudWatch os agrega de acordo com o período especificado. O CloudWatch não agrega dados automaticamente entre regiões, mas é possível usar a matemática de métricas para agregar métricas de diferentes regiões.

Você pode publicar pontos de dados para uma métrica que compartilham não apenas o mesmo carimbo de data/hora, mas também o mesmo namespace e dimensões. O CloudWatch retorna

estatísticas agregadas para esses pontos de dados. Você também pode publicar vários pontos de dados para as mesmas métricas ou métricas diferentes, com qualquer time stamp.

Para conjuntos de dados grandes, você pode inserir um conjunto de dados pré-agregados chamado conjunto de estatísticas. Com conjuntos de estatísticas, você atribui ao CloudWatch os valores Min, Max, Sum e SampleCount para vários pontos de dados. Isso é usado com frequência quando você precisa coletar dados muitas vezes em um minuto. Por exemplo, suponha que você tenha uma métrica para a latência de solicitação de uma página da web. Não faz sentido publicar dados com cada acesso à página da web. Sugerimos que você colete a latência de todos os acessos a essa página da Web, faça a agregação uma vez por minuto e envie esse conjunto de estatísticas para o CloudWatch.

O Amazon CloudWatch não diferencia a origem de uma métrica. Se você publicar uma métrica com o mesmo namespace e dimensões de origens diferentes, o CloudWatch a tratará como uma única métrica. Isso pode ser útil para as métricas de serviço em um sistema escalado e distribuído. Por exemplo, todos os hosts em uma aplicação de servidor da Web podem publicar métricas idênticas que representem a latência das solicitações em processamento. O CloudWatch as trata como uma única métrica, permitindo que você obtenha as estatísticas para os valores mínimo, máximo, médio e soma de todas as solicitações em sua aplicação.

Percentis

Um percentil indica a posição relativa de um valor no conjunto de dados. Por exemplo, o 95º percentil significa que 95% dos dados são inferiores a esse valor e 5% são superiores a esse valor. Percentis ajudam você a ter uma melhor compreensão da distribuição de seus dados de métrica.

Percentis geralmente são usados para isolar anomalias. Em uma distribuição normal, 95% dos dados ficam dentro de dois desvios padrão da média e 99,7% ficam dentro de três desvios padrão da média. Todos os dados que ficam fora dos três desvios padrão normalmente são considerados uma anomalia porque diferem de muito do valor médio. Por exemplo, suponha que você esteja monitorando a utilização da CPU de suas instâncias EC2 para garantir que seus clientes tenham uma boa experiência. Se você monitorar a média, isso poderá ocultar anomalias. Se você monitorar o máximo, uma única anomalia poderá se desviar dos resultados. Usando percentis, é possível monitorar o 95º percentil de utilização da CPU para verificar se há instâncias com uma carga pesada incomum.

Algumas métricas do CloudWatch oferecem suporte a percentis como estatística. Para essas métricas, você pode monitorar seu sistema e suas aplicações usando percentis da mesma forma que

usaria as outras estatísticas do CloudWatch (Média, Mínimo, Máximo e Soma). Por exemplo, ao criar um alarme, você pode usar percentis como a função estatística. É possível especificar o percentil com até dez casas decimais (por exemplo, p95.0123456789).

Estatísticas de percentil estão disponíveis para métricas personalizadas, contanto que você publique os pontos de dados brutos e não resumidos para a métrica personalizada. As estatísticas de percentil não estão disponíveis para métricas quando qualquer um dos valores de métrica são números negativos.

O CloudWatch precisa dos pontos e dados brutos para calcular percentis. Se publicar dados usando um conjunto de estatísticas, você só poderá recuperar estatísticas de percentis para esses dados se uma das seguintes condições for verdadeira:

- O valor SampleCount do conjunto de estatísticas é 1 e Min, Max e Sum são todos iguais.
- Min e Max são iguais, e Sum é igual a Min multiplicado por SampleCount.

Os seguintes produtos da AWS incluem métricas compatíveis com estatísticas percentis.

- API Gateway
- Application Load Balancer
- Amazon EC2
- Elastic Load Balancing
- Kinesis
- Amazon RDS

O CloudWatch também é compatível com médias aparadas e outras estatísticas de performance, que podem ter um uso semelhante aos percentis. Para ter mais informações, consulte Definições de estatísticas do CloudWatch.

Alarmes

É possível usar um alarme para iniciar automaticamente ações em seu nome. Um alarme observa uma única métrica ao longo de um período especificado e realiza uma ou mais ações especificadas com base no valor da métrica em relação a um limite especificado ao longo do tempo. A ação é uma notificação enviada a um tópico do Amazon SNS ou a uma política de Auto Scaling. Você também pode adicionar alarmes aos painéis.

Os alertas invocam ações apenas para alterações de estado mantidas. Os alarmes do CloudWatch não invocam ações só porque estão em um determinado estado. O estado deve ter sido alterado e mantido por uma quantidade especificada de períodos.

Ao criar um alarme, selecione um período de monitoramento de alarme maior ou igual à resolução da métrica. Por exemplo, o monitoramento básico para o Amazon EC2 fornece métricas para suas instâncias a cada cinco minutos. Ao definir um alarme em uma métrica de monitoramento básico, selecione um período de, pelo menos, 300 segundos (5 minutos). O monitoramento detalhado para o Amazon EC2 fornece métricas para suas instâncias com uma resolução de 1 minuto. Ao definir um alarme em uma métrica de monitoramento detalhado, selecione um período de, pelo menos, 60 segundos (1 minuto).

Se você definir um alarme em uma métrica de alta resolução, pode especificar um alarme de alta resolução com um período de 10 ou 30 segundos ou pode definir um alarme regular com um período de qualquer múltiplo de 60 segundos. Há um custo maior para alarmes de alta resolução. Para obter mais informações sobre as métricas de alta resolução, consulte Publicar métricas personalizadas do .

Para obter mais informações, consulte Usar alarmes do Amazon CloudWatch e Criar um alarme a partir de uma métrica em um gráfico.

Faturamento e custos

Para obter mais informações sobre a definição de preços do CloudWatch, consulte Definição de preço do Amazon CloudWatch.

Para obter informações que podem ajudar a analisar sua fatura e, possivelmente, a otimizar e reduzir custos, consulte Faturamento e custos do CloudWatch.

Recursos do Amazon CloudWatch

Os recursos relacionados a seguir podem ajudar você à medida que trabalha com este serviço.

Recurso	Descrição
Perguntas frequentes sobre o Amazon CloudWatch	As perguntas frequentes abordam as dúvidas mais comuns entre os desenvolvedores deste produto.

Recurso	Descrição
Centro do desenvolvedor da AWS	Um ponto de partida central para localizar documentação, exemplos de código, notas de release e outras informaçõ es para ajudar você a desenvolver aplicativos inovadores com a AWS.
AWS Management Console	O console permite realizar a maioria das funções do Amazon CloudWatch e de várias outras ofertas da AWS sem programação.
Fóruns de discussão do Amazon CloudWatch	Fórum da comunidade para os desenvolvedores discutire m questões técnicas relacionadas ao Amazon CloudWatc h.
AWS Support	O centro para criar e gerenciar os seus casos do AWS Support. Também inclui links para outros recursos úteis, como fóruns, perguntas frequentes, status de integridade do serviço e AWS Trusted Advisor.
Informações do produto Amazon CloudWatch	A primeira página da Web para obter informações sobre o Amazon CloudWatch.
Entre em contato conosco	Um ponto central de contato para consultas relativas a faturamento da AWS, conta, eventos, abuso etc.

Começar a usar

Para usar o Amazon CloudWatch, você precisa de uma conta da AWS. Sua conta da AWS permite o uso de serviços (por exemplo, o Amazon EC2) para gerar métricas que você pode visualizar no console do CloudWatch, uma interface de clique baseada na Web. Além disso, você pode instalar e configurar a interface de linha de comando (CLI) da AWS.

Cadastre-se em uma Conta da AWS

Se você ainda não tem Conta da AWS, siga as etapas a seguir para criar uma.

Para se cadastrar em uma Conta da AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções on-line.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

   Quando você se cadastra em uma Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como uma prática recomendada de segurança, atribua o acesso administrativo para um usuário e use somente o usuário-raiz para executar tarefas que requerem o acesso de usuário-raiz.

A AWS envia um e-mail de confirmação depois que o processo de cadastramento é concluído. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando https://aws.amazon.com/ e selecionando Minha conta.

Criar um usuário com acesso administrativo

Depois de se cadastrar em uma Conta da AWS, proteja seu Usuário raiz da conta da AWS, habilite o AWS IAM Identity Center e crie um usuário administrativo para não usar o usuário raiz em tarefas cotidianas.

Proteger seu Usuário raiz da conta da AWS

1. Faça login no AWS Management Console como o proprietário da conta ao escolher a opção Usuário raiz e inserir o endereço de e-mail da Conta da AWS. Na próxima página, digite sua senha.

   Para obter ajuda ao fazer login usando o usuário raiz, consulte Fazer login como usuário raiz no Guia do usuário do Início de Sessão da AWS.

2. Ative a autenticação multifator (MFA) para o usuário raiz.c

   Para obter instruções, consulte Habilitar um dispositivo MFA virtual para o usuário raiz de sua conta da Conta da AWS para seu (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.

   Para obter instruções, consulte Habilitar AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center.

2. No Centro de Identidade do IAM, conceda o acesso administrativo para um usuário.

   Para obter um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso dos usuários com o Diretório do Centro de Identidade do IAM padrão no Guia do usuário do AWS IAM Identity Center.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com seu usuário do Centro de Identidade do IAM, use a URL de login que foi enviada ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

  Para obter ajuda com o login utilizando um usuário do Centro de Identidade do IAM, consulte Fazer login no portal de acesso da AWS, no Guia do usuário do Início de Sessão da AWS.

Atribuir acesso para usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte Create a permission set no Guia do usuário do AWS IAM Identity Center.

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Add groups no Guia do usuário do AWS IAM Identity Center.

Faça login no console do Amazon CloudWatch

Para fazer login no console do Amazon CloudWatch

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Se necessário, use a barra de navegação para mudar para a região onde estão seus recursos da AWS.
3. Mesmo se esta for a primeira vez que você usa o console do CloudWatch, o Your Metrics (Suas métricas) já pode informar métricas porque você usou um produto da AWS que envia automaticamente as métricas ao Amazon CloudWatch gratuitamente. Outros produtos necessitam que você habilite métricas.

   Se você não tiver nenhum alarme, a seção Seus alarmes terá um botão Criar alarme.

Configurar o AWS CLI

É possível usar a AWS CLI ou a CLI do Amazon CloudWatch para executar comandos do CloudWatch. A AWS CLI substitui a CLI do CloudWatch. Os novos recursos do CloudWatch são incluídos apenas na AWS CLI.

Para obter informações sobre como instalar e configurar a AWS CLI, consulte Como configurar com a AWS Command Line Interface no Manual do usuário da AWS Command Line Interface.

Para obter informações sobre como instalar e configurar a CLI do Amazon CloudWatch, consulte Configurar a interface de linha de comando na Referência da CLI do Amazon CloudWatch.

Conceitos básicos do Amazon CloudWatch

Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

A página inicial de visão geral do CloudWatch é exibida.

A visão geral exibe os itens a seguir, atualizados automaticamente.

- Alarmes por serviço da AWS exibe uma lista dos serviços da AWS que você usa em sua conta, junto com o estado dos alarmes nesses serviços. Ao lado, dois ou quatro alarmes em sua conta são exibidos. O número de alarmes dependerá de quantos serviços da AWS você usa. Os alarmes mostrados são aqueles no estado ALARM ou que mudaram de estado mais recentemente.

  Essas áreas superiores ajudam você a avaliar rapidamente a integridade dos serviços da AWS, visualizando os estados de alarme em todos os serviços e os alarmes que mudaram de estado mais recentemente. Isso ajuda a monitorar e fazer o diagnóstico de problemas rapidamente.
- Abaixo dessas áreas está o painel padrão, se houver. O painel padrão é um painel personalizado que você criou e denominou como CloudWatch-Default. Essa é uma maneira prática de você adicionar métricas sobre os próprios serviços ou aplicativos personalizados à página de visão geral ou de trazer métricas-chave adicionais dos serviços da AWS que você deseja monitorar.

> **Note**
>
> Os painéis automáticos na página inicial do CloudWatch exibem apenas informações da conta atual, mesmo que a conta seja uma conta de monitoramento configurada para a observabilidade entre contas do CloudWatch. Para obter informações sobre como criar painéis personalizados entre contas, consulte Painel da observabilidade entre contas do CloudWatch.

Nessa visão geral, você pode ver um painel de métricas entre serviços de vários serviços da AWS ou focar sua visão em um grupo de recursos específico ou em um serviço específico da AWS. Isso permite restringir a exibição a um subconjunto de recursos no qual você tem interesse. Para obter mais informações, consulte as seções a seguir.

Visualizar o painel automático predefinido para um único serviço

Para visualizar o painel automático predefinido para um único serviço

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

   A página inicial é exibida.
2. No painel de navegação à esquerda, escolha Painéis.
3. Escolha a guia Painéis automáticos e escolha o serviço que deseja visualizar.
4. Para alternar para a visualização dos alarmes desse serviço, marque a caixa de seleção Em alarme, Dados insuficientes ou OK na parte superior da tela onde o nome do serviço é exibido atualmente.
5. Ao exibir métricas, concentre-se em uma determinada métrica de várias maneiras:

   a. Para ver mais detalhes sobre as métricas em qualquer gráfico, passe o mouse sobre o gráfico e escolha o ícone de ações View in metrics (Exibir em métricas).

      O gráfico é exibido em uma nova guia, com as métricas relevantes listadas abaixo do gráfico. Personalize a exibição desse gráfico, alterando as métricas e os recursos mostrados, a estatística, o período e outros fatores para obter uma compreensão melhor da situação atual.

b. Exiba os eventos de log no intervalo de tempo mostrado no gráfico. Isso pode ajudar a descobrir eventos acontecidos na infraestrutura que estão causando uma alteração inesperada nas métricas.

   Para consultar os eventos de log, passe o mouse sobre o gráfico e escolha o ícone de ações View in logs (Exibir em logs).

   A exibição do CloudWatch Logs é visualizada em uma nova guia com uma lista dos grupos de logs. Para consultar os eventos de log em um desses grupos de logs ocorridos durante o período mostrado no gráfico original, escolha esse grupo de logs.

6. Ao exibir alarmes, concentre-se em um determinado alarme de várias maneiras:

   - Para consultar mais detalhes sobre um alarme, passe o mouse sobre o alarme e escolha o ícone View in alarms (Exibir em alarmes).

     A exibição de alarmes é mostrada em uma nova guia, exibindo uma lista dos alarmes, além de detalhes sobre o alarme escolhido. Para consultar o histórico desse alarme, escolha a guia History (Histórico).

7. Os alarmes são sempre atualizados uma vez por minuto. Para atualizar a exibição, escolha o ícone de atualização (duas setas curvas) na parte superior direita da tela. Para alterar a taxa de renovação automática de itens na tela que não sejam alarmes, escolha a seta para baixo ao lado do ícone de atualização e escolha uma taxa de atualização. Também opte por desativar a atualização automática.
8. Para alterar o intervalo de tempo mostrado em todos os gráficos e alarmes exibidos atualmente ao lado de Time range (Intervalo de tempo) na parte superior da tela, escolha o intervalo. Para selecionar mais opções de intervalo de tempo do que as exibidas por padrão, escolha custom (personalizada).
9. Para retornar ao painel entre serviços, escolha Overview (Visão geral) na lista na parte superior da tela que atualmente mostra o serviço no qual você está se concentrando.

   Como alternativa, em qualquer exibição, escolha CloudWatch na parte superior da tela para limpar todos os filtros e retornar à página de visão geral.

Visualizar o painel predefinido entre serviços

Você pode alternar para a tela do painel de controle entre serviços e interagir com painéis para todos os serviços da AWS que você estiver usando. O console do CloudWatch exibe os painéis em ordem alfabética e mostra uma ou duas métricas principais em cada painel.

> **Note**
>
> Se você estiver usando cinco ou mais serviços da AWS, o console do CloudWatch não exibirá o painel de controle entre serviços na tela Overview (Visão geral).

Para abrir o painel entre serviços

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

   Você será direcionado para a tela Overview (Visão geral).
2. Na tela Overview (Visão geral), selecione o menu suspenso Overview (Visão geral) e então escolha Cross service dashboard (Painel entre serviços).

   Você será direcionado para a tela do painel entre serviços.
3. (Opcional) Se você estiver usando a interface original, role até a seção Cross-service dashboard (Painel entre serviços) e depois escolha View Cross-service dashboard (Exibir painel de controle entre serviços).

   Você será direcionado para a tela do painel de controle entre serviços.
4. Concentre-se em um determinado serviço de duas maneiras:

   a. Para consultar mais métricas-chave de um serviço, escolha o nome na lista na parte superior da tela, onde Cross service dashboard (Painel entre serviços) é mostrado no momento. Ou escolha View Service dashboard (Exibir painel de serviço) ao lado do nome do serviço.

      Um painel automático desse serviço é exibido, mostrando mais métricas desse serviço. Além disso, para alguns serviços, a parte inferior do painel de serviço exibe recursos relacionados a esse serviço. Escolha um desses recursos para esse console de serviço e se concentre mais nesse recurso.

b. Para ver todos os alarmes relacionados a um serviço, escolha o botão no lado direito da tela ao lado do nome desse serviço. O texto nesse botão indica quantos alarmes você criou nesse serviço e se algum está no estado ALARM.

   Quando os alarmes são exibidos, vários alarmes com configurações semelhantes (como dimensões, limite ou período) podem ser mostrados em um único gráfico.

   Assim, é possível exibir detalhes sobre um alarme e consultar o histórico de alarmes. Para isso, passe o mouse sobre o gráfico do alarme e escolha o ícone de ações View in alarms (Exibir em alarmes).

   A exibição de alarmes é mostrada em uma nova guia do navegador, exibindo uma lista dos alarmes, além de detalhes sobre o alarme escolhido. Para consultar o histórico desse alarme, escolha a guia History (Histórico).

5. Concentre-se em recursos em um determinado grupo de recursos. Para isso, escolha o grupo de recursos na lista na parte superior da página onde All resources (Todos os recursos) é exibido.

   Para obter mais informações, consulte Visualizar um painel predefinido para um grupo de recursos.

6. Para alterar o intervalo de tempo mostrado em todos os gráficos e alarmes exibidos atualmente, selecione o intervalo desejado ao lado de Time range (Intervalo de tempo) na parte superior da tela. Escolha custom (personalizada) para selecionar mais opções de intervalo de tempo do que as exibidas por padrão.
7. Os alarmes são sempre atualizados uma vez por minuto. Para atualizar a exibição, escolha o ícone de atualização (duas setas curvas) na parte superior direita da tela. Para alterar a taxa de renovação automática de itens na tela que não sejam alarmes, escolha a seta para baixo ao lado do ícone de atualização e escolha a taxa de atualização desejada. Também opte por desativar a atualização automática.

Remover um serviço do painel entre serviços

É possível evitar que as métricas de um serviço sejam exibidas no painel entre serviços. Isso ajuda a concentrar o painel nos serviços que você mais deseja monitorar.

Se você remover um serviço do painel entre serviços, os alarmes desse serviço continuarão sendo exibidos nas visualizações dos alarmes.

Para remover as métricas de um serviço do painel entre serviços

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

   A página inicial é exibida.
2. Na parte superior da página, em Overview (Visão geral), escolha o serviço que você deseja remover.

   A exibição muda para mostrar métricas apenas desse serviço.
3. Escolha Actions (Ações) e desmarque a caixa de seleção ao lado de Show on cross service dashboard (Mostrar no painel entre serviços).

Visualizar um painel predefinido para um grupo de recursos

Concentre a exibição para mostrar métricas e alarmes em um único grupo de recursos. Usar grupos de recursos permite utilizar tags para organizar projetos, concentrar-se em um subconjunto da arquitetura ou diferenciar os ambientes de produção e desenvolvimento. Eles também permitem se concentrar em cada um desses grupos de recursos na visão geral do CloudWatch. Para obter mais informações, consulte O que é o AWS Resource Groups?

Quando você se concentra em um grupo de recursos, a tela muda para só mostrar os serviços nos quais você tem recursos marcados como parte desse grupo de recursos. A área de alarmes recentes exibe apenas alarmes associados a recursos que fazem parte do grupo de recursos. Além disso, se você tiver criado um painel com o nome CloudWatch-Default-ResourceGroupName, ele será exibido na área Default dashboard (Painel padrão).

Detalhe ainda mais se concentrando em um único serviço da AWS e em um grupo de recursos ao mesmo tempo. O procedimento a seguir explica apenas como manter o foco em um grupo de recursos.

Para se concentrar em um único grupo de recursos

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Na parte superior da página, onde All resources (Todos os recursos) é exibido, escolha um grupo de recursos.
3. Para consultar mais métricas relacionadas a esse grupo de recursos, ao lado da parte inferior da tela, escolha View cross service dashboard (Exibir painel entre serviços).

O painel entre serviços é exibido, mostrando apenas os serviços relacionados a esse grupo de recursos. Para cada serviço, uma ou duas métricas-chave são exibidas.

4. Para alterar o intervalo de tempo mostrado em todos os gráficos e alarmes exibidos atualmente, em Time range (Intervalo de tempo) na parte superior da tela, selecione um intervalo. Para selecionar mais opções de intervalo de tempo do que as exibidas por padrão, escolha custom (personalizada).
5. Os alarmes são sempre atualizados uma vez por minuto. Para atualizar a exibição, escolha o ícone de atualização (duas setas curvas) na parte superior direita da tela. Para alterar a taxa de renovação automática de itens na tela que não sejam alarmes, escolha a seta para baixo ao lado do ícone de atualização e escolha uma taxa de atualização. Também opte por desativar a atualização automática.
6. Para retornar à exibição de informações sobre todos os recursos na conta, perto da parte superior da tela onde o nome do grupo de recursos é exibido no momento, escolha All resources (Todos os recursos).

Visualizar o painel predefinido entre serviços

Você pode alternar para a tela do painel de controle entre serviços e interagir com painéis para todos os serviços da AWS que você estiver usando. O console do CloudWatch exibe seus painéis em ordem alfabética e apresenta uma ou duas métricas principais de cada serviço.

> **Note**
>
> Se você estiver usando cinco ou mais serviços da AWS, o console do CloudWatch não exibirá o painel de controle entre serviços na tela Overview (Visão geral).

Para abrir o painel entre serviços

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

   Você será direcionado para a tela Overview (Visão geral).
2. Na tela Overview (Visão geral), selecione o menu suspenso Overview (Visão geral) e então escolha Cross service dashboard (Painel entre serviços).

   Você será direcionado para a tela do painel entre serviços.

3. (Opcional) Se você estiver usando a interface original, role até a seção Cross-service dashboard (Painel entre serviços) e depois escolha View Cross-service dashboard (Exibir painel de controle entre serviços).

   Você será direcionado para a tela do painel de controle entre serviços.

4. Concentre-se em um determinado serviço de duas maneiras:

   a. Para consultar mais métricas-chave de um serviço, escolha o nome na lista na parte superior da tela, onde Cross service dashboard (Painel entre serviços) é mostrado no momento. Ou escolha View Service dashboard (Exibir painel de serviço) ao lado do nome do serviço.

      Um painel automático desse serviço é exibido, mostrando mais métricas desse serviço. Além disso, para alguns serviços, a parte inferior do painel de serviço exibe recursos relacionados a esse serviço. Escolha um desses recursos para esse console de serviço e se concentre mais nesse recurso.

   b. Para ver todos os alarmes relacionados a um serviço, escolha o botão no lado direito da tela ao lado do nome desse serviço. O texto nesse botão indica quantos alarmes você criou nesse serviço e se algum está no estado ALARM.

      Quando os alarmes são exibidos, vários alarmes com configurações semelhantes (como dimensões, limite ou período) podem ser mostrados em um único gráfico.

      Assim, é possível exibir detalhes sobre um alarme e consultar o histórico de alarmes. Para isso, passe o mouse sobre o gráfico do alarme e escolha o ícone de ações View in alarms (Exibir em alarmes).

      A exibição de alarmes é mostrada em uma nova guia do navegador, exibindo uma lista dos alarmes, além de detalhes sobre o alarme escolhido. Para consultar o histórico desse alarme, escolha a guia History (Histórico).

5. Concentre-se em recursos em um determinado grupo de recursos. Para isso, escolha o grupo de recursos na lista na parte superior da página onde All resources (Todos os recursos) é exibido.

   Para obter mais informações, consulte Visualizar um painel predefinido para um grupo de recursos.

6. Para alterar o intervalo de tempo mostrado em todos os gráficos e alarmes exibidos atualmente, selecione o intervalo desejado ao lado de Time range (Intervalo de tempo) na parte superior da

tela. Escolha custom (personalizada) para selecionar mais opções de intervalo de tempo do que as exibidas por padrão.

7. Os alarmes são sempre atualizados uma vez por minuto. Para atualizar a exibição, escolha o ícone de atualização (duas setas curvas) na parte superior direita da tela. Para alterar a taxa de renovação automática de itens na tela que não sejam alarmes, escolha a seta para baixo ao lado do ícone de atualização e escolha a taxa de atualização desejada. Também opte por desativar a atualização automática.

Remover um serviço da exibição no painel entre serviços

É possível evitar que as métricas de um serviço sejam exibidas no painel entre serviços. Isso ajuda a concentrar o painel nos serviços que você mais deseja monitorar.

Se você remover um serviço do painel entre serviços, os alarmes desse serviço continuarão sendo exibidos nas visualizações dos alarmes.

Para remover as métricas de um serviço do painel entre serviços

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

   A página inicial é exibida.

2. Na parte superior da página, em Overview (Visão geral), escolha o serviço que você deseja remover.

   A exibição muda para mostrar métricas apenas desse serviço.

3. Escolha Actions (Ações) e desmarque a caixa de seleção ao lado de Show on cross service dashboard (Mostrar no painel entre serviços).

Visualizar um painel predefinido para um único serviço da AWS

Na página inicial do CloudWatch, concentre a exibição em um único produto da AWS. Detalhe ainda mais se concentrando em um único serviço da AWS e em um grupo de recursos ao mesmo tempo. O procedimento a seguir só mostra como se concentrar em um serviço da AWS.

Para se concentrar em um único serviço

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

A página inicial é exibida.

2. Em Visão geral, onde Visão geral é exibida atualmente na lista suspensa, escolha Painéis de serviços.
3. Escolha o serviço no qual deseja se concentrar.

   A exibição muda para exibir gráficos de métricas-chave do serviço selecionado.
4. Para alternar para a visualização dos alarmes desse serviço, marque a caixa de seleção Em alarme, Dados insuficientes ou OK na parte superior da tela onde o nome do serviço é exibido atualmente.
5. Ao exibir métricas, concentre-se em uma determinada métrica de várias maneiras:

   a. Para ver mais detalhes sobre as métricas em qualquer gráfico, passe o mouse sobre o gráfico e escolha o ícone de ações View in metrics (Exibir em métricas).

      O gráfico é exibido em uma nova guia, com as métricas relevantes listadas abaixo do gráfico. Personalize a exibição desse gráfico, alterando as métricas e os recursos mostrados, a estatística, o período e outros fatores para obter uma compreensão melhor da situação atual.

   b. Exiba os eventos de log no intervalo de tempo mostrado no gráfico. Isso pode ajudar a descobrir eventos acontecidos na infraestrutura que estão causando uma alteração inesperada nas métricas.

      Para consultar os eventos de log, passe o mouse sobre o gráfico e escolha o ícone de ações View in logs (Exibir em logs).

      A exibição do CloudWatch Logs é visualizada em uma nova guia com uma lista dos grupos de logs. Para consultar os eventos de log em um desses grupos de logs ocorridos durante o período mostrado no gráfico original, escolha esse grupo de logs.
6. Ao exibir alarmes, concentre-se em um determinado alarme de várias maneiras:

   - Para consultar mais detalhes sobre um alarme, passe o mouse sobre o alarme e escolha o ícone View in alarms (Exibir em alarmes).

     A exibição de alarmes é mostrada em uma nova guia, exibindo uma lista dos alarmes, além de detalhes sobre o alarme escolhido. Para consultar o histórico desse alarme, escolha a guia History (Histórico).

7. Os alarmes são sempre atualizados uma vez por minuto. Para atualizar a exibição, escolha o ícone de atualização (duas setas curvas) na parte superior direita da tela. Para alterar a taxa de renovação automática de itens na tela que não sejam alarmes, escolha a seta para baixo ao lado do ícone de atualização e escolha uma taxa de atualização. Também opte por desativar a atualização automática.
8. Para alterar o intervalo de tempo mostrado em todos os gráficos e alarmes exibidos atualmente ao lado de Time range (Intervalo de tempo) na parte superior da tela, escolha o intervalo. Para selecionar mais opções de intervalo de tempo do que as exibidas por padrão, escolha custom (personalizada).
9. Para retornar ao painel entre serviços, escolha Overview (Visão geral) na lista na parte superior da tela que atualmente mostra o serviço no qual você está se concentrando.

   Como alternativa, em qualquer exibição, escolha CloudWatch na parte superior da tela para limpar todos os filtros e retornar à página de visão geral.

Visualizar um painel predefinido para um grupo de recursos

Concentre a exibição para mostrar métricas e alarmes em um único grupo de recursos. Usar grupos de recursos permite utilizar tags para organizar projetos, concentrar-se em um subconjunto da arquitetura ou diferenciar os ambientes de produção e desenvolvimento. Eles também permitem se concentrar em cada um desses grupos de recursos na visão geral do CloudWatch. Para obter mais informações, consulte O que é o AWS Resource Groups?

Quando você se concentra em um grupo de recursos, a tela muda para só mostrar os serviços nos quais você tem recursos marcados como parte desse grupo de recursos. A área de alarmes recentes exibe apenas alarmes associados a recursos que fazem parte do grupo de recursos. Além disso, se você tiver criado um painel com o nome CloudWatch-Default-ResourceGroupName, ele será exibido na área Default dashboard (Painel padrão).

Detalhe ainda mais se concentrando em um único serviço da AWS e em um grupo de recursos ao mesmo tempo. O procedimento a seguir só mostra como se concentrar em um grupo de recursos.

Para se concentrar em um único grupo de recursos

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Na parte superior da página, onde All resources (Todos os recursos) é exibido, escolha um grupo de recursos.

3. Para consultar mais métricas relacionadas a esse grupo de recursos, ao lado da parte inferior da tela, escolha View cross service dashboard (Exibir painel entre serviços).

   O painel entre serviços é exibido, mostrando apenas os serviços relacionados a esse grupo de recursos. Para cada serviço, uma ou duas métricas-chave são exibidas.
4. Para alterar o intervalo de tempo mostrado em todos os gráficos e alarmes exibidos atualmente, em Time range (Intervalo de tempo) na parte superior da tela, selecione um intervalo. Para selecionar mais opções de intervalo de tempo do que as exibidas por padrão, escolha custom (personalizada).
5. Os alarmes são sempre atualizados uma vez por minuto. Para atualizar a exibição, escolha o ícone de atualização (duas setas curvas) na parte superior direita da tela. Para alterar a taxa de renovação automática de itens na tela que não sejam alarmes, escolha a seta para baixo ao lado do ícone de atualização e escolha uma taxa de atualização. Também opte por desativar a atualização automática.
6. Para retornar à exibição de informações sobre todos os recursos na conta, perto da parte superior da tela onde o nome do grupo de recursos é exibido no momento, escolha All resources (Todos os recursos).

Faturamento e custos do CloudWatch

Esta seção descreve como os recursos do Amazon CloudWatch geram custos e fornece métodos que podem ajudar você a analisar, otimizar e reduzir os custos do CloudWatch. Ao longo desta seção, às vezes usamos preços ao descrever os recursos do CloudWatch. Para obter informações sobre preços, consulte Preço do Amazon CloudWatch.

Tópicos

Analisar os dados de custo e uso do CloudWatch com o Explorador de Custos

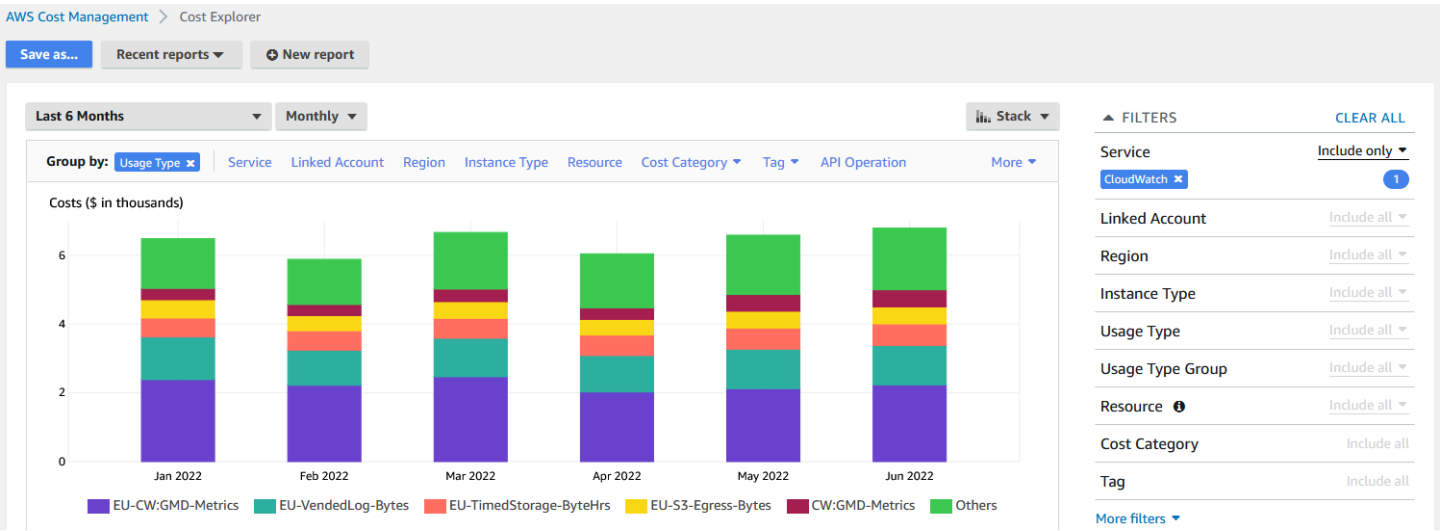
Com o AWS Cost Explorer, você pode visualizar e analisar dados de custo e uso de Serviços da AWS ao longo do tempo, incluindo o CloudWatch. Para obter mais informações, consulte Conceitos básicos do AWS Cost Explorer.

O procedimento a seguir descreve como usar o Explorador de Custos para visualizar e analisar dados de custo e uso do CloudWatch.

Para visualizar e analisar dados de uso e custo do CloudWatch

1. Faça login no console do Explorador de Custos em https://console.aws.amazon.com/cost-management/home#/custom.
2. Em FILTERS (FILTROS), selecione CloudWatch para Service (Serviço).
3. Em Group by (Agrupar por), escolha Usage Type (Tipo de uso). Você também pode agrupar os resultados por outras categorias, como:
   - API Operation (Operação de API): veja quais operações de API geraram mais custos.
   - Region (Região): veja quais regiões geraram mais custos.

A imagem a seguir mostra um exemplo dos custos gerados pelos recursos do CloudWatch ao longo de seis meses.



Para ver quais recursos do CloudWatch geraram mais custos, analise os valores de `UsageType`. Por exemplo, `EU-CW:GMD-Metrics` representa os custos gerados pelas solicitações de API em massa do CloudWatch.

> **Note**
>
> As strings de `UsageType` correspondem a recursos e regiões específicos. Por exemplo, a primeira parte de `EU-CW:GMD-Metrics` (`EU`) corresponde à região da Europa (Irlanda) e a segunda parte de `EU-CW:GMD-Metrics` (`GMD-Metrics`) corresponde às solicitações de API em massa do CloudWatch.
>
> A string inteira de `UsageType` pode ser formatada como `<Region>-CW:<Feature>` ou `<Region>-<Feature>`.
>
> Para melhorar a legibilidade, as strings de `UsageType` em todas tabelas deste documento foram encurtadas para seus sufixos. Por exemplo, `EU-CW:GMD-Metrics` foi encurtada para `GMD-Metrics`.

A tabela a seguir inclui os nomes de cada recurso e sub-recurso do CloudWatch e lista as strings de `UsageType`.

Recurso do CloudWatch	Sub-recurso do CloudWatch	**`UsageType`**
Métricas do CloudWatch	Métricas personalizadas	`MetricMonitorUsage`

Recurso do CloudWatch	Sub-recurso do CloudWatch	**`UsageType`**
	Monitoramento detalhado	`MetricMonitorUsage`
	Métricas incorporadas	`MetricMonitorUsage`
Solicitações de API do CloudWatch	Solicitações de API	`Requests`
	Em massa (Get)	`GMD-Metrics`
	Contributor Insights	`GIRR-Metrics`
	Snapshot de imagem bitmap	`GMWI-Metrics`
Fluxos de métricas do CloudWatch	Fluxos de métricas	`MetricStreamUsage`
Painéis do CloudWatch	Painel com 50 métricas ou menos	`DashboardsUsageHour-Basic`
	Painel com mais de 50 métricas	`DashboardsUsageHour`
Alarmes do CloudWatch	Standard (metric alarm) (Padrão [métricas de alarme])	`AlarmMonitorUsage`
	High resolution (metric alarm) (Atla resolução [métricas de alarme])	`HighResAlarmMonitorUsage`
	Metrics Insights query alarm (Alarme de consulta ao Metrics Insights)	`MetricInsightAlarmUsage`

Recurso do CloudWatch	Sub-recurso do CloudWatch	**`UsageType`**
	Composite (aggregated alarm) (Composto [alarme agregado])	`CompositeAlarmMonitorUsage`
CloudWatch Application Signals	Application Signals	`Application-Signals`
Logs personalizados do CloudWatch	Coleta (ingestão)	`DataProcessing-Bytes`
	Armazenamento (arquivo)	`TimedStorage-ByteHrs`
	Análise (consulta)	`DataScanned-Bytes`
Logs de acesso infrequente do CloudWatch	Coleta (ingestão)	`DataProcessingIA-Bytes`
Logs fornecidos do CloudWatch	Entrega (Amazon CloudWatch Logs)	`VendedLog-Bytes`
	Entrega (Logs de acesso infrequente do CloudWatch Logs)	`VendedLogIA-Bytes`
	Entrega (Amazon Simple Storage Service)	`S3-Egress-ComprBytes` `S3-Egress-Bytes`
	Entrega (Amazon Data Firehose)	`FH-Egress-Bytes`
Contributor Insights	CloudWatch Logs (regras)	`ContributorInsightRules`

Recurso do CloudWatch	Sub-recurso do CloudWatch	**`UsageType`**
	CloudWatch Logs (eventos)	`ContributorInsightEvents`
	Amazon DynamoDB (regras)	`ContributorRulesManaged`
	DynamoDB (eventos)	`ContributorEventsManaged`
Canários (Synthetics)	Executar	`Canary-runs`
Evidently	Eventos	`Evidently-event`
	Unidades de análise	`Evidently-eau`
RUM	Eventos	`RUM-event`

Analisar os dados de custo e uso do CloudWatch com o AWS Cost and Usage Report e o Athena

Outra maneira de analisar os dados de custo e uso do CloudWatch é usar o AWS Cost and Usage Report com o Amazon Athena. O AWS Cost and Usage Report contém um conjunto abrangente de dados de custo e uso. Você pode criar relatórios que monitoram custos e uso e publicá-los em um bucket do S3 de sua escolha. Também pode baixar e excluir relatórios do bucket do S3. Para obter mais informações, consulte O que são AWS Cost and Usage Reports? no Guia do usuário do AWS Cost and Usage Reports.

> **Note**
>
> Não há custo para usar o AWS Cost and Usage Report. Você paga apenas pelo armazenamento quando publica relatórios no Amazon Simple Storage Service (Amazon S3). Para obter mais informações, consulte Quotas and restrictions (Cotas e restrições) no Guia do usuário do AWS Cost and Usage Report.

O Athena é um serviço de consulta que você pode usar com o AWS Cost and Usage Report para analisar dados de custo e uso. Você pode consultar relatórios em seu bucket do S3 sem precisar baixá-los primeiro. Para obter mais informações, consulte O que é o Amazon Athena? no Guia do usuário do Amazon Athena. Para obter mais informações, consulte O que é o Amazon Athena? no Guia do usuário do Amazon Athena. Para obter mais informações sobre preços, consulte Preço do Amazon Athena.

O procedimento a seguir descreve como habilitar o AWS Cost and Usage Report e integrar o serviço ao Athena. O procedimento contém dois exemplos de consultas que você pode usar para analisar dados de custo e uso do CloudWatch.

> **Note**
>
> Você pode usar qualquer um dos exemplos de consulta neste documento. Todos os exemplos de consultas neste documento correspondem a um banco de dados chamado de costandusagereport e mostram resultados para o mês de abril de 2022. Você pode alterar essas informações. No entanto, antes de executar uma consulta, verifique se o nome do seu banco de dados corresponde ao nome do banco de dados na consulta.

Analisar dados de custo e uso do CloudWatch com AWS Cost and Usage Reports e o Athena

1. Habilite o AWS Cost and Usage Report. Para obter mais informações, consulte Creating cost and usage reports (Criar relatórios de custos e uso) no Guia do usuário do AWS Cost and Usage Report.

> **Tip**
>
> Ao criar seus relatórios, selecione Include resource IDs (Incluir IDs dos recursos). Caso contrário, os relatórios não incluirão a coluna `line_item_resource_id`. Essa linha ajuda a identificar mais precisamente os custos ao analisar dados de custo e uso.

2. Integrar o AWS Cost and Usage Report ao Athena Para obter mais informações, consulte Configuração do Athena usando modelos do AWS CloudFormation no Guia do usuário do AWS Cost and Usage Reports.
3. Faça consultas ao seus relatórios de custo e uso.

Exemplo: consulta do Athena

Você pode usar a consulta a seguir para mostrar quais recursos do CloudWatch geraram mais custos em um determinado mês.

```
SELECT
CASE
-- Metrics
WHEN line_item_usage_type LIKE '%%MetricMonitorUsage%%' THEN 'Metrics (Custom, Detailed
 monitoring management portal EMF)'
WHEN line_item_usage_type LIKE '%%Requests%%' THEN 'Metrics (API Requests)'
WHEN line_item_usage_type LIKE '%%GMD-Metrics%%' THEN 'Metrics (Bulk API Requests)'
WHEN line_item_usage_type LIKE '%%MetricStreamUsage%%' THEN 'Metric Streams'
-- Dashboard
WHEN line_item_usage_type LIKE '%%DashboardsUsageHour%%' THEN 'Dashboards'
-- Alarms
WHEN line_item_usage_type LIKE '%%AlarmMonitorUsage%%' THEN 'Alarms (Standard)'
WHEN line_item_usage_type LIKE '%%HighResAlarmMonitorUsage%%' THEN 'Alarms (High
 Resolution)'
WHEN line_item_usage_type LIKE '%%MetricInsightAlarmUsage%%' THEN 'Alarms (Metrics
 Insights)'
WHEN line_item_usage_type LIKE '%%CompositeAlarmMonitorUsage%%' THEN 'Alarms
 (Composite)'
-- Logs
WHEN line_item_usage_type LIKE '%%DataProcessing-Bytes%%' THEN 'Logs (Collect - Data
 Ingestion)'
-- Logs
WHEN line_item_usage_type LIKE '%%DataProcessingIA-Bytes%%' THEN 'Infrequent Access
 Logs (Collect - Data Ingestion)'
```

```
WHEN line_item_usage_type LIKE '%%TimedStorage-ByteHrs%%' THEN 'Logs (Storage -
 Archival)'
WHEN line_item_usage_type LIKE '%%DataScanned-Bytes%%' THEN 'Logs (Analyze - Logs
 Insights queries)'
-- Vended Logs
WHEN line_item_usage_type LIKE '%%VendedLog-Bytes%%' THEN 'Vended Logs (Delivered to
 CW)'
WHEN line_item_usage_type LIKE '%%VendedLogIA-Bytes%%' THEN 'Vended Infrequent Access
 Logs (Delivered to CW)'
WHEN line_item_usage_type LIKE '%%FH-Egress-Bytes%%' THEN 'Vended Logs (Delivered to
 Kinesis FH)'
WHEN (line_item_usage_type LIKE '%%S3-Egress-Bytes%%') OR (line_item_usage_type LIKE '%
%S3-Egress-
ComprBytes%%') THEN 'Vended Logs (Delivered to S3)'
-- Other
WHEN line_item_usage_type LIKE '%%Application-Signals%%' THEN 'Application Signals'
WHEN line_item_usage_type LIKE '%%Canary-runs%%' THEN 'Synthetics'
WHEN line_item_usage_type LIKE '%%Evidently%%' THEN 'Evidently'
WHEN line_item_usage_type LIKE '%%RUM-event%%' THEN 'RUM'
ELSE 'Others'
END AS UsageType,
-- REGEXP_EXTRACT(line_item_resource_id,'^(?:.+?:){5}(.+)$',1) as ResourceID,
-- SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
-- AND line_item_usage_account_id = '123456789012' - If you want to filter on a
 specific account, you can
remove this comment at the beginning of the line and specify an AWS account.
GROUP BY
1
ORDER BY
TotalSpend DESC,
UsageType;
```

Exemplo: consulta do Athena

Você pode usar a consulta a seguir para mostrar os resultados de `UsageType` e `Operation`. Isso mostra como os recursos do CloudWatch geraram custos. Os resultados também mostram os valores de `UsageQuantity` e `TotalSpend`, para que você possa ver os custos totais de uso.

> **Tip**
>
> Para obter mais informações sobre `UsageType`, adicione a linha a seguir a esta consulta: `line_item_line_item_description`
> Essa linha cria uma coluna chamada Description (Descrição).

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_usage_type AS UsageType,
line_item_operation AS Operation,
line_item_resource_id AS ResourceID,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation
```

Práticas recomendadas para otimizar e reduzir custos

Métricas do CloudWatch

Muitos Serviços da AWS, como o Amazon Elastic Compute Cloud (Amazon EC2), o Amazon S3 e o Amazon Data Firehose, enviam métricas automaticamente para o CloudWatch sem nenhum custo. No entanto, as métricas agrupadas nas seguintes categorias podem incorrer em custos adicionais:

- Métricas personalizadas, monitoramento detalhado e métricas incorporadas
- Solicitações de API
- Fluxos de métricas

Para obter mais informações, consulte Como usar métricas do Amazon CloudWatch.

Métricas personalizadas, monitoramento detalhado e métricas incorporadas

Métricas personalizadas

Você pode criar métricas personalizadas para organizar pontos de dados em qualquer ordem e taxa.

Todas as métricas personalizadas são cobradas pro rata por hora. Elas só são medidas quando enviadas para o CloudWatch. Para obter informações sobre o preço de métricas do CloudWatch, consulte Preço do Amazon CloudWatch.

A tabela a seguir lista os sub-recursos relevantes para as métricas do CloudWatch. A tabela inclui as strings de `UsageType` e `Operation`, que podem ajudar você a analisar e identificar custos relacionados a métricas.

> **Note**
>
> Para obter mais detalhes sobre as métricas listadas na tabela a seguir ao consultar dados de custo e uso com o Athena, verifique as strings de `Operation` em relação aos resultados mostrados para `line_item_operation`.

Sub-recurso do CloudWatch	**`UsageType`**	**`Operation`**	Finalidade

Sub-recurso do CloudWatch	**`UsageType`**	**`Operation`**	Finalidade
Métricas personalizadas	`MetricMonitorUsage`	`MetricStorage`	Métricas personalizadas
Monitoramento detalhado	`MetricMonitorUsage`	`MetricStorage:AWS/{Service}`	Monitoramento detalhado
Métricas incorporadas	`MetricMonitorUsage`	`MetricStorage:AWS/Logs-EMF`	Métricas incorporadas do Logs
Filtros de log	`MetricMonitorUsage`	`MetricStorage:AWS/CloudWatchLogs`	Filtros métricos do grupo de logs

Monitoramento detalhado

O CloudWatch tem dois tipos de monitoramento:

- Monitoramento básico

  O monitoramento básico é gratuito e está ativado automaticamente para todos os Serviços da AWS compatíveis com esse recurso.
- Monitoramento detalhado

  O monitoramento detalhado incorre em custos e adiciona aprimoramentos diferentes, a depender do AWS service (Serviço da AWS). Você pode optar por ativar o monitoramento detalhado em cada AWS service (Serviço da AWS) compatível com o recurso. Para obter mais informações, consulte Monitoramento básico e monitoramento detalhado.

> **Note**
>
> Outros Serviços da AWS são compatíveis com o monitoramento detalhado e podem usar outro nome para se referir a esse recurso. Por exemplo, o monitoramento detalhado do Amazon S3 é conhecido como métricas de solicitação.

Assim como as métricas personalizadas, o monitoramento detalhado é cobrado pro rata por hora e só é medido quando os dados são enviados para o CloudWatch. O monitoramento detalhado gera custos pelo número de métricas enviadas ao CloudWatch. Para reduzir custos, só habilite o monitoramento detalhado quando necessário. Para obter informações sobre os preços do monitoramento detalhado, consulte Preço do Amazon CloudWatch.

Exemplo: consulta do Athena

Você pode usar a consulta a seguir para mostrar as instâncias do EC2 nas quais o monitoramento detalhado foi habilitado.

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_usage_type AS UsageType,
line_item_operation AS Operation,
line_item_resource_id AS ResourceID,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_operation='MetricStorage:AWS/EC2'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation,
line_item_line_item_description
```

```
ORDER BY line_item_operation
```

Métricas incorporadas

Com o formato de métrica incorporada do CloudWatch, você pode ingerir dados de aplicações como dados de log para poder gerar métricas acionáveis. Para obter mais informações, consulte Ingerir logs de alta cardinalidade e gerar métricas com o formato de métricas incorporadas do CloudWatch.

As métricas incorporadas geram custos pelo número de logs ingeridos e arquivados e pelo número de métricas personalizadas geradas.

A tabela a seguir lista os sub-recursos relevantes para o formato de métricas incorporadas do CloudWatch. A tabela inclui as strings de `UsageType` e `Operation`, que podem ajudar você a analisar e identificar custos.

Sub-recurso do CloudWatch	**`UsageType`**	**`Operation`**	Finalidade
Métricas personalizadas	`MetricMonitorUsage`	`MetricStorage:AWS/Logs-EMF`	Métricas incorporadas do Logs
Ingestão de logs	`DataProcessing-Bytes`	`PutLogEvents`	Carrega um lote de eventos de logs no fluxo ou grupo de logs especificado
Arquivamento de logs	`TimedStorage-ByteHrs`	`HourlyStorageMetering`	Armazena logs por hora e por byte no CloudWatch Logs

Para analisar custos, use o AWS Cost and Usage Report com o Athena a fim de identificar quais métricas estão gerando custos e como eles são gerados.

Para aproveitar ao máximo os custos gerados pelo formato de métrica incorporada do CloudWatch, evite criar métricas com base em dimensões de alta cardinalidade. Assim, o CloudWatch não cria uma métrica personalizada para cada combinação de dimensão exclusiva. Para obter mais informações, consulte Dimensões.

Se você estiver usando o CloudWatch Container Insights para aproveitar o formato de métrica incorporada, poderá usar o AWS Distro para OpenTelemetry como alternativa para utilizar melhor os custos relacionados à métrica. Você pode usar o Container Insights para coletar, agregar e resumir métricas e logs de aplicações e microsserviços conteinerizados. Quando você habilita o Container Insights, o atendente do CloudWatch envia seus logs para o CloudWatch, onde eles são usados para gerar métricas incorporadas. No entanto, o atendente do CloudWatch envia apenas um número fixo de métricas para o CloudWatch e as cobranças são geradas por todas as métricas disponíveis, incluindo as que não são usadas. Com o AWS Distro para OpenTelemetry, você pode configurar e personalizar quais métricas e dimensões são enviadas para o CloudWatch. Isso ajuda a reduzir o volume de dados e o custo que o Container Insights gera. Para obter mais informações, consulte os seguintes recursos do :

- Como usar o Container Insights
- AWS Distro para OpenTelemetry

Solicitações de API

O CloudWatch tem os seguintes tipos de solicitações de API:

- Solicitações de API
- Em massa (Get)
- Contributor Insights
- Snapshot de imagem bitmap

As solicitações de API geram custos de acordo com o tipo de solicitação e o número de métricas solicitadas.

A tabela a seguir lista os tipos de solicitações de API e inclui as strings de `UsageType` e `Operation`, que podem ajudar você a analisar e identificar custos relacionados a APIs.

Tipo de solicitação de API	**`UsageType`**	**`Operation`**	Finalidade
Solicitações de API	`Requests`	`GetMetric Statistics`	Recupera as estatísti cas das métricas especificadas

Tipo de solicitação de API	**UsageType**	**Operation**	Finalidade
	Requests	ListMetrics	Lista as métricas especificadas
	Requests	PutMetricData	Publica pontos de dados de métricas no CloudWatch
	Requests	GetDashboard	Exibe detalhes dos painéis especificados
	Requests	ListDashboards	Lista os painéis em sua conta
	Requests	PutDashboard	Cria ou atualiza um painel
	Requests	DeleteDashboards	Exclui todos os painéis especificados
Em massa (Get)	GMD-Metrics	GetMetricData	Recupera valores de métricas do CloudWatch
Contributor Insights	GIRR-Metrics	GetInsightRuleReport	Retorna dados de séries temporais coletados por uma regra do Contributor Insights
Snapshot de imagem bitmap	GMWI-Metrics	GetMetricWidgetImage	Recupera um snapshot de uma ou mais métricas do CloudWatch como uma imagem bitmap

Para analisar os custos, use o Explorador de Custos e agrupe os resultados por API Operation (Operação de API).

Os custos das solicitações de API variam, e você incorre em custos quando excede o número de chamadas de API fornecidas no limite do nível gratuito da AWS.

> **Note**
>
> `GetMetricData` e `GetMetricWidgetImage` não estão incluídas no limite do nível gratuito da AWS. Para obter mais informações, consulte Como usar o nível gratuito da AWS no Guia do usuário do AWS Billing.

As solicitações de API que normalmente geram custos são `Put` e `Get`.

PutMetricData

`PutMetricData` gera custos sempre que é chamada e pode incorrer em custos significativos dependendo do caso de uso. Para obter mais informações, consulte PutMetricData na Referência de APIs do Amazon CloudWatch.

Para aproveitar ao máximo os custos gerados por `PutMetricData`, agrupe mais dados em suas chamadas de API. Dependendo do caso de uso, considere usar o CloudWatch Logs ou o formato de métrica incorporada do CloudWatch para injetar dados métricos. Para obter mais informações, consulte os seguintes recursos do :

- O que é o Amazon CloudWatch Logs? no Guia do usuário do Amazon CloudWatch Logs
- Ingerir logs de alta cardinalidade e gerar métricas com o formato de métricas incorporadas do CloudWatch
- Lowering costs and focusing on our customers with Amazon CloudWatch embedded custom metrics (Reduzir custos e focar nos clientes com as métricas incorporadas personalizadas do Amazon CloudWatch)

GetMetricData

`GetMetricData` também pode gerar custos significativos. Casos de uso comuns que geram custos envolvem ferramentas de monitoramento de terceiros que extraem dados para gerar insights. Para obter mais informações, consulte GetMetricData na Referência de APIs do Amazon CloudWatch.

Para reduzir os custos gerados por `GetMetricData`, considere extrair apenas dados monitorados e usados ou extrair dados com menos frequência. Dependendo do caso de uso, você pode usar fluxos de métricas, em vez de `GetMetricData`, o que lhe permite enviar dados quase em tempo real para terceiros a um custo menor. Para obter mais informações, consulte os seguintes recursos do :

- Usar fluxos de métricas
- Fluxos de métrica do CloudWatch: enviar métricas da AWS para parceiros e suas aplicações em tempo real

`GetMetricStatistics`

Dependendo do caso de uso, você pode usar `GetMetricStatistics` em vez de `GetMetricData`. Com `GetMetricData`, você pode recuperar dados rapidamente e em grande quantidade. No entanto, `GetMetricStatistics` está incluída no limite do nível gratuito da AWS até um milhão de solicitações de API, o que pode ajudar a reduzir custos se você não precisar recuperar esse número de métricas e pontos de dados por chamada. Para obter mais informações, consulte os seguintes recursos do :

- GetMetricStatistics na Referência de APIs do Amazon CloudWatch
- Should I use GetMetricData or GetMetricStatistics? (Devo usar GetMetricData ou GetMetricStatistics?)

> **Note**
>
> Os chamadores externos fazem chamadas de API. Atualmente, a única maneira de identificar esses chamadores é enviando uma solicitação de suporte técnico para a equipe do CloudWatch pedindo essa informação. Para obter informações sobre como criar uma solicitação de suporte técnico, consulte Como obtenho suporte técnico da AWS?.

Fluxos de métricas do CloudWatch

Com os fluxos de métricas do CloudWatch, você pode enviar métricas continuamente para destinos da AWS e de outros provedores de serviços.

Os fluxos de métricas geram custos de acordo com o número de atualizações de métricas. As atualizações de métricas sempre incluem valores para as seguintes estatísticas:

- `Minimum`
- `Maximum`
- `Sample Count`
- `Sum`

Para obter mais informações, consulte Estatísticas que podem ser transmitidas.

Para analisar os custos gerados pelos fluxos de métricas do CloudWatch, use o AWS Cost and Usage Report com o Athena. Assim, você pode identificar quais fluxos de métricas estão gerando custos e como esses custos são gerados.

Exemplo: consulta do Athena

Você pode usar a consulta a seguir para monitorar os fluxos de métricas que geram custos por nome do recurso da Amazon (ARN).

```
SELECT
SPLIT_PART(line_item_resource_id,'/',2) AS "Stream Name",
line_item_resource_id as ARN,
SUM(CAST(line_item_unblended_cost AS decimal(16,2))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
-- AND line_item_usage_account_id = '123456789012' - If you want to filter on a
 specific account, you can
remove this comment at the beginning of the line and specify an AWS account.
AND line_item_usage_type LIKE '%%MetricStreamUsage%%'
GROUP BY line_item_resource_id
ORDER BY TotalSpend DESC
```

Para reduzir os custos gerados pelos fluxos de métricas do CloudWatch, transmita apenas as métricas que agregam valor empresarial. Você também pode interromper ou pausar qualquer fluxo de métricas que não esteja usando.

Alarmes do CloudWatch

Com os alarmes do CloudWatch, você pode criar alarmes baseados em uma única métrica, alarmes baseados em uma consulta ao Metrics Insights e alarmes compostos que monitoram outros alarmes.

> **Note**
>
> Os custos para alarmes de métricas e compostos são calculados proporcionalmente por hora. Você será cobrado pelos alarmes somente enquanto seus alarmes existirem. Para otimizar os custos, certifique-se de não deixar para trás alarmes mal configurados ou de baixo valor. Para ajudar com isso, você pode automatizar a limpeza dos alarmes do CloudWatch que não são mais necessários. Para obter mais informações, consulte Limpeza de alarmes do Amazon CloudWatch em escala

Alarmes de métricas

Os alarmes de métrica têm as seguintes configurações de resolução:

- Padrão (avaliado a cada 60 segundos)
- Alta resolução (avaliado a cada 10 segundos)

Quando você cria um alarme de métrica, seus custos são baseados na configuração de resolução do seu alarme e no número de métricas às quais seu alarme faz referência. Por exemplo, um alarme de métrica que faz referência a uma métrica incorre em um custo de métrica de alarme por hora. Para obter mais informações, consulte Uso de alarmes do Amazon CloudWatch.

Se você criar um alarme de métrica que contém uma expressão matemática métrica, a qual faz referência a várias métricas, você incorrerá em um custo para cada métrica de alarme referenciada na expressão matemática métrica. Para obter informações sobre como criar um alarme de métricas que contém uma expressão matemática métrica, consulte Criar um alarme do CloudWatch com base em uma expressão matemática métrica.

Se você criar um alarme de detecção de anomalias, no qual o alarme analisa dados de métricas anteriores para criar um modelo de valores esperados, você incorrerá em um custo para cada métrica de alarme referenciada em seu alarme, além de duas métricas adicionais, para as métricas de banda superior e inferior criadas pelo modelo de detecção de anomalias. Para obter informações

sobre como criar um alarme de detecção de anomalias, consulte Criar um alarme do CloudWatch com base na detecção de anomalias.

Alarmes de consulta ao Metrics Insights

Os alarmes de consulta ao Metric Insights são um tipo específico de alarme métrico, disponível somente com a resolução padrão (avaliado a cada 60 segundos).

Quando você cria um alarme de consulta ao Metric Insights, os custos são baseados no número de métricas analisadas pela consulta que o alarme referencia. Por exemplo, um alarme de consulta ao Metric Insights que referencia uma consulta cujo filtro corresponde a dez métricas incorre no custo de dez métricas analisadas por hora. Para obter mais informações sobre preços, veja um exemplo de preço em Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

Se você criar um alarme que contenha uma consulta ao Metrics Insights e uma expressão matemática de métrica, ele será relatado como um alarme de consulta ao Metrics Insights. Se o alarme contiver uma expressão matemática métrica que referencie outras métricas, além das métricas analisadas pela consulta ao Metrics Insights, você incorrerá em um custo adicional para cada métrica de alarme referenciada na expressão matemática da métrica. Para obter informações sobre como criar um alarme de métricas que contém uma expressão matemática métrica, consulte Criar um alarme do CloudWatch com base em uma expressão matemática métrica.

Alarmes compostos

Os alarmes compostos contêm expressões de regras que especificam como devem avaliar os estados de outros alarmes para determinar seus próprios estados. Os alarmes compostos incorrem em um custo padrão por hora, independentemente de quantos outros alarmes eles avaliem. Alarmes aos quais os alarmes compostos fazem referência em expressões de regras incorrem em custos separados. Para obter mais informações, consulte Criar um alarme composto.

Tipos de uso de alarmes

A tabela a seguir lista os sub-recursos relevantes para os alarmes do CloudWatch. A tabela inclui as strings de `UsageType`, que podem ajudar você a analisar e identificar custos relacionados a alarmes.

Sub-recurso do CloudWatch	**`UsageType`**
Alarme de métricas padrão	`AlarmMonitorUsage`

Sub-recurso do CloudWatch	**`UsageType`**
High-resolution metric alarm (Alarme de métrica de alta resolução)	`HighResAlarmMonitorUsage`
Metrics Insights query alarm (Alarme de consulta ao Metrics Insights)	`MetricInsightAlarmUsage`
Alarme composto	`CompositeAlarmMonitorUsage`

Redução de custos do alarme

Para otimizar os custos gerados por alarmes de expressão matemática métrica que agregam quatro ou mais métricas, você pode agregar dados antes que os dados sejam enviados ao CloudWatch. Dessa forma, você pode criar um alarme para uma única métrica em vez de um alarme que agrega dados para várias métricas. Para obter mais informações, consulte o tópico sobre como Publicar métricas personalizadas.

Para otimizar os custos gerados pelos alarmes de consulta ao Metrics Insights, você pode garantir que o filtro usado para a consulta corresponda apenas às métricas que você deseja monitorar.

A melhor maneira de reduzir custos é remover todos os alarmes desnecessários ou não utilizados. Por exemplo, você pode excluir alarmes que avaliam métricas emitidas por recursos da AWS que não existem mais.

Exemplo: verificar os alarmes no estado ***`INSUFFICIENT_DATA`*** com ***`DescribeAlarms`***

Se você excluir um recurso, mas não os alarmes de métricas que o recurso emite, os alarmes ainda existirão e normalmente entrarão no estado `INSUFFICIENT_DATA`. Para verificar alarmes que estão no estado `INSUFFICIENT_DATA`, use o seguinte comando da AWS Command Line Interface (AWS CLI).

```
$ aws cloudwatch describe-alarms –state-value INSUFFICIENT_DATA
```

Outras maneiras de reduzir custos incluem:

- Criar alarmes para as métricas corretas.
- Verificar se há algum alarme ativado em uma região na qual você não está trabalhando.

- Lembre-se de que, embora os alarmes compostos reduzam o ruído, eles também geram custos adicionais.
- Ao decidir se deseja criar um alarme padrão ou um alarme de alta resolução, considerar o caso de uso e o valor que cada tipo de alarme traz.

CloudWatch Logs

O Amazon CloudWatch Logs tem os seguintes tipos de log:

- Logs personalizados (logs que você cria para suas aplicações)
- Logs fornecidos (logs que outros Serviços da AWS, como o Amazon Virtual Private Cloud (Amazon VPC) e o Amazon Route 53, criam em seu nome)

Para obter mais informações sobre logs fornecidos, consulte Habilitar o registro em log de determinados serviços da AWS no Manual do usuário do Amazon CloudWatch Logs.

Os logs personalizados e fornecidos geram custos com base no número de logs coletados, armazenados e analisados. Além disso, os logs fornecidos geram custos em entregas ao Amazon S3 e ao Firehose.

A tabela a seguir lista os recursos e sub-recursos relevantes do CloudWatch Logs. A tabela inclui as strings de `UsageType` e `Operation`, que podem ajudar você a analisar e identificar custos relacionados a logs.

Recurso do CloudWatch Logs	Sub-recurso do CloudWatch Logs	**`UsageType`**	**`Operation`**	Finalidade
Logs personalizados	Coleta (ingestão)	`DataProcessing-Bytes`	`PutLogEvents`	Carrega um lote de logs em um fluxo de logs específico
	Armazenamento (arquivo)	`TimedStorage-ByteHrs`	`HourlyStorageMetering`	Armazena logs por hora e por byte no

Recurso do CloudWatch Logs	Sub-recurso do CloudWatch Logs	**`UsageType`**	**`Operation`**	Finalidade
				CloudWatch Logs
	Análise (consultas do Logs Insights)	`DataScanned-Bytes`	`StartQuery`	Registra dados analisados por consultas do CloudWatch Logs Insights
Logs fornecidos	Entrega (CloudWatch Logs)	`VendedLog-Bytes`	`PutLogEvents`	Carrega um lote de logs em um fluxo de logs específico
	Entrega (Amazon S3)	`S3-Egress-ComprBytes` `S3-Egress-Bytes`	`LogDelivery`	Envia logs fornecidos (CloudWatch, Amazon S3 ou Firehose)
	Entrega (Firehose)	`FH-Egress-Bytes`	`LogDelivery`	Envia logs fornecidos (CloudWatch, Amazon S3 ou Firehose)

Para analisar custos, use o AWS Cost and Usage Report com o Athena a fim de identificar quais logs estão gerando custos e como esses custos são gerados.

Exemplo: consulta do Athena

Você pode usar a consulta a seguir para monitorar os logs que geram custos por ID do recurso.

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_resource_id AS ResourceID,
line_item_usage_type AS UsageType,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_operation IN
 ('PutLogEvents','HourlyStorageMetering','StartQuery','LogDelivery')
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation
ORDER BY
TotalSpend DESC
```

Para aproveitar ao máximo os custos gerados pelo CloudWatch Logs, considere as seguintes recomendações:

- Registre somente os eventos que agregam valor empresarial. Isso ajuda você a gerar menos custos com a ingestão.
- Altere suas configurações de retenção de logs para gerar menos custos de armazenamento. Para obter mais informações, consulte Alterar a retenção de dados de log no CloudWatch Logs no Guia do usuário do Amazon CloudWatch Logs.
- Execute consultas que o CloudWatch Logs Insights salva automaticamente no histórico. Dessa forma, você gera menos custos para análise. Para obter mais informações, consulte Exibir as consultas em execução ou o histórico de consultas no Guia do usuário do Amazon CloudWatch Logs.
- Use o atendente do CloudWatch para coletar logs do sistema e da aplicação e enviá-los ao CloudWatch. Dessa forma, você pode coletar somente eventos de log que atendam aos seus

critérios. Para obter mais informações, consulte Amazon CloudWatch Agent adiciona suporte para expressões de filtros de log.

Para reduzir os custos de logs fornecidos, considere o caso de uso e, em seguida, determine se os logs devem ser enviados para o CloudWatch ou para o Amazon S3. Para obter mais informações, consulte Logs enviados ao Amazon S3 no Guia do usuário do Amazon CloudWatch Logs.

> **Tip**
>
> Se quiser usar filtros de métricas, filtros de assinatura, CloudWatch Logs Insights e Contributor Insights, envie os logs fornecidos para o CloudWatch.
> Como alternativa, se estiver trabalhando com logs de fluxo da VPC para fins de auditoria e conformidade, envie os logs fornecidos para o Amazon S3.
> Para obter informações sobre como rastrear as cobranças geradas pela publicação de logs de fluxo da VPC em buckets do S3, consulte Uso de AWS Cost and Usage Reports e tags de alocação de custos para entender a ingestão de dados de logs de fluxo da VPC no Amazon S3.

Para obter informações adicionais sobre como aproveitar ao máximo os custos gerados pelo CloudWatch Logs, consulte Which log group is causing a sudden increase in my CloudWatch Logs bill? (Qual grupo de logs causou um aumento súbito na minha fatura do CloudWatch Logs?).

Usar painéis do Amazon CloudWatch

Os painéis do Amazon CloudWatch são páginas iniciais personalizáveis no console do CloudWatch que você pode usar para monitorar seus recursos em uma única visualização, mesmo os recursos distribuídos em regiões diferentes. Você pode usar os painéis do CloudWatch para criar visualizações personalizadas das métricas e dos alarmes para os recursos da AWS.

Com os painéis, você pode criar o seguinte:

- Uma única visualização para determinadas métricas e alarmes para ajudar a avaliar a integridade des seus recursos e de suas aplicações em uma ou mais regiões. Você pode selecionar a cor usada para cada métrica em cada gráfico, para que possa monitorar com facilidade a mesma métrica em vários gráficos.
- Um guia estratégico que fornece orientação para os membros da equipe durante eventos operacionais sobre como responder a incidentes específicos.
- Uma visualização comum de medições de recursos e aplicativos críticos que pode ser compartilhada pelos membros da equipe para obter um fluxo de comunicação mais rápido durante eventos operacionais.

Se você tiver várias contas da AWS, poderá configurar a observabilidade entre contas do CloudWatch e depois criar, em suas contas de monitoramento, painéis avançados entre várias contas. Esses painéis podem incluir gráficos de métricas de contas de origem e widgets do CloudWatch Logs Insights com consultas dos grupos de logs das contas de origem. Além disso, os alarmes que você cria na conta de monitoramento podem acompanhar métricas nas contas de origem. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

É possível criar painéis via console ou usando a AWS CLI ou a operação `PutDashboard` da API. Você pode adicionar painéis a uma lista de favoritos, onde é possível acessar não apenas seus painéis favoritos, mas também os painéis visitados recentemente. Para obter mais informações, consulte Adicionar um painel à sua lista de favoritos.

Para acessar os painéis do CloudWatch, você precisa de um destes itens:

- A política `AdministratorAccess`
- A política `CloudWatchFullAccess`
- Uma política personalizada que inclui uma ou mais destas permissões específicas:
  - `cloudwatch:GetDashboard` e `cloudwatch:ListDashboards` para poder visualizar painéis

- `cloudwatch:PutDashboard` para poder criar ou modificar painéis
- `cloudwatch:DeleteDashboards` para poder excluir painéis

Conteúdo

Criar um painel do CloudWatch

Para começar, crie um painel do CloudWatch. É possível criar vários painéis e adicionar painéis a uma lista de favoritos. Você não está limitado ao número de painéis que pode ter na Conta da AWS. Todos os painéis são globais. Eles não são específicos de uma região.

O procedimento a seguir mostra como criar um painel usando o console do CloudWatch. Você poderá então usar a operação `PutDashboard` da API para criar um painel da interface de linha de comando. A operação da API contém uma string JSON que define o conteúdo do painel. Para obter mais informações sobre como criar um painel usando a operação `PutDashboard` da API, consulte PutDashboard na Referência da API do Amazon CloudWatch.

> **Tip**
>
> Se estiver criando um novo painel com a operação `PutDashboard` da API, você poderá usar a string JSON de um painel que já existe.

Para criar um painel via console

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e Create dashboard (Criar painel).
3. Na caixa de diálogo Create new dashboard (Criar novo painel), insira um nome para o painel e escolha Create dashboard (Criar painel).

   Se você usar o nome CloudWatch-Default ou CloudWatch-Default-***ResourceGroupName***, o painel será mostrado na visão geral da página inicial do CloudWatch sob o Default Dashboard (Painel padrão). Para ter mais informações, consulte Conceitos básicos do Amazon CloudWatch.
4. Na caixa de diálogo Add to this dashboard (Adicionar a este painel), siga um destes procedimentos:
   - Para adicionar um gráfico ao painel, escolha Line (Linha) ou Stacked area (Área empilhada) e, em seguida, escolha Configure (Configurar). Na caixa de diálogo Add metric graph (Adicionar gráfico da métrica), selecione as métricas para o gráfico e escolha Create widget (Criar widget). Se uma métrica não aparecer na caixa de diálogo porque não há dados publicados há mais de 14 dias, você poderá adicioná-la manualmente. Para ter mais informações, consulte Criar gráficos de métricas manualmente em um painel do CloudWatch.
   - Para adicionar um número exibindo uma métrica para o painel, escolha Number (Número) e, e, seguida, Configure (Configurar). Na caixa de diálogo Add metric graph (Adicionar gráfico da métrica), selecione as métricas para o gráfico e escolha Create widget (Criar widget).
   - Para adicionar um bloco de texto ao painel, escolha Text (Texto) e Configure (Configurar). Na caixa de diálogo New text widget (Novo widget de texto), em Markdown, adicione e formate o texto usando Markdown. Em seguida, escolha Create widget (Criar widget).
5. (Opcional) Escolha Add widget (Adicionar widget) e repita a etapa 4 para adicionar outro widget ao painel. Você pode repetir essa etapa diversas vezes.

   Para cada gráfico no painel, há um ícone de informações no canto superior direito. Escolha esse ícone para ver as descrições das métricas no gráfico.
6. Escolha Save dashboard (Salvar painel).

Painel da observabilidade entre contas do CloudWatch

Se você tiver várias contas da AWS, poderá configurar a observabilidade entre contas do CloudWatch e depois criar, em suas contas de monitoramento, painéis avançados entre várias contas. Você pode pesquisar, visualizar e analisar facilmente métricas, logs e rastreamentos sem barreiras entre contas.

Para obter mais informações sobre a configuração da observabilidade entre contas do CloudWatch, consulte Observabilidade entre contas do CloudWatch.

Com a observabilidade entre contas do CloudWatch, você usar um painel em uma conta de monitoramento para fazer o seguinte:

- Pesquisar, visualizar e criar gráficos das métricas que residem nas contas de origem. Um único gráfico pode incluir métricas de várias contas.
- Criar alarmes na conta de monitoramento que observem as métricas nas contas de origem.
- Visualizar eventos de logs de grupos de logs localizados nas contas de origem e executar consultas do CloudWatch Logs Insights de grupos de logs nas contas de origem. Uma única consulta do CloudWatch Logs Insights em uma conta de monitoramento pode consultar vários grupos de logs em várias contas de origem de uma só vez.
- Visualizar os nós das contas de origem em um mapa de rastreamento no X-Ray. Depois, você pode filtrar o mapa para contas de origem específicas.

Quando você está conectado a uma conta de monitoramento, um emblema azul de conta de monitoramento aparece no canto superior direito de cada página compatível com a funcionalidade de observabilidade entre contas do CloudWatch.

Painéis entre contas e entre regiões

Você pode criar painéis entre contas e entre regiões, que resumem os dados do CloudWatch de várias contas e regiões da AWS em um único painel. A partir desse painel de alto nível, você pode obter uma visualização de toda a aplicação e fazer uma busca detalhada em painéis mais específicos sem precisar fazer login e sair de contas nem alternar regiões.

É possível criar painéis entre contas e entre regiões no AWS Management Console e programaticamente.

Pré-requisito

Antes de criar um painel entre contas e entre regiões, você deve habilitar pelo menos uma conta de compartilhamento e uma conta de monitoramento. Além disso, para poder usar o console do CloudWatch a fim de criar um painel entre contas, é necessário habilitar a funcionalidade entre contas para o console. Para ter mais informações, consulte Console do CloudWatch entre contas e entre regiões.

Criar e usar um painel entre contas e entre regiões com o AWS Management Console

Você pode usar o AWS Management Console para criar um painel entre contas e entre regiões.

Para criar um painel entre contas e entre regiões

1. Faça login na conta de monitoramento.
2. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação, escolha Painéis.
4. Escolha um painel ou crie um novo.
5. Na parte superior da tela, é possível alternar entre contas e regiões. Ao criar seu painel, você pode incluir widgets de várias contas e regiões. Os widgets incluem gráficos, alarmes e widgets do CloudWatch Logs Insights.

Criar um gráfico com métricas de diferentes contas e regiões

1. Faça login na conta de monitoramento.
2. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas).
4. Selecione a conta e a região das quais você deseja adicionar métricas. Você pode selecionar a conta e a região nos menus suspensos Account (Conta) e Region (Região) no canto superior direito da tela.
5. Adicione as métricas desejadas ao gráfico. Para ter mais informações, consulte Criar gráficos de métricas.
6. Repita as etapas 4 a 5 para adicionar métricas de outras contas e regiões.

7. (Opcional) Escolha a guia Graphed metrics (Métricas com gráficos) e adicione uma função matemática métrica que use as métricas escolhidas. Para ter mais informações, consulte Usar matemática de métricas.

   Você também pode configurar um único gráfico para incluir várias funções `SEARCH`. Cada pesquisa pode se referir a uma conta ou região diferente.
8. Ao terminar o gráfico, escolha Actions (Ações), Add to dashboard (Adicionar ao painel).

   Selecione seu painel entre contas e escolha Add to dashboard (Adicionar ao painel).

Adicionar um alarme de uma conta diferente ao painel de controle entre contas

1. Faça login na conta de monitoramento.
2. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
3. Na parte superior da página, escolha a conta onde o alarme está localizado.
4. No painel de navegação, selecione Alarmes.
5. Marque a caixa de seleção ao lado do alarme que você deseja adicionar e escolha Add to dashboard (Adicionar ao painel).
6. Selecione o painel entre contas ao qual você deseja adicioná-lo e escolha Add to dashboard (Adicionar ao painel).

Criar um painel entre contas e entre regiões de maneira programática

Você pode usar as APIs da AWS e SDKs para criar painéis de forma programática. Para obter mais informações, consulte PutDashboard.

Para habilitar painéis entre contas e entre regiões, adicionamos novos parâmetros à estrutura do corpo do painel, conforme mostrado na tabela e nos exemplos a seguir. Para obter mais informações sobre a estrutura geral do corpo do painel, consulte Estrutura do corpo do painel e sintaxe.

Parâmetro	Use	Escopo	Padrão
`accountId`	Especifica o ID da conta onde o widget ou a métrica está localizado.	Widget ou métrica	Conta que está atualmente conectada

Parâmetro	Use	Escopo	Padrão
`region`	Especifica a região da métrica.	Widget ou métrica	Região atual selecionada no console

Os exemplos a seguir ilustram a origem JSON para widgets em um painel entre contas e entre regiões.

Este exemplo define o campo `accountId` como o ID da conta de compartilhamento no nível do widget. Especifica que todas as métricas nesse widget virão dessa conta de compartilhamento e região.

```
{
    "widgets": [
        {
            ...
            "properties": {
              "metrics": [
                     …..
                     ],
                    "accountId": "111122223333",
                    "region": "us-east-1"
           }
       }
  ]
}
```

Este exemplo define o campo `accountId` de forma diferente no nível de cada métrica. Neste exemplo, as diferentes métricas nesta expressão matemática métrica vêm de contas de compartilhamento e regiões diferentes.

```
{
"widgets": [
    {
        ...
        "properties": {
            "metrics": [
                [ { "expression": "SUM(METRICS())", "label": "[avg: ${AVG}]
 Expression1", "id": "e1", "stat": "Sum" } ],
```

```
                [ "AWS/EC2", "CPUUtilization", { "id": "m2", "accountId":
 "5555666677778888", "region": "us-east-1", "label": "[avg: ${AVG}] ApplicationALabel
 " } ],
                [ ".", ".", { "id": "m1", "accountId": "9999000011112222", "region":
 "eu-west-1", "label": "[avg: ${AVG}] ApplicationBLabel" } ]
            ],
            "view": "timeSeries",
            "region":"us-east-1", ---> home region of the metric. Not present in above
 example
            "stacked": false,
            "stat": "Sum",
            "period": 300,
            "title": "Cross account example"
        }
    }
]
}
```

Este exemplo mostra um widget de alarme.

```
{
    "type": "metric",
    "x": 6,
    "y": 0,
    "width": 6,
    "height": 6,
    "properties": {
        "accountID": "111122223333",
        "title": "over50",
        "annotations": {
            "alarms": [
                "arn:aws:cloudwatch:us-east-1:379642911888:alarm:over50"
            ]
        },
        "view": "timeSeries",
        "stacked": false
    }
}
```

Este exemplo é para um widget do CloudWatch Logs Insights.

```
{
    "type": "log",
```

```
    "x": 0,
    "y": 6,
    "width": 24,
    "height": 6,
    "properties": {
        "query": "SOURCE 'route53test' | fields @timestamp, @message\n| sort @timestamp
 desc\n| limit 20",
        "accountId": "111122223333",
        "region": "us-east-1",
        "stacked": false,
        "view": "table"
    }
}
```

Outra maneira de criar painéis programaticamente é primeiro criar um no AWS Management Console e, depois, copiar a origem JSON desse painel. Para fazer isso, carregue o painel e escolha Actions (Ações), View/edit source (Exibir/editar origem). Depois, você pode copiar esse painel JSON para usar como um modelo a fim de criar painéis semelhantes.

Crie painéis flexíveis com variáveis de painel

Use variáveis de painel para criar painéis flexíveis que possam exibir rapidamente conteúdos diferentes em vários widgets, dependendo do valor de um campo de entrada no painel. Por exemplo, é possível criar um painel que pode alternar rapidamente entre diferentes funções do Lambda ou IDs de instância do Amazon EC2, ou um que pode alternar para diferentes regiões da AWS.

Após criar um painel que use uma variável, será possível copiar o mesmo padrão de variáveis para outros painéis existentes.

O uso de variáveis de painel melhora o fluxo de trabalho operacional das pessoas que usam seus painéis. Isso também pode reduzir seus custos, pois você está usando variáveis de painel em um painel em vez de criar vários painéis semelhantes.

> **Note**
>
> Se você compartilhar um painel contendo variáveis, as pessoas com as quais você compartilhar o painel não poderão alterar os valores das variáveis.

Tipos de variáveis de painel

A variável de painel pode ser uma variável de propriedade ou uma variável de padrão.

- As variáveis de propriedade alteram todas as instâncias de uma propriedade em todos os widgets no painel. Essa propriedade pode ser qualquer propriedade JSON na fonte JSON de um painel, como `region`. Ou pode ser um nome de dimensão para uma métrica, como `InstanceID` ou `FunctionName`.

  Para ver um tutorial que usa uma variável de propriedade, consulte Tutorial: crie um painel Lambda com o nome da função como variável.

  Para obter mais informações sobre a fonte JSON de painéis, consulte Estrutura e sintaxe do corpo de painéis. No console do CloudWatch, é possível ver a fonte JSON de qualquer painel personalizado escolhendo Ações, Visualizar/editar fonte.
- As variáveis de padrão usam um padrão de expressão regular para alterar toda uma propriedade JSON ou somente uma parte dela.

  Para ver um tutorial que use uma variável de padrão, consulte Tutorial: crie um painel que use um padrão de expressão regular para alternar entre regiões.

As variáveis de propriedade se aplicam à maioria dos casos de uso e são menos complexas de configurar.

Tópicos

- Tutorial: crie um painel Lambda com o nome da função como variável
- Tutorial: crie um painel que use um padrão de expressão regular para alternar entre regiões
- Copiar uma variável para um outro painel

Tutorial: crie um painel Lambda com o nome da função como variável

As etapas deste procedimento ilustram como criar um painel flexível que mostre uma variedade de gráficos de métricas, usando uma variável de propriedade. Isso inclui uma caixa de seleção suspensa no painel que pode ser usada para alternar as métricas em todos os gráficos entre as diferentes funções do Lambda.

Outros exemplos de casos de uso desse tipo de painel incluem o uso `InstanceId` como variável para criar um painel de métricas com uma lista suspensa para IDs de instância. Como alternativa, é possível criar um painel que use `region` como variável para exibir o mesmo conjunto de métricas de diferentes regiões.

Para usar uma variável de propriedade do painel para criar um painel Lambda flexível

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis, Criar painel.
3. Insira um nome para o painel e escolha Criar painel.
4. Adicione widgets ao painel que exibam métricas para uma função do Lambda. Ao criar esses widgets, especifique Lambda, Por nome de função para as métricas do widget. Para a função, especifique uma das funções do Lambda que você deseja incluir nesse painel.

   Para obter mais informações sobre a adição de widgets a um painel, consulte Criar e trabalhar com widgets nos painéis do CloudWatch.
5. Depois de adicionar os widgets, ao visualizar o painel, escolha Ações, Variáveis, Criar uma variável.
6. Escolha a variável Propriedade.
7. Em Propriedade que a variável altera, escolha FunctionName.
8. Em Tipo de entrada, para esse caso de uso, recomendamos escolher Menu de seleção (suspenso). Isso cria um menu suspenso no painel onde é possível selecionar o nome da função do Lambda para a qual exibir as métricas.

   Se isso fosse para um painel que alterna entre apenas dois ou três valores diferentes para uma variável, o botão de opção seria uma boa escolha.

   Se você preferir inserir ou colar valores para a variável, escolha Entrada de texto. Essa opção não inclui uma lista suspensa ou botões de opção.
9. Ao escolher Menu de seleção (suspenso), você deverá em seguida escolher se deseja preencher o menu inserindo valores ou usando uma pesquisa de métricas. Para esse caso de uso, vamos supor que você tenha um grande número de funções do Lambda e não queira inserir todas elas manualmente. Escolha Usar os resultados de uma pesquisa de métricas e faça o seguinte:

   a. Escolha Consultas pré-criadas, Lambda, Erros.

(Escolher Erros não adiciona a métrica Erros ao painel. No entanto, isso preenche rapidamente a caixa de seleção da variável FunctionName.)

b. Escolha Por nome de função e, em seguida, escolha Pesquisar.

Sob o botão Pesquisar, você verá FunctionName selecionado. Você também verá uma mensagem sobre quantos valores de dimensão FunctionName foram encontrados para preencher a caixa de entrada.

10. (Opcional) Para obter mais configurações, escolha Configurações secundárias e siga um ou mais dos procedimentos a seguir:

- Para personalizar o nome da sua variável, insira o nome em Nome da variável personalizada.
- Para personalizar o rótulo para o campo de entrada da variável, insira o rótulo em Rótulo de entrada.
- Para definir o valor padrão para essa variável quando o painel for aberto pela primeira vez, insira o padrão em Valor padrão.

11. Escolha Adicionar variável.

Uma caixa de seleção suspensa FunctionName aparecerá na parte superior do painel. É possível selecionar uma função do Lambda nessa caixa e todos os widgets que usam a variável exibirão informações sobre a função selecionada.

Posteriormente, se você adicionar ao painel mais widgets que observem as métricas do Lambda com a dimensão FunctionName, eles usarão automaticamente a variável.

Tutorial: crie um painel que use um padrão de expressão regular para alternar entre regiões

As etapas deste procedimento ilustram como criar um painel flexível que pode alternar entre regiões. Este tutorial usa uma variável de padrão de expressão regular em vez de uma variável de propriedade. Para ver um tutorial que usa uma variável de propriedade, consulte Tutorial: crie um painel Lambda com o nome da função como variável.

Para muitos casos de uso, é possível criar um painel que alterne entre regiões usando uma variável de propriedade. Mas, caso os widgets dependam de algum nome do recurso da Amazon (ARN) que inclua nomes de regiões, você deverá usar uma variável de padrão para alterar os nomes das regiões dentro dos ARNs.

Para usar uma variável de padrão de painel para criar um painel flexível de várias regiões

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis, Criar painel.
3. Insira um nome para o painel e escolha Criar painel.
4. Adicione widgets ao painel. Ao adicionar os widgets que você deseja que exibam dados específicos da região, evite especificar quaisquer dimensões com valores que apareçam em apenas uma região. Por exemplo, para métricas do Amazon EC2, especifique métricas que sejam agregadas em vez de métricas que usem InstanceID como uma dimensão.

   Para obter mais informações sobre a adição de widgets a um painel, consulte Criar e trabalhar com widgets nos painéis do CloudWatch.
5. Depois de adicionar os widgets, ao visualizar o painel, escolha Ações, Variáveis, Criar uma variável.
6. Escolha Variável de padrão.
7. Em Propriedade que a variável altera, insira o nome da região atual do painel, como **`us-east-2`**.

   Você terá inserido a região correta se o rótulo abaixo dessa caixa exibir os widgets que serão afetados pela variável.
8. Em Tipo de entrada, para esse caso de uso, selecione o botão de opção.
9. Em Definir como as entradas são preenchidas, escolha Criar uma lista de valores personalizados.
10. Em Criar seus valores personalizados, insira as regiões entre as quais você deseja alternar, com uma região em cada linha. Depois de cada região, insira uma vírgula e, em seguida, o rótulo a ser exibido para esse botão de opção. Por exemplo:

    `us-east-1, N. Virginia`

    `us-east-2, Ohio`

    `eu-west-3, Paris`

    À medida que você preenche os valores personalizados, o painel Visualização é atualizado para exibir a aparência dos botões de opção.
11. (Opcional) Para obter mais configurações, escolha Configurações secundárias e siga um ou mais dos procedimentos a seguir:

- Para personalizar o nome da sua variável, insira o nome em Nome da variável personalizada.
- Para personalizar o rótulo para o campo de entrada da variável, insira o rótulo em Rótulo de entrada. Para este tutorial, insira **`Region:`**.

  Se você inserir um valor aqui, o painel de visualização será atualizado para exibir a aparência dos botões de opção.
- Para definir o valor padrão para essa variável quando o painel for aberto pela primeira vez, insira o padrão em Valor padrão.

12. Escolha Adicionar variável.

    O painel será exibido, com um rótulo Região: ao lado dos botões de opção das regiões próximas à parte superior. Quando você alterna entre regiões, todos os widgets que usam a variável exibirão informações sobre a região selecionada.

Copiar uma variável para um outro painel

Após criar um painel com variáveis úteis, será possível copiar essas variáveis para outros painéis existentes. Para obter mais informações sobre variáveis de painel, consulte Crie painéis flexíveis com variáveis de painel.

Para copiar uma variável para um outro painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis e, em seguida, escolha o painel que contém o widget numérico que deseja excluir. Insira uma string para encontrar painéis com nomes correspondentes, se necessário.
3. Escolha Ações, Variáveis, Gerenciar variáveis.
4. Escolha o botão de opção ao lado da variável que você deseja copiar e escolha Copiar para outro painel.
5. Escolha a caixa de seleção e comece a digitar o nome do painel para o qual você deseja copiar a variável.
6. Selecione o nome do painel e escolha Copiar variável.

Criar e trabalhar com widgets nos painéis do CloudWatch

Use os tópicos desta seção para criar e trabalhar com gráficos, alarmes e widgets de texto em seus painéis.

Conteúdo

Adicionar ou remover um gráfico de um painel do CloudWatch

É possível adicionar gráficos que contenham uma ou mais métricas ao painel do CloudWatch. Os tipos de gráficos que podem ser adicionados ao painel incluem Line (Linhas), Stacked area (Área empilhada), Number (Números), Gauge (Medidor), Bar (Barras) e Pie (Pizza). Os gráficos poderão ser removidos do painel quando não forem mais necessários. Os procedimentos nesta seção descrevem como adicionar e remover gráficos do painel. Para obter informações sobre como editar um gráfico no painel, consulte Editar um gráfico em um painel do CloudWatch.

Para adicionar um gráfico a um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Escolha o símbolo + e, em seguida, selecione o tipo de gráfico que deseja adicionar ao seu painel. Em seguida, escolha Avançar.

- Se você selecionar Line (Linhas), Stacked area (Área empilhada), Bar (Barras) ou Pie (Pizza), escolha Metrics (Métricas).

4. Na guia Navegar, pesquise ou procure as métricas a serem representadas graficamente e selecione as que você deseja.
5. (Opcional) Para alterar o intervalo de tempo do gráfico, selecione um dos intervalos de tempo predefinidos na parte superior da tela. Os intervalos de tempo variam de 1 hora a 1 semana (1h, 3h, 12h, 1d, 3d ou 1w).

   Para definir seu próprio intervalo de tempo, escolha Custom (Personalizado).

   - (Opcional) Para que esse widget continue usando o intervalo de tempo selecionado, mesmo que o intervalo de tempo para o resto do painel seja alterado posteriormente, escolha Persistir intervalo de tempo.
6. (Opcional) Para alterar o tipo de widget do seu gráfico, use o menu suspenso próximo aos intervalos de tempo predefinidos.
7. (Opcional) Em Graphed metrics (Métricas em gráfico), é possível adicionar um rótulo dinâmico à métrica e alterar o rótulo, a cor do rótulo, a estatística e o período da métrica. Também é possível determinar a posição dos rótulos no eixo Y da esquerda para a direita.

   a. Para adicionar um rótulo dinâmico, escolha Graphed metrics (Métricas em gráfico) e Dynamic labels (Rótulos dinâmicos). Rótulos dinâmicos exibem uma estatística sobre sua métrica na legenda do gráfico. Os rótulos dinâmicos são atualizados automaticamente sempre que o painel ou o gráfico são atualizados. Por padrão, os valores dinâmicos que você adiciona a rótulos aparecem no início dos rótulos. Para ter mais informações, consulte Usar rótulos dinâmicos.

   b. Para alterar a cor de uma métrica, escolha o quadrado colorido próximo à métrica.

   c. Para alterar a estatística, selecione o menu suspenso em Statistic (Estatística) e, em seguida, escolha um novo valor. Para obter mais informações, consulte Estatísticas.

   d. Para alterar o período, selecione o menu suspenso sob a coluna Period (Período) e, em seguida, escolha um novo valor.
8. Se você estiver criando um widget de medidor, deverá escolher a guia Opções e especificar os valores Mínimo e Máximo a serem usados nas duas extremidades do medidor.
9. (Opcional) Para personalizar o eixo Y, escolha a guia Options (Opções). É possível inserir um rótulo personalizado em Left Y Axis (Eixo Y esquerdo) no campo do rótulo. Se o gráfico exibir valores no lado direito do eixo Y, também será possível personalizar esse rótulo. Também é

possível definir limites mínimos e máximos para os valores do eixo Y de modo que o gráfico exiba somente os intervalos de valores especificados.

10. (Opcional) Para adicionar ou editar anotações horizontais em gráficos de linhas ou áreas empilhadas, ou para adicionar limites aos widgets de medidor, escolha Opções:

    a. Para adicionar uma anotação horizontal ou limite, escolha Adicionar anotação horizontal ou Adicionar limite.

    b. Em Rótulo, insira um rótulo para a anotação e escolha o ícone de marca de seleção.

    c. Em Value (Valor), escolha o ícone de papel e caneta ao lado do valor atual. Em seguida, insira seu novo valor. Depois de inserir o valor, escolha o ícone de marca de verificação.

    d. Em Fill (Preenchimento), selecione o menu suspenso e especifique como sua anotação usará o sombreamento. Você pode escolher None (Nenhum), Above (Acima), Between (Entre) ou Below (Abaixo). Para alterar a cor do preenchimento, escolha o quadrado colorido próximo à anotação.

    e. Para Axis (Eixo), especifique se a anotação é exibida no lado esquerdo ou no lado direito do eixo Y.

    f. Para ocultar uma anotação, desmarque a caixa de seleção ao lado da anotação que deseja ocultar.

    g. Para excluir uma anotação, selecione X em Actions (Ações).

    > **Note**
    >
    > É possível repetir essas etapas para adicionar várias anotações horizontais ou limites ao mesmo gráfico ou medidor.

11. (Opcional) Para adicionar ou editar anotações verticais, escolha Options (Opções):

    a. Para adicionar uma anotação vertical, selecione Add vertical annotation (Adicionar anotação vertical).

    b. Em Label (Rótulo), escolha o ícone de papel e caneta ao lado da anotação atual. Em seguida, insira sua nova anotação. Se desejar mostrar apenas a data e hora, deixe o campo de rótulo em branco.

    c. Em Date (Data), escolha a data e a hora atuais e insira as novas data e hora.

    d. Em Fill (Preenchimento), selecione o menu suspenso e especifique como sua anotação usará o sombreamento. Você pode escolher None (Nenhum), Above (Acima), Between

(Entre) ou Below (Abaixo). Para alterar a cor do preenchimento, selecione o quadrado colorido próximo à anotação.

e. Para ocultar uma anotação, desmarque a caixa de seleção ao lado da anotação que deseja ocultar.

f. Para excluir uma anotação, selecione X em Actions (Ações).

> **Note**
>
> Repita essas etapas para adicionar várias anotações verticais ao mesmo gráfico.

12. Selecione Create widget (Criar widget).
13. Escolha Save dashboard (Salvar painel).

Para remover um gráfico de um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. No canto superior direito do gráfico que deseja remover, escolha Widget actions (Ações do widget) e, em seguida, escolha Delete (Excluir).
4. Escolha Save dashboard (Salvar painel).

Criar gráficos de métricas manualmente em um painel do CloudWatch

Se não houve uma métrica publicada nos últimos 14 dias, você não poderá encontrá-la quando estiver procurando métricas para adicionar a um gráfico em um painel do CloudWatch. Use as seguintes etapas para adicionar qualquer métrica manualmente a um gráfico existente.

Para adicionar uma métrica que você não pode encontrar ao procurar um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. O painel já deve conter o gráfico no qual você deseja adicionar a métrica. Se não, crie o gráfico e adicione qualquer métrica a ele. Para ter mais informações, consulte Adicionar ou remover um gráfico de um painel do CloudWatch .
4. Escolha Actions (Ações), View/edit source (Exibir/editar origem).

Um bloco JSON é exibido. O bloco especifica os widgets no painel e seu conteúdo. O exemplo a seguir mostra uma parte desse bloco, que define um gráfico.

```
{
            "type": "metric",
            "x": 0,
            "y": 0,
            "width": 6,
            "height": 3,
            "properties": {
                "view": "singleValue",
                "metrics": [
                    [ "AWS/EBS", "VolumeReadOps", "VolumeId",
 "vol-1234567890abcdef0" ]
                ],
                "region": "us-west-1"
            }
        },
```

Neste exemplo, a seção a seguir define a métrica mostrada no gráfico.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ]
```

5. Adicione uma vírgula após o colchete de fechamento, se ainda não houver uma, e adicione uma seção semelhante entre colchetes após a vírgula. Nesta nova seção, especifique o namespace, o nome da métrica e as dimensões necessárias da métrica que você está adicionando ao gráfico. Veja um exemplo a seguir.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ],
[ "MyNamespace", "MyMetricName", "DimensionName", "DimensionValue" ]
```

Para obter mais informações sobre a formatação de métricas em JSON, consulte Properties of a Metric Widget Object (Propriedades de um objeto widget de métrica).

6. Escolha Atualizar.

Trabalhar com gráficos existentes

Siga os procedimentos destas seções para editar e modificar seus widgets de gráficos existentes do painel.

Tópicos

Editar um gráfico em um painel do CloudWatch

Você pode editar os gráficos que adiciona ao painel do CloudWatch. Você pode alterar o título, a estatística ou o período de um gráfico. É possível adicionar, atualizar e remover métricas de gráficos. Se o gráfico contiver mais de uma métrica, você poderá reduzir a confusão ocultando métricas que não está usando. Os procedimentos nesta seção descrevem como editar um gráfico no painel. Para obter informações sobre como criar um gráfico, consulte Adicionar ou remover gráficos em um painel do CloudWatch.

New interface

Para editar um gráfico em um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. No canto superior direito do gráfico que deseja editar, escolha Widget actions (Ações do widget) e, em seguida, escolha Edit (Editar).
4. Para alterar o título do gráfico, escolha o ícone de papel e caneta ao lado do título atual. Insira o novo título e, em seguida, escolha Apply (Aplicar).
5. (Opcional) Para alterar o intervalo de tempo do seu gráfico, selecione um dos intervalos de tempo predefinidos na área superior do gráfico. Os intervalos de tempo variam de 1 hora a 1 semana (1h, 3h, 12h, 1d, 3d ou 1w).

   Para definir seu próprio intervalo de tempo, escolha Custom (Personalizado).

- (Opcional) Para que esse widget continue usando o intervalo de tempo selecionado, mesmo que o intervalo de tempo para o resto do painel seja alterado posteriormente, escolha Persistir intervalo de tempo.

6. Para alterar o tipo de widget do seu gráfico, use o menu suspenso próximo aos intervalos de tempo predefinidos.
7. Em Graphed metrics (Métricas em gráfico), é possível adicionar um rótulo dinâmico à métrica e alterar o rótulo, a cor do rótulo, a estatística e o período da métrica. Também é possível determinar a posição dos rótulos no eixo Y da esquerda para a direita.

   a. Para adicionar um rótulo dinâmico para uma métrica, escolha Dynamic labels (Rótulos dinâmicos). Rótulos dinâmicos exibem uma estatística sobre a métrica na legenda do gráfico. Os rótulos dinâmicos são atualizados automaticamente sempre que o painel ou o gráfico são atualizados. Por padrão, os valores dinâmicos que você adiciona a rótulos aparecem no início dos rótulos. Para ter mais informações, consulte Usar rótulos dinâmicos.

   b. Para alterar a cor de uma métrica, escolha o quadrado colorido próximo à métrica.

   c. Para alterar a estatística, escolha o valor da estatística sob a coluna Statistic (Estatística) e, em seguida, escolha um novo valor. Para ter mais informações, consulte Estatísticas.

   d. Para alterar o período, escolha o valor do período sob a coluna Period (Período) e, em seguida, escolha um novo valor.

8. Para adicionar ou editar anotações horizontais, selecione Options (Opções):

   a. Para adicionar uma anotação horizontal, selecione Add horizontal annotation (Adicionar anotação horizontal).

   b. Em Label (Rótulo), escolha o ícone de papel e caneta ao lado da anotação atual. Em seguida, insira sua nova anotação. Depois de inserir sua anotação, escolha o ícone de marca de verificação.

   c. Em Value (Valor), escolha o ícone de papel e caneta ao lado do valor da métrica atual. Em seguida, insira o novo valor da métrica. Depois de inserir o valor, selecione o ícone de marca de verificação.

   d. Em Fill (Preenchimento), escolha o menu suspenso abaixo da coluna e, em seguida, especifique como sua anotação usará o sombreamento. Você pode escolher None (Nenhum), Above (Acima), Between (Entre) ou Below (Abaixo). Se você escolher Between (Entre), outro novo campo de rótulo e valor será exibido.

> **Tip**
>
> Você pode alterar a cor do preenchimento escolhendo o quadrado colorido próximo à anotação.

e. Para Axis (Eixo), especifique se a anotação é exibida no lado esquerdo ou no lado direito do eixo Y.

f. Para ocultar uma anotação, desmarque a caixa de seleção ao lado da anotação que deseja ocultar no gráfico.

g. Para excluir uma anotação, selecione X na coluna Actions (Ações).

> **Note**
>
> Você pode repetir essas etapas para adicionar várias anotações horizontais ao mesmo gráfico.

9. Para adicionar ou editar anotações verticais, escolha Options (Opções):

a. Para adicionar uma anotação vertical, selecione Add vertical annotation (Adicionar anotação vertical).

b. Em Label (Rótulo), escolha o ícone de papel e caneta ao lado da anotação atual. Em seguida, insira sua nova anotação. Depois de inserir sua anotação, escolha o ícone de marca de verificação.

> **Tip**
>
> Para mostrar apenas a data e hora, deixe o campo de rótulo em branco.

c. Em Date (Data), escolha a data e a hora atuais. Em seguida, insira a nova data e hora.

d. Em Fill (Preenchimento), escolha o menu suspenso abaixo da coluna e, em seguida, especifique como sua anotação usará o sombreamento. Você pode escolher None (Nenhum), Above (Acima), Between (Entre) ou Below (Abaixo). Se você escolher Between (Entre), um novo campo de rótulo e valor será exibido.

> **Tip**
>
> Você pode alterar a cor do preenchimento escolhendo o quadrado colorido próximo à anotação.

> **Note**
>
> Repita essas etapas para adicionar várias anotações verticais ao mesmo gráfico.

e. Para ocultar uma anotação, desmarque a caixa de seleção ao lado da anotação que deseja ocultar no gráfico.

f. Para excluir uma anotação, selecione X na coluna Actions (Ações).

10. Para personalizar o eixo Y, escolha a guia Options (Opções). Em Left Y Axis (Eixo Y esquerdo), é possível adicionar um rótulo personalizado para Label (Rótulo). Se o gráfico exibir valores no eixo Y direito, esse rótulo também poderá ser personalizado. Também é possível definir valores mínimos e máximos nos valores do eixo Y para que o gráfico exiba somente o intervalo de valores especificado.
11. Após terminar de fazer as alterações, selecione Update widget (Atualizar widget).

Para ocultar ou alterar a posição de uma legenda do gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. No canto superior direito do gráfico que deseja editar, escolha Widget actions (Ações do widget). Escolha Legend (Legenda) e selecione Hidden (Oculta), Bottom (Inferior) ou Right (Direita).

Para ocultar temporariamente as métricas em um gráfico em um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Selecione o quadrado colorido para a métrica que deseja ocultar no rodapé do gráfico. Um X aparece no quadrado colorido quando o ponteiro do mouse passa sobre ele e se torna cinza ao ser escolhido.

4. Para restaurar a métrica oculta, desmarque o X no quadrado cinza.

Original interface

Para editar um gráfico em um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Passe o mouse sobre o canto superior direito do gráfico que deseja editar. Escolha Widget actions (Ações do widget) e, em seguida, escolha Edit (Editar).
4. Para alterar o título do gráfico, escolha o ícone de lápis próximo título atual. Em seguida, insira o novo título.
5. Para alterar o intervalo de tempo do gráfico, escolha um dos intervalos de tempo predefinidos na área superior do gráfico. Eles variam de 1 hora a 1 semana (1h,3h,12h,1d,3d ou1w).
   - Para definir seu próprio intervalo de tempo, escolha custom (personalizado).
6. Para alterar o tipo de widget do gráfico, selecione a guia Graph options (Opções do gráfico). Você pode escolher Line (Linha), Stacked area (Área empilhada),Number (Número),Bar (Barras) ou Pie (Pizza).

   > **Tip**
   >
   > É possível alterar o tipo de widget do gráfico escolhendo o menu suspenso próximo aos intervalos de tempo predefinidos.

7. Em Graphed metrics (Métricas em gráfico), é possível adicionar um rótulo dinâmico à métrica e alterar o rótulo, a cor do rótulo, a estatística e o período da métrica. Também é possível determinar a posição dos rótulos no eixo Y da esquerda para a direita.

   a. Para adicionar um rótulo dinâmico para uma métrica, escolha Dynamic labels (Rótulos dinâmicos). Rótulos dinâmicos exibem uma estatística sobre a métrica na legenda do gráfico. Os rótulos dinâmicos são atualizados automaticamente sempre que o painel ou o gráfico são atualizados. Por padrão, os valores dinâmicos que você adiciona a rótulos aparecem no início dos rótulos. Para ter mais informações, consulte Usar rótulos dinâmicos.

   b. Para alterar a cor de uma métrica, escolha o quadrado colorido próximo à métrica.

c. Para alterar a estatística, escolha o valor da estatística sob a coluna Statistic (Estatística) e, em seguida, escolha um novo valor. Para ter mais informações, consulte Estatísticas.

d. Para alterar o período, escolha o valor do período sob a coluna Period (Período) e, em seguida, escolha um novo valor.

8. Para adicionar ou editar anotações horizontais, selecione Graph options (Opções de gráfico):

a. Para adicionar uma anotação horizontal, selecione Add horizontal annotation (Adicionar anotação horizontal).

b. Em Label (Rótulo), escolha o ícone de lápis ao lado da anotação atual. Em seguida, insira sua nova anotação. Depois de inserir sua anotação, escolha o ícone de marca de verificação.

c. Em Value (Valor), escolha o ícone de lápis ao lado do valor da métrica atual. Em seguida, insira o novo valor da métrica. Depois de inserir o valor, selecione o ícone de marca de verificação.

d. Em Fill (Preenchimento), escolha o menu suspenso abaixo da coluna e, em seguida, especifique como sua anotação usará o sombreamento. Você pode escolher None (Nenhum), Above (Acima), Between (Entre) ou Below (Abaixo). Se você escolher Between (Entre), um novo campo de rótulo e valor será exibido.

> **Tip**
>
> Você pode alterar a cor do preenchimento escolhendo o quadrado colorido próximo à anotação.

e. Para Axis (Eixo), especifique se a anotação é exibida no lado esquerdo ou no lado direito do eixo Y.

f. Para ocultar uma anotação, desmarque a caixa de seleção ao lado da anotação que deseja ocultar no gráfico.

g. Para excluir uma anotação, selecione X na coluna Actions (Ações).

> **Note**
>
> Você pode repetir essas etapas para adicionar várias anotações horizontais ao mesmo gráfico.

9. Para adicionar ou editar anotações verticais, escolha Graph options (Opções do gráfico):

   a. Para adicionar uma anotação vertical, selecione Add vertical annotation (Adicionar anotação vertical).

   b. Em Label (Rótulo), escolha o ícone de lápis ao lado da anotação atual. Em seguida, insira sua nova anotação. Depois de inserir sua anotação, escolha o ícone de marca de verificação.

   > **Tip**
   >
   > Para mostrar apenas a data e hora, deixe o campo de rótulo em branco.

   c. Em Date (Data), escolha o ícone de lápis ao lado da data e da hora atuais. Em seguida, insira a nova data e hora.

   d. Em Fill (Preenchimento), escolha o menu suspenso abaixo da coluna e, em seguida, especifique como sua anotação usará o sombreamento. Você pode escolher None (Nenhum), Above (Acima), Between (Entre) ou Below (Abaixo). Se você escolher Between (Entre), um novo campo de rótulo e valor será exibido.

   > **Tip**
   >
   > Você pode alterar a cor do preenchimento escolhendo o quadrado colorido próximo à anotação.

   > **Note**
   >
   > Repita essas etapas para adicionar várias anotações verticais ao mesmo gráfico.

   e. Para ocultar uma anotação, desmarque a caixa de seleção ao lado da anotação que deseja ocultar no gráfico.

   f. Para excluir uma anotação, selecione X na coluna Actions (Ações).

10. Para personalizar o eixo Y, selecione Graph options (Opções do gráfico). Em Left Y Axis (Eixo Y esquerdo), é possível adicionar um rótulo personalizado para Label (Rótulo). Se o gráfico exibir valores no eixo Y direito, esse rótulo também poderá ser personalizado. Também é possível definir valores mínimos e máximos nos valores do eixo Y para que o gráfico exiba somente o intervalo de valores especificado.

11. Após terminar de fazer as alterações, selecione Update widget (Atualizar widget).

Para ocultar ou alterar a posição de uma legenda do gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Mova o ponteiro do mouse sobre o canto superior direito do gráfico que deseja editar e escolha Widget actions (Ações do widget). Escolha Legend (Legenda) e selecione Hidden (Oculta), Bottom (Inferior) ou Right (Direita).

Para ocultar temporariamente as métricas em um gráfico em um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Selecione o quadrado colorido para a métrica que deseja ocultar no rodapé do gráfico. Um X aparece no quadrado colorido quando o ponteiro do mouse passa sobre ele e se torna cinza ao ser escolhido.
4. Para restaurar a métrica oculta, desmarque o X no quadrado cinza.

Mover ou redimensionar um gráfico em um painel do CloudWatch

Você pode organizar e redimensionar gráficos em seu painel do CloudWatch.

Para mover um gráfico em um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Execute um destes procedimentos:

   - Passe o mouse sobre o título do gráfico até que o ícone de seleção apareça. Selecione e arraste o gráfico para um novo local no painel.
   - Para mover o widget para o canto superior esquerdo ou inferior esquerdo do painel, escolha as reticências verticais no canto superior direito do widget para abrir o menu Ações do widget. Em seguida, escolha Mover e escolha para onde mover o widget.
4. Escolha Save dashboard (Salvar painel).

Para redimensionar um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Passe o mouse sobre o gráfico e arraste o canto inferior direito para aumentar ou diminuir o tamanho do gráfico. Pare de arrastar o canto quando tiver o tamanho desejado.
4. Escolha Save dashboard (Salvar painel).

Para aumentar um gráfico temporariamente

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Selecione o gráfico. Como alternativa, passe o mouse sobre o título do gráfico e escolha Widget actions (Ações de widget), Enlarge (Aumentar).

Renomear um gráfico em um painel do CloudWatch

Você pode alterar o nome padrão que o CloudWatch atribui a um gráfico no painel.

Para alterar o nome de um gráfico em um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Passe o mouse sobre o título do gráfico e escolha Widget actions (Ações de widget) e Edit (Editar).
4. Na tela Edit graph (Editar gráfico), na parte superior, escolha o título do gráfico.
5. Em Title (Título), insira um novo nome e escolha Ok (marca de seleção). No canto inferior direito da tela Edit graph (Editar gráfico), selecione Update widget (Atualizar widget).

Adicionar um widget do explorador de métricas a um painel do CloudWatch

Os widgets do explorador de métricas incluem gráficos de vários recursos que têm a mesma etiqueta ou compartilham a mesma propriedade do recurso, como um tipo de instância. Esses widgets permanecem atualizados à medida que os recursos correspondentes são criados ou

excluídos. Adicionar widgets do explorador de métricas ao painel ajuda a solucionar problemas de seu ambiente de forma mais eficiente.

Por exemplo, você pode monitorar sua frota de instâncias do EC2 atribuindo etiquetas que representam seus ambientes, como produção ou teste. Em seguida, é possível usar essas etiquetas para filtrar e agregar as métricas operacionais, como `CPUUtilization`, para entender a integridade e a performance das instâncias do EC2 associadas a cada etiqueta.

As etapas a seguir explicam como adicionar um widget do explorador de métricas a um painel com o console. Também é possível adicioná-lo de maneira programática usando o AWS CloudFormation. Para obter mais informações, consulte Definição de objeto do widget do explorador de métricas e AWS::CloudWatch::Dashboard.

Para adicionar um widget do explorador de métricas a um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha o nome do painel no qual deseja adicionar o widget do explorador de métricas.
4. Escolha o símbolo +.
5. Escolha Explorer (Explorador) e depois Next (Próximo).

> **Note**
>
> Você deve ter optado pela nova exibição do painel para poder adicionar um widget do explorador de métricas. Para aceitar, escolha Dashboards (Painéis) no painel de navegação e selecione try out the new interface (experimentar a nova interface) no banner na parte superior da página.

6. Execute um destes procedimentos:
   - Para usar um modelo, escolha Pre-filled Explorer widget (Widget do explorador pré-preenchido) e selecione um modelo a ser usado.
   - Para criar uma visualização personalizada, escolha Empty Explorer widget (Esvaziar widget do explorador).
7. Escolha Criar.

Se você usou um modelo, o widget aparecerá em seu painel com as métricas selecionadas. Se você estiver contente com o widget do explorador e o painel, selecione Save dashboard (Salvar painel).

Caso não tenha usado um modelo, prossiga para as próximas etapas.

8. No novo widget em Explorer (Explorador), na caixa Metrics (Métricas), escolha uma única métrica ou todas as métricas disponíveis de um serviço.

   Se preferir, você pode repetir essa etapa para adicionar mais métricas.

9. Para cada métrica selecionada, o CloudWatch exibe a estatística que será usada imediatamente após o nome da métrica. Para alterar isso, escolha o nome da estatística e selecione a estatística desejada.

10. Em From (De), escolha uma etiqueta ou uma propriedade do recurso para filtrar seus resultados.

    Depois de fazer isso, você pode, opcionalmente, repetir essa etapa para escolher mais etiquetas ou propriedades do recurso.

    Se você escolher vários valores da mesma propriedade, como dois tipos de instância do EC2, o explorador exibirá todos os recursos que correspondem à propriedade escolhida. É tratado como uma operação OR.

    Se você escolher propriedades ou etiquetas diferentes, como a etiqueta **`Production`** e o tipo de instância M5, somente os recursos que correspondem a todas essas seleções serão exibidos. Isso é tratado como uma operação AND.

11. (Opcional) Em Aggregate by (Agregar por), escolha uma estatística a ser usada para agregar as métricas. Em seguida, ao lado de for (para), escolha como agregar a métrica na lista. É possível agregar todos os recursos que são exibidos no momento ou agregar por uma única etiqueta ou propriedade do recurso.

    Dependendo de como você escolher agregar, o resultado pode ser uma única série temporal ou várias séries temporais.

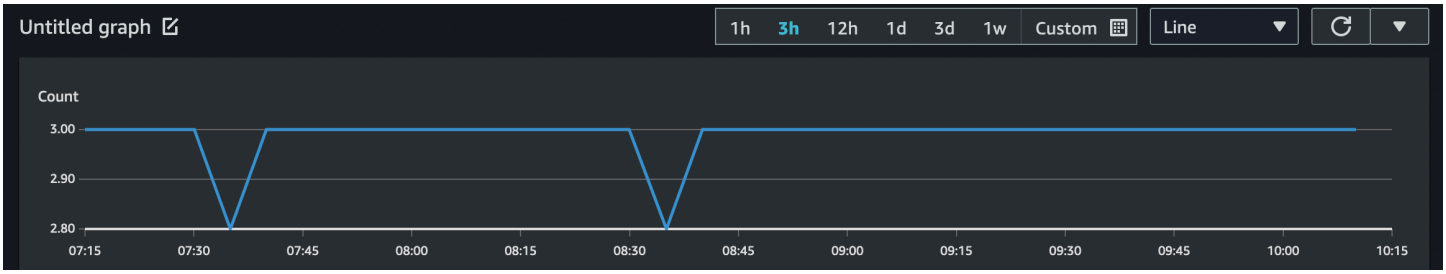
12. Em Split by (Dividir por), você pode optar por dividir um único gráfico com várias séries temporais em vários gráficos. A divisão pode ser feita por critérios variados, que você escolhe em Split by (Dividir por).

13. Em Graph options (Opções de gráficos), é possível refinar o gráfico alterando o período, o tipo de gráfico, o posicionamento da legenda e o layout.

14. Se você estiver contente com o widget do explorador e o painel, selecione Save dashboard (Salvar painel).

Adicionar ou remover um widget de linhas em um painel do CloudWatch

Com o widget de linhas, é possível comparar métricas ao longo de vários períodos. Também é possível usar o recurso de zoom de minimapa do widget para inspecionar seções de gráficos de linhas sem alterar entre as visualizações com zoom aumentado ou diminuído. Os procedimentos nesta seção descrevem como adicionar e remover um widget de linhas em um painel do CloudWatch. Para obter informações sobre como usar o recurso de zoom de minimapa do widget com gráficos de linhas, consulte Aumentar o zoom em um gráfico de linhas ou de área empilhada.



Para adicionar um widget de linhas a um painel

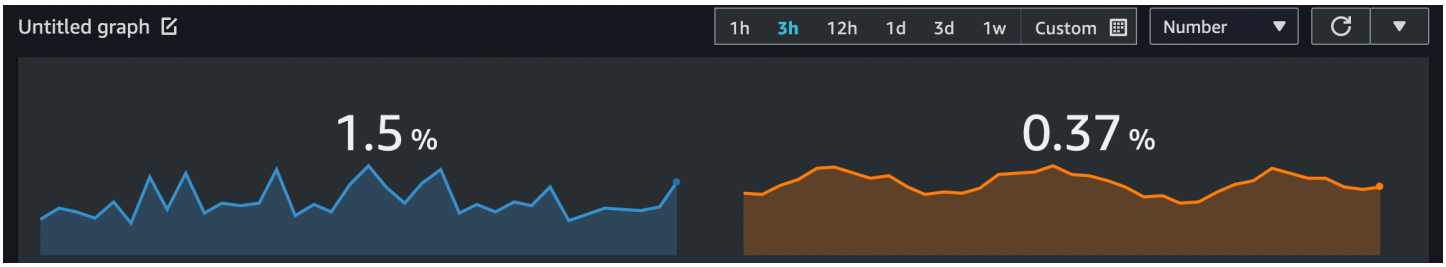
1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Escolha o símbolo + e selecione Line (Linha).
4. Escolha Metrics (Métricas).
5. Selecione Browse (Procurar) e, em seguida, selecione a métrica que deseja mostrar no gráfico.
6. Selecione Create widget (Criar widget) e, em seguida, escolha Save dashboard (Salvar painel).

Para remover um widget de linhas de um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. No canto superior direito do widget de linhas que deseja remover, escolha Widget actions (Ações do widget) e, em seguida, escolha Delete (Excluir).
4. Escolha Save dashboard (Salvar painel).

Adicionar ou remover um widget numérico de um painel do CloudWatch

Com o widget numérico, é possível observar os valores e tendências mais recentes da métrica assim que eles aparecem. Como o widget numérico inclui o recurso de sparkline, é possível visualizar as metades superior e inferior das tendências métricas em um único gráfico. Os procedimentos nesta seção descrevem como adicionar e remover um widget numérico em um painel do CloudWatch.



> **Note**
>
> Somente a nova interface oferece suporte ao recurso de sparkline. Quando você cria um widget numérico, o recurso de sparkline é incluído automaticamente.

Para adicionar um widget numérico a um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Escolha o símbolo + e selecione Number (Número).
4. Na guia Navegar, pesquise ou navegue até a métrica que você deseja exibir.
5. (Opcional) Para alterar a cor do recurso de sparkline, escolha Graphed metrics (Métricas em gráfico) e selecione a caixa de cores ao lado do rótulo da métrica. Um menu no qual é possível escolher uma cor diferente ou inserir um código de cores hexadecimal com seis dígitos para especificar uma cor é então exibido.
6. (Opcional) Para desativar o recurso de sparkline, escolha Options (Opções). Em Sparkline, a caixa de seleção.
7. (Opcional) Para alterar o intervalo de tempo do seu widget numérico, selecione um dos intervalos de tempo predefinidos na área superior do widget. Os intervalos de tempo variam de 1 hora a 1 semana (1h, 3h, 12h, 1d, 3d ou 1w).

   Para definir seu próprio intervalo de tempo, escolha Custom (Personalizado).

- (Opcional) Para que esse widget continue usando o intervalo de tempo selecionado, mesmo que o intervalo de tempo para o resto do painel seja alterado posteriormente, escolha Persistir intervalo de tempo.

8. (Opcional) Para que o widget numérico exiba um agregado (1h, 3h, 12h, 1d, 3d ou 1s).

   Para definir seu próprio intervalo de tempo, escolha Custom (Personalizado).

   - (Opcional) Para que esse widget exiba uma média do valor da métrica em todo o intervalo de tempo, em vez do valor mais recente, escolha Opções. O valor do intervalo de tempo mostra o valor de todo o intervalo de tempo.

9. Selecione Create widget (Criar widget) e escolha Save dashboard (Salvar painel).

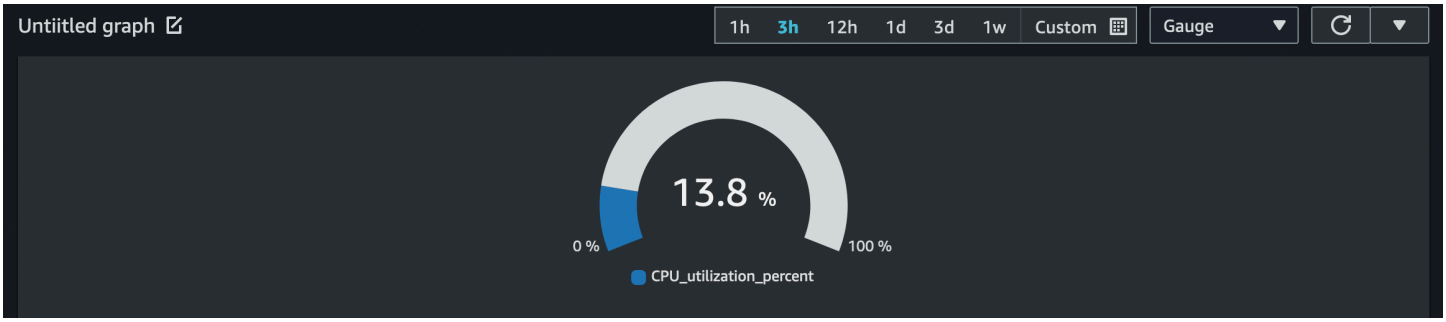
> **Tip**
>
> Você pode desativar o recurso de sparkline do widget numérico na tela do painel. No canto superior direito do widget numérico que deseja modificar, escolha Widget actions (Ações do widget). Selecione Sparkline e, em seguida, escolha Hide sparkline (Ocultar sparkline).

Para remover um widget numérico de um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e, em seguida, escolha o painel que contém o widget numérico que deseja excluir.
3. No canto superior direito do widget numérico que deseja remover, escolha Widget actions (Ações do widget) e, em seguida, escolha Delete (Excluir).
4. Escolha Save dashboard (Salvar painel).

Adicionar ou remover um widget de medidor em um painel do CloudWatch

Com o widget de medidor, é possível visualizar valores métricos localizados entre os intervalos. Por exemplo, você poderá usar o widget de medidor para adicionar ao gráfico percentuais e a utilização da CPU a fim de observar e diagnosticar quaisquer possíveis problemas de desempenho. Os procedimentos nesta seção descrevem como adicionar e remover um widget de medidor em um painel do CloudWatch.



> **Note**
>
> Somente a nova interface no console do CloudWatch oferece suporte à criação do widget de medidor. É necessário definir um intervalo de medidor ao criar esse widget.

Para adicionar um widget de medidor a um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Na tela do painel, escolha o símbolo + e, em seguida, selecione Gauge (Medidor).
4. Selecione Browse (Procurar) e, em seguida, selecione a métrica que deseja mostrar no gráfico.
5. Escolha Options. Em Gauge range (Faixa do medidor), defina valores para Min (Mínimo) e Max (Máximo). Para porcentagens, como utilização da CPU, recomendamos definir os valores de `Min` como `0` e de `Max` como `100`.
6. (Opcional) Para alterar a cor do widget de medidor, escolha Graphed metrics (Métricas em gráfico) e selecione a caixa de cores ao lado do rótulo da métrica. Um menu no qual é possível escolher uma cor diferente ou inserir um código de cores hexadecimal com seis dígitos para especificar uma cor é então exibido.
7. Selecione Create widget (Criar widget) e escolha Save dashboard (Salvar painel).

Para remover um widget de medidor de um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e, em seguida, escolha o painel que contém o widget de medidor que deseja excluir.
3. No canto superior direito do widget de medidor que deseja excluir, escolha Widget actions (Ações do widget) e, em seguida, escolha Delete (Excluir).

4. Escolha Save dashboard (Salvar painel).

Adicionar um widget personalizado a um painel do CloudWatch

Um widget personalizado é um widget de painel do CloudWatch que pode chamar qualquer função do AWS Lambda com parâmetros personalizados. Em seguida, ele exibe o HTML ou JSON retornado. Os widgets personalizados são uma forma simples de criar uma visualização de dados personalizada em um painel. Se você pode gravar um código do Lambda e criar HTML, poderá criar um widget personalizado útil. Além disso, a Amazon fornece vários widgets personalizados prontos que você pode criar sem nenhum código.

Quando você cria uma função Lambda para usar como um widget personalizado, recomendamos incluir o prefixo customWidget no nome da função. Isso ajuda a saber quais funções do Lambda são seguras para uso quando você adicionar widgets personalizados ao seu painel.

Os widgets personalizados se comportam como outros widgets em seu painel. Eles podem ser atualizados e atualizados automaticamente, redimensionados e movidos. Reagem ao intervalo de tempo do painel.

Se você configurou a funcionalidade entre contas do console do CloudWatch, poderá adicionar um widget personalizado criado em uma conta aos painéis de outras contas. Para ter mais informações, consulte Console do CloudWatch entre contas e entre regiões.

Também é possível usar widgets personalizados em seu próprio site com o recurso de compartilhamento de painel do CloudWatch. Para ter mais informações, consulte Compartilhar painéis do CloudWatch.

Tópicos

Detalhes sobre widgets personalizados

Os widgets personalizados funcionam desta maneira:

1. O painel do CloudWatch chama a função Lambda que contém o código do widget. Ele aprova parâmetros personalizados que são definidos no widget.
2. A função Lambda retorna uma string de HTML, JSON ou Markdown. O Markdown é retornado como JSON no seguinte formato:

```
{"markdown":"markdown content"}
```

3. O painel exibe o HTML ou JSON retornado.

Se a função retornar HTML, a maioria das etiquetas HTML será compatível. Você pode usar estilos Cascading Style Sheets (CSS) e gráficos vetoriais escaláveis (SVG) para criar exibições sofisticadas.

O estilo padrão de elementos HTML, como links e tabelas, segue o estilo dos painéis do CloudWatch. Você pode personalizar esse estilo com estilos em linha, usando a etiqueta `<style>`. Também é possível desativar os estilos padrão, incluindo um único elemento HTML com a classe de `cwdb-no-default-styles`. O exemplo a seguir desativa os estilos padrão: `<div class="cwdb-no-default-styles"></div>`.

Cada chamada feita por um widget personalizado para o Lambda inclui um elemento `widgetContext` com o seguinte conteúdo, para fornecer informações de contexto úteis ao desenvolvedor da função Lambda.

```
{
    "widgetContext": {
        "dashboardName": "Name-of-current-dashboard",
        "widgetId": "widget-16",
        "accountId": "012345678901",
        "locale": "en",
        "timezone": {
            "label": "UTC",
            "offsetISO": "+00:00",
            "offsetInMinutes": 0
        },
        "period": 300,
        "isAutoPeriod": true,
        "timeRange": {
            "mode": "relative",
            "start": 1627236199729,
            "end": 1627322599729,
            "relativeStart": 86400012,
```

```
            "zoom": {
                "start": 1627276030434,
                "end": 1627282956521
             }
        },
        "theme": "light",
        "linkCharts": true,
        "title": "Tweets for Amazon website problem",
        "forms": {
            "all": {}
        },
        "params": {
            "original": "param-to-widget"
        },
        "width": 588,
        "height": 369
    }
}
```

Estilo CSS padrão

Os widgets personalizados fornecem os seguintes elementos de estilo CSS padrão:

- Você pode usar a classe de CSS btn para adicionar um botão. Ele gira uma âncora (<a>) em um botão como no exemplo a seguir:

  ```
  <a class="btn" href=https://amazon.com”>Open Amazon</a>
  ```

- É possível usar a classe de CSS btn btn-primary para adicionar um botão primário.
- Os seguintes elementos são estilizados por padrão: table,select, cabeçalhos (h1, h2 e h3), texto pré-formatado (pre), input eárea de texto.

Usar o parâmetro describe

É altamente recomendável oferecer suporte ao describe em suas funções, mesmo que ele retorne apenas uma string vazia. Caso você não oferecça suporte, e ele for chamado no widget personalizado, ele exibirá o conteúdo do widget como se fosse documentação.

Se incluir o parâmetro describe, a função Lambda retornará a documentação no formato Markdown e não fará mais nada.

Quando você cria um widget personalizado no console, depois de selecionar a função Lambda, um botão Get documentation (Obter documentação) é exibido. Se você escolher este botão, a função será invocada com o parâmetro describe, e a documentação da função será retornada. Se a documentação estiver bem formatada em markdown, o CloudWatch analisará a primeira entrada na documentação que está rodeada por três caracteres de crase simples (```) no YAML. Em seguida, preencherá automaticamente a documentação nos parâmetros. Veja a seguir um exemplo dessa documentação bem formatada.

```
``` yaml
echo: <h1>Hello world</h1>
```
```

Segurança e JavaScript

Por motivos de segurança, o JavaScript não é permitido no HTML retornado. Remover o JavaScript impede problemas de escalonamento de permissão, em que o gravador da função Lambda injeta um código que poderia ser executado com permissões maiores do que o usuário que está visualizando o widget no painel.

Se o HTML retornado contiver algum código JavaScript ou outras vulnerabilidades de segurança conhecidas, ele será removido do HTML antes de ser renderizado no painel. Por exemplo, as etiquetas **<iframe>** e **<use>** não são permitidas e são removidas.

Widgets personalizados não serão executados por padrão em um painel. Para fazer isso, você deve permitir explicitamente que um widget personalizado seja executado se você confiar na função do Lambda que ele invoca. É possível escolher permitir uma vez ou permitir sempre, tanto para widgets individuais quanto para todo o painel. Você também pode negar permissões para widgets individuais e para todo o painel.

Interatividade no widget personalizado

Mesmo que o JavaScript não seja permitido, há outras maneiras de permitir a interatividade com o HTML retornado.

- Qualquer elemento no HTML retornado pode ser marcado com configuração especial em uma etiqueta `<cwdb-action>`, que pode exibir informações em pop-ups, solicitar confirmação em cliques e chamar qualquer função Lambda quando esse elemento for escolhido. Por exemplo, é possível definir botões que chamem qualquer API da AWS usando uma função Lambda. O HTML

retornado pode ser definido para substituir o conteúdo do widget do Lambda existente ou ser exibido dentro de um modal.

- O HTML retornado pode conter links que abrem novos consoles, abrem outras páginas de clientes ou carregam outros painéis.
- O HTML pode incluir o atributo `title` em um elemento, o que fornecerá informações adicionais se o usuário passar o mouse sobre esse elemento.
- O elemento pode incluir seletores CSS, como `:hover`, que pode invocar animações ou outros efeitos de CSS. Também é possível mostrar ou ocultar elementos na página.

Definição e uso do <cwdb-action>

O elemento `<cwdb-action>` define um comportamento no elemento imediatamente anterior. O conteúdo do `<cwdb-action>` é HTML para exibir ou um bloco JSON de parâmetros para passar para uma função Lambda.

A seguir há um exemplo de um elemento `<cwdb-action>`.

```
<cwdb-action
     action="call|html"
     confirmation="message"
     display="popup|widget"
     endpoint="<lambda ARN>"
     event="click|dblclick|mouseenter">

     html | params in JSON
</cwdb-action>
```

- action: os valores válidos são `call`, que chama uma função Lambda, e `html`, que exibe qualquer HTML contido em `<cwdb-action>`. O padrão é `html`.
- confirmation: exibe uma mensagem de confirmação que deve ser confirmada antes da ação ser executada, permitindo que o cliente cancele.
- display: os valores válidos são popup e `widget`, que substituem o conteúdo do próprio widget. O padrão é `widget`.
- endpoint: o nome do recurso da Amazon (ARN) da função Lambda a ser invocada. Isso será necessário se `action` for `call`.

- event: define o evento no elemento anterior que invoca a ação. Os valores válidos são `click`, `dblclick` e `mouseenter`. O evento `mouseenter` só pode ser usado em combinação com a ação `html`. O padrão é `click`.

Exemplos

Veja a seguir um exemplo de como usar a etiqueta `<cwdb-action>` para criar um botão que reinicializa uma instância do Amazon EC2 usando uma chamada de função Lambda. Exibe o êxito ou a falha da chamada em um pop-up.

```
<a class="btn">Reboot Instance</a>
<cwdb-action action="call" endpoint="arn:aws:lambda:us-
east-1:123456:function:rebootInstance" display="popup">
       { "instanceId": "i-342389adbfef" }
</cwdb-action>
```

O próximo exemplo exibe mais informações em um pop-up.

```
<a>Click me for more info in popup</a>
<cwdb-action display="popup">
   <h1>Big title</h1>
   More info about <b>something important</b>.
</cwdb-action>
```

Este exemplo é um botão Next (Próximo) que substitui o conteúdo de um widget por uma chamada para uma função Lambda.

```
<a class="btn btn-primary">Next</a>
<cwdb-action action="call" endpoint="arn:aws:lambda:us-
east-1:123456:function:nextPage">
   { "pageNum": 2 }
</cwdb-action>
```

Criar um widget personalizado

Para criar um widget personalizado, é possível usar um dos exemplos fornecidos pela AWS, ou é possível criar seus próprios. Os exemplos da AWS incluem exemplos em JavaScript e Python e são criados por uma pilha do AWS CloudFormation. Para obter uma lista de exemplos, consulte Exemplo de widgets personalizados.

Para criar um widget personalizado em um painel do CloudWatch

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Escolha o símbolo +.
4. Escolha Custom widget (Widget personalizado).
5. Use um dos seguintes métodos:
   - Para usar um widget personalizado de exemplo fornecido pela AWS, faça o seguinte:
     a. Selecione o exemplo na caixa suspensa.

        O console do AWS CloudFormation é iniciado em um novo navegador. No console do AWS CloudFormation, faça o seguinte:

     b. (Opcional) Personalize o nome da pilha do AWS CloudFormation.
     c. Faça seleções para quaisquer parâmetros usados pelo exemplo.
     d. Selecione Estou ciente de que o AWS CloudFormation pode criar recursos do IAM e escolha Criar pilha.
   - Para criar seu próprio widget personalizado fornecido pela AWS, faça o seguinte:
     a. Escolha Próximo.
     b. Escolha selecionar sua função Lambda em uma lista ou insira seu nome do recurso da Amazon (ARN). Se você selecioná-la de uma lista, especifique também a região onde a função está e a versão a ser usada.
     c. Em Parameters (Parâmetros), faça seleções para quaisquer parâmetros usados pela função.
     d. Insira um título para o widget.
     e. Em Update on (Atualizar em), configure quando o widget deve ser atualizado (quando a função Lambda deve ser chamada novamente). Pode ser uma ou mais destas opções: Refresh para atualizá-lo quando o painel for atualizado automaticamente, Resize (Redimensionar) para atualizá-lo sempre que o widget for redimensionado, ou Time Range (Período) para atualizá-lo sempre que o intervalo de tempo do painel for ajustado, inclusive quando os gráficos são ampliados.
     f. Se estiver contente com a pré-visualização, selecione Create widget (Criar widget).

Exemplo de widgets personalizados

A AWS fornece exemplos de widgets personalizados em JavaScript e Python. É possível criar esses widgets de exemplo usando o link para cada widget desta lista. Se preferir, você pode criar e personalizar um widget usando o console do CloudWatch. Os links desta lista abrem um console do AWS CloudFormation e usam um link de criação rápida do AWS CloudFormation para criar o widget personalizado.

Também é possível acessar as exemplos de widget personalizados no GitHub.

Seguindo esta lista, os exemplos completos do widget Echo são exibidos para cada idioma.

JavaScript

Widgets personalizados de exemplo em JavaScript

- Echo: um ecoador básico que você pode usar para testar como HTML aparece em um widget personalizado, sem precisar gravar um novo widget.
- Hello world: um widget iniciante muito básico.
- Custom widget debugger (Depurador do widget personalizado): um widget depurador que exibe informações úteis sobre o ambiente do runtime do Lambda.
- Query CloudWatch Logs Insights (Consultar o CloudWatch Logs Insights): execute e edite consultas do CloudWatch Logs Insights.
- Run Amazon Athena queries (Executar consultas do Amazon Athena): executa e edita consultas do Athena.
- Chamar API da AWS: chama qualquer API somente leitura da AWS e exibe os resultados no formato JSON.
- Fast CloudWatch bitmap graph (Grafo de bitmap rápido do CloudWatch): renderize gráficos do CloudWatch usando no lado do servidor, para exibição rápida.
- Text widget from CloudWatch dashboard (Widget de texto do painel do CloudWatch): exibe o primeiro widget de texto do painel do CloudWatch especificado.
- CloudWatch metric data as a table (Dados de métrica do CloudWatch como uma tabela): exibe dados métricos brutos do CloudWatch em uma tabela.
- Amazon EC2 table (Tabela do Amazon EC2): exibe as principais instâncias do EC2 por utilização da CPU. Este widget também inclui um botão Reboot (Reiniciar), que é desabilitado por padrão.

- Implantações do AWS CodeDeploy: exibe implantações do CodeDeploy.
- Relatório do AWS Cost Explorer: exibe um relatório sobre o custo de cada serviço da AWS no intervalo de tempo selecionado.
- Display content of external URL (Exibir conteúdo de URL externa): exibe o conteúdo de uma URL acessível externamente.
- Display an Amazon S3 object (Exibir um objeto do Amazon S3): exibe um objeto em um bucket do Amazon S3 em sua conta.
- Simple SVG pie chart (Gráfico de pizza SVG simples) exemplo de um widget gráfico baseado em SVG.

Python

Widgets personalizados de exemplo em Python

- Echo: um ecoador básico que pode ser usado para testar como HTML aparece em um widget personalizado, sem precisar gravar um novo widget.
- Hello world: um widget iniciante muito básico.
- Custom widget debugger (Depurador do widget personalizado): um widget depurador que exibe informações úteis sobre o ambiente do runtime do Lambda.
- Chamar API da AWS: chama qualquer API somente leitura da AWS e exibe os resultados no formato JSON.
- Fast CloudWatch bitmap graph (Grafo de bitmap rápido do CloudWatch): renderize gráficos do CloudWatch usando no lado do servidor, para exibição rápida.
- Send dashboard snapshot by email (Enviar snapshot do painel por e-mail): faça um snapshot do painel atual e envie-o aos destinatários do email.
- Send dashboard snapshot to Amazon S3 (Enviar snapshot do paiel ao Amazon S3): faça um snapshot do painel atual e armazene-o no Amazon S3.
- Text widget from CloudWatch dashboard (Widget de texto do painel do CloudWatch): exibe o primeiro widget de texto do painel do CloudWatch especificado.
- Display content of external URL (Exibir conteúdo de URL externa): exibe o conteúdo de uma URL acessível externamente.
- RSS reader: exibe feeds RSS.
- Display an Amazon S3 object (Exibir um objeto do Amazon S3): exibe um objeto em um bucket do Amazon S3 em sua conta.

- Simple SVG pie chart (Gráfico de pizza SVG simples) exemplo de um widget gráfico baseado em SVG.

Widget Echo em JavaScript

A seguir está o widget de exemplo Echo em JavaScript.

```
const DOCS = `
## Echo
A basic echo script. Anything passed in the \`\`\`echo\`\`\` parameter is returned as
 the content of the custom widget.
### Widget parameters
Param | Description
---|---
**echo** | The content to echo back

### Example parameters
\`\`\` yaml
echo: <h1>Hello world</h1>
\`\`\`
`;

exports.handler = async (event) => {
    if (event.describe) {
        return DOCS;
    }

    let widgetContext = JSON.stringify(event.widgetContext, null, 4);
    widgetContext = widgetContext.replace(/</g, '&lt;');
    widgetContext = widgetContext.replace(/>/g, '&gt;');

    return `${event.echo || ''}<pre>${widgetContext}</pre>`;
};
```

Widget Echo em Python

A seguir está o widget de exemplo Echo em Python.

```
import json

DOCS = """
## Echo
```

```
A basic echo script. Anything passed in the ```echo``` parameter is returned as the
 content of the custom widget.
### Widget parameters
Param | Description
---|---
**echo** | The content to echo back

### Example parameters
``` yaml
echo: <h1>Hello world</h1>
```"""

def lambda_handler(event, context):
    if 'describe' in event:
        return DOCS

    echo = event.get('echo', '')
    widgetContext = event.get('widgetContext')
    widgetContext = json.dumps(widgetContext, indent=4)
    widgetContext = widgetContext.replace('<', '<')
    widgetContext = widgetContext.replace('>', '>')

    return f'{echo}<pre>{widgetContext}</pre>'
```

Widget Echo em Java

A seguir está o widget de exemplo Echo em Java.

```
package example;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

public class Handler implements RequestHandler<Event, String>{

  static String DOCS = ""
    + "## Echo\n"
    + "A basic echo script. Anything passed in the ```echo``` parameter is returned as
 the content of the custom widget.\n"
    + "### Widget parameters\n"
    + "Param | Description\n"
```

```
    + "---|---\n"
    + "**echo** | The content to echo back\n\n"
    + "### Example parameters\n"
    + "```yaml\n"
    + "echo: <h1>Hello world</h1>\n"
    + "```\n";

  Gson gson = new GsonBuilder().setPrettyPrinting().create();

  @Override
  public String handleRequest(Event event, Context context) {

    if (event.describe) {
      return DOCS;
    }

    return (event.echo != null ? event.echo : "") + "<pre>" +
 gson.toJson(event.widgetContext) + "</pre>";
  }
}

class Event {

    public boolean describe;
    public String echo;
    public Object widgetContext;

    public Event() {}

    public Event(String echo, boolean describe, Object widgetContext) {
        this.describe = describe;
        this.echo = echo;
        this.widgetContext = widgetContext;
    }
}
```

Adicionar ou remover um widget de texto de um painel do CloudWatch

Um widget de texto contém um bloco de texto no formato Markdown. Você pode adicionar, editar ou remover widgets de texto no painel do CloudWatch.

Para adicionar um widget de texto a um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Escolha o símbolo +.
4. Escolha Texto.
5. Em Markdown, adicione e formate o texto usando Markdown e escolha Create widget (Criar widget).
6. Para tornar o widget de texto transparente, escolha Fundo transparente.
7. Escolha Save dashboard (Salvar painel).

Para editar um widget de texto em um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Passe o mouse sobre o canto superior direito do bloco de texto e escolha Widget actions (Ações de widget). Em seguida, escolha Edit (Editar).
4. Atualize o texto conforme necessário e, em seguida, escolha Update widget (Atualizar widget).
5. Escolha Save dashboard (Salvar painel).

Para remover um widget de texto de um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Passe o mouse sobre o canto superior direito do bloco de texto e escolha Widget actions (Ações de widget). Em seguida, selecione Excluir.
4. Escolha Save dashboard (Salvar painel).

Adicionar ou remover um widget de alarme em um painel do CloudWatch

Para adicionar um widget de alarme a um painel, escolha uma das seguintes opções:

- Adicione um único alarme em um widget que exibe tanto o gráfico da métrica do alarme quanto o status do alarme.

- Você pode adicionar um widget de status de alarme para exibir o status de vários alarmes em uma grade. Somente os nomes dos alarmes e seus status atuais são exibidos. Os gráficos não são mostrados. Até 100 alarmes podem ser incluídos em um widget de status de alarme.

Para adicionar um único alarme, incluindo seu gráfico, a um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), selecione o alarme a ser adicionado e escolha Add to Dashboard (Adicionar ao painel).
3. Selecione um painel, escolha um tipo de widget (Linha, Área empilhada ou Número) e, em seguida, escolha Adicionar ao painel.
4. Para ver o alarme no painel, escolha Dashboards (Painéis) no painel de navegação e selecione o painel.
5. (Opcional) Para aumentar temporariamente o gráfico de alarme, selecione o gráfico.
6. (Opcional) Para alterar o tipo de widget, passe o mouse sobre o título do gráfico, escolha Ações de widget e, em seguida, escolha Tipo de widget.

Para adicionar um widget de status de alarme a um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Escolha o símbolo +.
4. Escolha Alarm status (Status do alarme).
5. Marque as caixas de seleção ao lado dos alarmes que você deseja adicionar ao widget e escolha Create widget (Criar widget).
6. Escolha Add to dashboard (Adicionar ao painel).

Para remover um widget de alarme de um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Passe o mouse sobre o widget, escolha Ações de widget e escolha Excluir.
4. Escolha Save dashboard (Salvar painel). Se você tentar sair do painel antes de salvar as alterações, será solicitado a salvar ou descartar as suas alterações.

Adicionar ou remover um widget de tabela de dados de um painel do CloudWatch

Com o widget de tabela de dados, é possível visualizar os pontos de dados brutos da sua métrica e um breve resumo desses dados brutos. Como o widget da tabela de dados não é um gráfico para realizar a abstração dos dados reais, é mais fácil compreender os pontos de dados apresentados. Os procedimentos nesta seção descrevem como adicionar e remover um widget de tabela de dados de um painel do CloudWatch.

Propriedades da tabela

Uma tabela de dados tem um conjunto padrão de propriedades que não requerem alterações nas opções ou na origem. Essas propriedades incluem uma coluna de rótulo fixa, todas as colunas de resumo habilitadas, os pontos de dados arredondados e as unidades convertidas.

Cada widget de tabela de dados pode ter as propriedades apresentadas a seguir. As informações sobre cada propriedade incluem como configurá-la na origem em JSON do painel. Para obter mais informações sobre a origem em JSON do painel, consulte Dashboard Body Structure and Syntax.

Resumo

As colunas de resumo são uma nova propriedade introduzida com o widget de tabela de dados. Essas colunas correspondem a um subconjunto específico de resumos da sua tabela atual. Por exemplo, o resumo Sum é a soma de todos os pontos de dados exibidos na linha. As colunas de resumo não são semelhantes às estatísticas do CloudWatch. A representação na origem é:

```
"table": {
       "summaryColumns": [
```

```
            "MIN",
            "MAX",
            "SUM",
            "AVG"
        ]
    },
```

Limites

Use isso para aplicar limites à sua tabela. Quando um ponto de dados está dentro de um limite, sua célula é destacada com a cor do limite. A representação na origem é:

```
"annotations": {
    "horizontal": [
        {
            "label": string,
            "value": int,
            "fill": "above" | "below"
        }
    ]
}
```

Unidade na coluna de rótulo

Para exibir qual unidade está associada à métrica, é possível habilitar essa opção para exibir a unidade na coluna de rótulo ao lado do rótulo. A representação na origem é:

```
"yAxis": {
    "left": {
        "showUnits": true | false
    }
}
```

Inverter linhas e colunas

Isso transforma a tabela para que os pontos de dados realizem a conversão de colunas para linhas e as métricas se tornem colunas. A representação na origem é:

```
 "table": {
    "layout": "vertical" | "horizontal"
}
```

Colunas de resumo fixas

Isso torna as colunas de resumo fixas para que permaneçam visíveis enquanto você realiza a rolagem. O rótulo já está fixo. A representação na origem é:

```
"table": {
    "stickySummary": true | false
}
```

Exibir somente colunas de resumo

Isso evita que as colunas de pontos de dados sejam exibidas, de forma que somente as colunas de rótulo e de resumo sejam exibidas. A representação na origem é:

```
 "table": {
    "showTimeSeriesData": false | true
}
```

Dados dinâmicos

Exibe o ponto de dados mais recente, mesmo que ele ainda não esteja totalmente agregado. A representação na origem é:

```
"liveData": true | false
```

Formato de número do widget

Exibe quantos dígitos cabem na célula antes de realizar o arredondamento ou a conversão. A representação na origem é:

```
"singleValueFullPrecision": true | false
```

Como adicionar um widget de tabela de dados a um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis e, em seguida, selecione um painel.
3. Escolha o botão +, selecione Tabela de dados e escolha Próximo.
4. Na guia Procurar, pesquise ou procure as métricas que deseja exibir no widget de tabela. Em seguida, selecione as métricas.

5. (Opcional) Para alterar o layout da tabela, escolha a guia Opções e selecione Inverter linhas e colunas.

   Também é possível usar a guia Opções para alterar quais colunas aparecem na tabela e exibir a unidade que está sendo usada na coluna Rótulo.

   > **Tip**
   >
   > Para exibir limites mais precisos, escolha Mostrar quantos dígitos cabem antes do arredondamento.

6. (Opcional) Para alterar o intervalo de tempo do widget da tabela de dados, selecione um dos intervalos de tempo definidos previamente na área superior do widget. Os intervalos de tempo variam de uma hora a uma semana. Para definir seu próprio intervalo de tempo, escolha Custom (Personalizado).
7. (Opcional) Para alterar o intervalo de tempo do widget da tabela de dados, selecione um dos intervalos de tempo definidos previamente na área superior do widget. Os intervalos de tempo variam de uma hora a uma semana. Para definir seu próprio intervalo de tempo, escolha Custom (Personalizado).
8. (Opcional) Para que esse widget continue usando o intervalo de tempo selecionado, mesmo que o intervalo de tempo do restante do painel seja alterado posteriormente, escolha Intervalo de tempo persistente.
9. Escolha Criar widget e, em seguida, Salvar painel.

Como remover um widget de tabela de um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. No canto superior direito do widget que você deseja remover, escolha Ações de widget e Excluir.
4. Escolha Save dashboard (Salvar painel).

Vincular e desvincular grafos em um painel do CloudWatch

Você pode vincular os gráficos em seu painel, de forma que, ao ampliar ou reduzir um gráfico, o mesmo ocorra com os outros gráficos simultaneamente. Você pode desvincular gráficos para limitar o zoom a um gráfico.

Para vincular os gráficos em um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Escolha Actions (Ações) e Link graphs (Vincular gráficos).

Para desvincular os gráficos em um painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Desmarque Actions (Ações) e Link graphs (Vincular gráficos).

Compartilhar painéis do CloudWatch

É possível compartilhar seus painéis do CloudWatch com pessoas que não têm acesso à conta da AWS. Isso permite que compartilhar painéis entre equipes, com partes interessadas e com pessoas externas a sua organização. É possível até exibir painéis em telas grandes em áreas de equipe ou incorporá-los a Wikis e outras páginas da Web.

> **Warning**
>
> Todas as pessoas com quem você compartilhar o painel receberão as permissões listadas em Permissões concedidas a pessoas com quem você compartilha o painel para a conta. Se você compartilhar o painel publicamente, todos os que tiverem o link para o painel terão essas permissões.
> As permissões `cloudwatch:GetMetricData` e `ec2:DescribeTags` não podem ter escopo para métricas específicas ou instâncias do EC2. Portanto, as pessoas com acesso ao painel podem consultar todas as métricas do CloudWatch e os nomes e etiquetas de todas as instâncias do EC2 na conta.

Ao compartilhar painéis, você pode designar quem pode exibir o painel de três formas:

- Compartilhe um só painel e atribua até cinco endereços de e-mail específicos de pessoas que poderão visualizar o painel. Cada um desses usuários cria sua própria senha, que devem inserir para exibir o painel.

- Compartilhe um único painel publicamente, de modo que qualquer pessoa que tenha o link possa visualizar o painel.
- Compartilhe todos os painéis do CloudWatch de sua conta e especifique um provedor de autenticação única (SSO) de terceiros para acesso ao painel. Todos os usuários que são membros da lista desse provedor de SSO podem acessar todos os painéis da conta. Para habilitar isso, integre o provedor de SSO ao Amazon Cognito. O provedor de SSO deve oferecer suporte a Security Assertion Markup Language (SAML). Para obter mais informações sobre o Amazon Cognito, consulte O que é o Amazon Cognito?

O compartilhamento de um painel não gera cobranças, mas os widgets dentro de um painel compartilhado incorrem em cobranças de acordo com as taxas padrão do CloudWatch. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

Quando você compartilha um painel, os recursos do Amazon Cognito são criados na região Leste dos EUA (Norte da Virgínia).

Important

Não modifique nomes e identificadores de recursos criados pelo processo de compartilhamento do painel. Isso inclui recursos do Amazon Cognito e do IAM. A modificação desses recursos pode causar uma funcionalidade inesperada e incorreta dos painéis compartilhados.

Note

Se você compartilhar um painel contendo widgets métricos com anotações de alarme, as pessoas com as quais você compartilhar o painel não verão esses widgets. Em vez disso, elas verão um widget em branco com uma mensagem informando que o widget não está disponível. Você ainda verá widgets métricos com anotações de alarme quando você mesmo visualizar o painel.

Permissões necessárias para compartilhar um painel

Para poder compartilhar painéis usando qualquer um dos métodos a seguir e ver quais painéis já foram compartilhados, é necessário fazer login como um usuário ou com um perfil do IAM que tenha determinadas permissões.

Para poder compartilhar painéis, seu usuário ou perfil do IAM deve incluir as permissões contidas na instrução de política a seguir:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/service-role/CWDBSharing*",
        "arn:aws:iam::*:policy/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cognito-idp:*",
        "cognito-identity:*",
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:GetDashboard",
    ],
    "Resource": [
        "*"
        // or the ARNs of dashboards that you want to share
    ]
}
```

Para ser capaz de ver quais painéis são compartilhados, mas não de compartilhar painéis, um usuário ou perfil do IAM pode incluir uma instrução de política semelhante à seguinte:

```
{
    "Effect": "Allow",
    "Action": [
        "cognito-idp:*",
        "cognito-identity:*"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:ListDashboards",
    ],
    "Resource": [
        "*"
    ]
}
```

Permissões concedidas a pessoas com quem você compartilha o painel

Ao compartilhar um painel, o CloudWatch cria uma função do IAM na conta, que dá as seguintes permissões às pessoas com quem você compartilha o painel:

- `cloudwatch:GetInsightRuleReport`
- `cloudwatch:GetMetricData`
- `cloudwatch:DescribeAlarms`
- `ec2:DescribeTags`

> **Warning**
>
> Todas as pessoas com quem você compartilhar o painel receberão essas permissões listadas para a conta. Se você compartilhar o painel publicamente, todos os que tiverem o link para o painel terão essas permissões.

As permissões `cloudwatch:GetMetricData` e `ec2:DescribeTags` não podem ter escopo para métricas específicas ou instâncias do EC2. Portanto, as pessoas com acesso ao painel podem consultar todas as métricas do CloudWatch e os nomes e etiquetas de todas as instâncias do EC2 na conta.

Quando você compartilha um painel, por padrão, as permissões criadas pelo CloudWatch restringem o acesso somente aos alarmes e às regras do Contributor Insights que estão no painel quando ele é compartilhado. Se você adicionar novos alarmes ou regras do Contributor Insights ao painel e quiser que eles também sejam vistos pelas pessoas com quem compartilhou o painel, atualize a política para permitir esses recursos.

Compartilhar um único painel com usuários específicos

Siga as etapas nesta seção para compartilhar um painel com até cinco endereços de e-mail que você escolher.

Note

Por padrão, todos os widgets do CloudWatch Logs no painel não são visíveis para as pessoas com quem você compartilha o painel. Para ter mais informações, consulte Permitir que as pessoas com quem você compartilha vejam widgets de tabelas de logs.
Por padrão, todos os widgets de alarmes compostos no painel não são visíveis para as pessoas com quem você compartilha o painel. Para ter mais informações, consulte Permitir que as pessoas com quem você compartilha vejam alarmes compostos.

Para compartilhar um painel com usuários específicos

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha o nome de seu painel.
4. Escolha Actions (Ações), Share dashboard (Compartilhar painel).
5. Ao lado de Share your dashboard and require a username and password (Compartilhar painel e solicitar um nome de usuário e senha), escolha Start sharing (Começar a compartilhar).
6. Em Add email addresses (Adicionar endereços de e-mail), insira os endereços de e-mail com os quais você deseja compartilhar o painel. É possível incluir até cinco endereços de e-mail.

7. Quando todos os endereços de e-mail forem inseridos, leia o contrato e marque a caixa de confirmação. Em seguida, escolha Preview policy (Pré-visualizar política).
8. Confirme se os recursos que serão compartilhados são os que você deseja e escolha Confirm and generate shareable link (Confirmar e gerar link compartilhável).
9. Na página seguinte, escolha Copy link to clipboard (Copiar link para área de transferência). Em seguida, você pode colar este link no e-mail e enviá-lo aos usuários convidados. Eles receberão automaticamente um e-mail separado com seu nome de usuário e uma senha temporária para usar para se conectar ao painel.

Compartilhar um único painel publicamente

Siga as etapas desta seção para compartilhar um painel publicamente. Isso pode ser útil para exibir o painel em uma tela grande em uma sala de equipe ou incorporá-lo a uma página Wiki.

> **Important**
>
> Compartilhar um painel publicamente o torna acessível a qualquer pessoa que tenha o link, sem autenticação. Faça isso somente para painéis que não contenham informações confidenciais.

> **Note**
>
> Por padrão, todos os widgets do CloudWatch Logs no painel não são visíveis para as pessoas com quem você compartilha o painel. Para ter mais informações, consulte Permitir que as pessoas com quem você compartilha vejam widgets de tabelas de logs.
> Por padrão, todos os widgets de alarmes compostos no painel não são visíveis para as pessoas com quem você compartilha o painel. Para ter mais informações, consulte Permitir que as pessoas com quem você compartilha vejam alarmes compostos.

Para compartilhar um painel publicamente

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha o nome de seu painel.

4. Escolha Actions (Ações), Share dashboard (Compartilhar painel).
5. Ao lado de Share your dashboard publicly (Compartilhar painel publicamente), escolha Start sharing (Comece a compartilhar).
6. Digite **`Confirm`** na caixa de texto.
7. Leia o contrato e marque a caixa de confirmação. Em seguida, escolha Preview policy (Pré-visualizar política).
8. Confirme se os recursos que serão compartilhados são os que você deseja e escolha Confirm and generate shareable link (Confirmar e gerar link compartilhável).
9. Na página seguinte, escolha Copy link to clipboard (Copiar link para área de transferência). Então você pode compartilhar o link. Qualquer pessoa com quem você compartilhar o link poderá acessar o painel, sem fornecer credenciais.

Compartilhe todos os painéis do CloudWatch da conta usando SSO

Use as etapas desta seção para compartilhar todos os painéis de sua conta com usuários usando autenticação única (SSO).

> **Note**
>
> Por padrão, todos os widgets do CloudWatch Logs no painel não são visíveis para as pessoas com quem você compartilha o painel. Para ter mais informações, consulte Permitir que as pessoas com quem você compartilha vejam widgets de tabelas de logs.
> Por padrão, todos os widgets de alarmes compostos no painel não são visíveis para as pessoas com quem você compartilha o painel. Para ter mais informações, consulte Permitir que as pessoas com quem você compartilha vejam alarmes compostos.

Para compartilhar seus painéis do CloudWatch com usuários que estão na lista de um provedor de SSO

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha o nome de seu painel.
4. Escolha Actions (Ações), Share dashboard (Compartilhar painel).
5. Escolha Go to CloudWatch Settings (Acessar configurações do CloudWatch).

6. Se o provedor de SSO que você deseja não estiver listado em Available SSO providers (Provedores de SSO disponíveis), escolha Manage SSO providers (Gerenciar provedores de SSO) e siga as instruções em Configurar SSO para compartilhar o painel do CloudWatch.

   Então, retorne ao console do CloudWatch e atualize o navegador. O provedor de SSO que você habilitou deverá aparecer na lista.
7. Escolha o provedor de SSO desejado na lista Available SSO providers (Provedores de SSO disponíveis).
8. Escolha Salvar alterações.

Configurar SSO para compartilhar o painel do CloudWatch

Para configurar o compartilhamento do painel por meio de um provedor de autenticação única de terceiros que ofereça suporte a SAML, siga estas etapas.

> **Important**
>
> É altamente recomendável não compartilhar painéis usando um provedor de SSO não SAML. Fazer isso causa o risco de permitir acidentalmente que terceiros acessem os painéis de sua conta.

Para configurar um provedor de SSO para habilitar o compartilhamento de painel

1. Integre o provedor de SSO ao Amazon Cognito. Para mais informações, consulte Integrar provedores de identidade SAML de terceiros com grupos de usuários do Amazon Cognito
2. Baixe o arquivo XML de metadados do provedor de SSO.
3. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
4. No painel de navegação, selecione Configurações.
5. Na seção Dashboard sharing (Compartilhar painel), escolha Configure (Configurar).
6. Selecione Manage SSO providers (Gerenciar provedores de SSO).

   Isso abrirá o console do Amazon Cognito na região Leste dos EUA (Norte da Virgínia) (us-east-1). Se você não visualizar User Pools (Grupos de usuários), o console do Amazon Cognito pode estar aberto em uma região diferente. Se for o caso, altere a região para Leste dos EUA (Norte da Virgínia) us-east-1 e prossiga para as próximas etapas.

7. Selecione o grupo CloudWatchDashboardSharing.
8. No painel de navegação, escolha Identity providers (Provedores de identidade).
9. Escolha SAML.
10. Insira um nome para o provedor de SSO em Provider name (Nome do provedor).
11. Escolha Select file (Selecionar arquivo) e selecione o arquivo XML de metadados baixado na etapa 1.
12. Escolha Create provider (Criar provedor).

Ver quantos de seus painéis são compartilhados

É possível usar o console do CloudWatch para ver quantos de seus painéis do CloudWatch estão sendo compartilhados com outras pessoas no momento.

Para ver quantos de seus painéis estão sendo compartilhados

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Configurações.
3. A seção Dashboard sharing (Compartilhamento de painéis) exibe quantos painéis estão sendo compartilhados.
4. Para ver quais painéis estão sendo compartilhados, escolha ***number*** dashboards shared (número de painéis compartilhados) em Username and password (Nome de usuário e senha) e em Public dashboards (Painéis públicos).

Ver quais painéis estão sendo compartilhados

É possível usar o console do CloudWatch para ver quais de seus painéis estão sendo compartilhados com outras pessoas no momento.

Para ver quais painéis estão sendo compartilhados

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Na lista de painéis, veja a coluna Share (Compartilhar). Os painéis que têm o ícone preenchido nesta coluna estão sendo compartilhados no momento.

4. Para ver com quais usuários um painel está sendo compartilhado, escolha o nome do painel e escolha Actions (Ações), Share dashboard (Compartilhar painel).

   A página Share dashboard ***dashboard name*** (Compartihar nome do painel) exibe como o painel está sendo compartilhado. Se desejar, você pode interromper o compartilhamento do painel escolhendo Stop sharing (Interromper o compartilhamento).

Interromper o compartilhamento de um ou mais painéis

Você pode interromper o compartilhamento de um único painel compartilhado ou de todos os painéis compartilhados de uma só vez.

Para interromper o compartilhamento de um único painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha o nome do painel compartilhado.
4. Escolha Actions (Ações), Share dashboard (Compartilhar painel).
5. Escolha Stop sharing (Interromper o compartilhamento).
6. Na caixa de confirmação, escolha Stop sharing (Interromper compartilhamento).

Para interromper o compartilhamento de todos os painéis compartilhados

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Configurações.
3. Na seção Dashboard sharing (Compartilhamento de painéis), escolha Stop sharing all dashboards (Interromper o compartilhamento de todos os painéis).
4. Na caixa de diálogo de confirmação, selecione Stop sharing all dashboards (Interromper o compartilhamento de todos os painéis).

Revisar permissões de painel compartilhadas e alterar o escopo de permissão

Use as etapas nesta seção para revisar as permissões dos usuários de seus painéis compartilhados ou alterar o escopo das permissões de paineis compartilhados.

Para revisar permissões de painéis compartilhados

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha o nome do painel compartilhado.
4. Escolha Actions (Ações), Share dashboard (Compartilhar painel).
5. Em Resources (Recursos), escolha IAM Role (Função do IAM).
6. No console do IAM, escolha a política exibida.
7. (Opcional) Para limitar os alarmes que os usuários do painel compartilhado podem ver, escolha Edit policy (Editar política) e mova a permissão `cloudwatch:DescribeAlarms` de sua posição atual para uma nova instrução `Allow` que lista os ARNs apenas dos alarmes que você deseja que sejam vistos pelos usuários do painel compartilhado. Veja o exemplo a seguir.

```
{
   "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": [
        "AlarmARN1",
        "AlarmARN2"
    ]
}
```

   Nesse caso, certifique-se de remover a permissão `cloudwatch:DescribeAlarms` de uma seção da política atual que se parece com esta:

```
{
   "Effect": "Allow",
    "Action": [
        "cloudwatch:GetInsightRuleReport",
        "cloudwatch:GetMetricData",
        "cloudwatch:DescribeAlarms",
        "ec2:DescribeTags"
    ],
    "Resource": "*"
}
```

8. (Opcional) Para limitar o escopo das regras do Contributor Insights que os usuários do painel compartilhado podem ver, escolha Edit policy (Editar política) e mova `cloudwatch:GetInsightRuleReport` de sua posição atual para uma nova instrução `Allow`

que lista os ARNs apenas das regras do Contributor Insights que você deseja que sejam vistos pelos usuários do painel compartilhado. Veja o exemplo a seguir.

```
{
   "Effect": "Allow",
    "Action": "cloudwatch:GetInsightRuleReport",
    "Resource": [
        "PublicContributorInsightsRuleARN1",
        "PublicContributorInsightsRuleARN2"
    ]
}
```

Nesse caso, certifique-se de remover `cloudwatch:GetInsightRuleReport` de uma seção da política atual que se parece com esta:

```
{
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetInsightRuleReport",
                "cloudwatch:GetMetricData",
                "cloudwatch:DescribeAlarms",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        }
```

Permitir que as pessoas com quem você compartilha vejam alarmes compostos

Por padrão, quando você compartilha um painel, os widgets de alarmes compostos no painel não são visíveis para as pessoas com quem você compartilha o painel. Para que widgets de alarmes compostos fiquem visíveis, é necessário adicionar uma permissão `DescribeAlarms: *` à política de compartilhamento do painel. Essa permissão deve ser semelhante a esta:

```
{
    "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": "*"
}
```

Warning

A instrução de política anterior dá acesso a todos os alarmes da conta. Para reduzir o escopo de `cloudwatch:DescribeAlarms`, é necessário usar uma instrução `Deny`. É possível adicionar uma instrução `Deny` à política e especificar os ARNs dos alarmes que você deseja bloquear. Essa instrução de negação deve ser semelhante a esta:

```
{
    "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": "*"
},
{
    "Effect": "Deny",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": [
        "SensitiveAlarm1ARN",
        "SensitiveAlarm1ARN"
    ]
}
```

Permitir que as pessoas com quem você compartilha vejam widgets de tabelas de logs

Por padrão, quando você compartilha um painel, os widgets do CloudWatch Logs Insights que estão no painel não são visíveis para as pessoas com quem você compartilha o painel. Isso afeta os widgets do CloudWatch Logs Insights que existem agora e os que são adicionados ao painel após o compartilhamento.

Para que essas pessoas possam ver widgets do CloudWatch Logs, é necessário adicionar permissões à função do IAM para compartilhamento de painel.

Para permitir que as pessoas com quem você compartilha um painel vejam os widgets do CloudWatch Logs

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha o nome do painel compartilhado.

4. Escolha Actions (Ações), Share dashboard (Compartilhar painel).
5. Em Resources (Recursos), escolha IAM Role (Função do IAM).
6. No console do IAM, escolha a política exibida.
7. Selecione Edit policy (Editar política) e adicione a instrução a seguir. Na nova instrução, recomendamos especificar os ARNs apenas dos grupos de log que você deseja compartilhar. Veja o exemplo a seguir.

```
{
            "Effect": "Allow",
            "Action": [
                "logs:FilterLogEvents",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:GetLogRecord",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "SharedLogGroup1ARN",
                "SharedLogGroup2ARN"
           ]
        },
```

8. Escolha Save Changes (Salvar alterações).

Se sua política do IAM para compartilhamento de painel já incluir essas cinco permissões com * como o recurso, é altamente recomendável alterar a política e especificar somente os ARNs dos grupos de logs que você deseja compartilhar. Por exemplo, se a seção `Resource` para essas permissões foi a seguinte:

```
"Resource": "*"
```

Altere a política para especificar somente os ARNs dos grupos de logs que você deseja compartilhar, como no exemplo a seguir:

```
"Resource": [
  "SharedLogGroup1ARN",
  "SharedLogGroup2ARN"
]
```

Permitir que as pessoas com quem você compartilha vejam widgets personalizados

Por padrão, quando você compartilha um painel, os widgets personalizados que estão no painel não são visíveis para as pessoas com quem você compartilha o painel. Isso afeta os widgets personalizados que existem agora e os que são adicionados ao painel após o compartilhamento.

Para que essas pessoas possam ver widgets personalizados, é necessário adicionar permissões à função do IAM para compartilhamento de painel.

Para permitir que as pessoas com quem você compartilha um painel vejam os widgets personalizados

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha o nome do painel compartilhado.
4. Escolha Actions (Ações), Share dashboard (Compartilhar painel).
5. Em Resources (Recursos), escolha IAM Role (Função do IAM).
6. No console do IAM, escolha a política exibida.
7. Selecione Edit policy (Editar política) e adicione a instrução a seguir. Na nova instrução, recomendamos especificar os ARNs apenas das funções do Lambda que você deseja compartilhar. Veja o exemplo a seguir.

```
{
  "Sid": "Invoke",
  "Effect": "Allow",
  "Action": [
      "lambda:InvokeFunction"
  ],
  "Resource": [
    "LambdaFunction1ARN",
    "LambdaFunction2ARN"
  ]
 }
```

8. Escolha Save Changes (Salvar alterações).

Se sua política do IAM para compartilhamento de painel já incluir essa permissão com * como recurso, é altamente recomendável alterar a política e especificar apenas os ARNs das funções do Lambda que você deseja compartilhar. Por exemplo, se a seção `Resource` para essas permissões foi a seguinte:

```
"Resource": "*"
```

Altere a política para especificar somente os ARNs dos widgets personalizados que você deseja compartilhar, como no exemplo a seguir:

```
"Resource": [
  "LambdaFunction1ARN",
  "LambdaFunction2ARN"
]
```

Usar dados em tempo real

Você pode escolher se os widgets de métrica exibir~ao dados em tempo real. Os dados em tempo real são dados publicados no último minuto que não foram totalmente agregados.

- Se os dados em tempo real estiverem desativados, somente os pontos de dados com um período de agregação de pelo menos um minuto no passado serão exibidos. Por exemplo, ao usar períodos de 5 minutos, o ponto de dados para 12:35 seria agregado de 12:35 a 12:40 e exibido às 12:41.
- Se os dados em tempo real estiverem ativados, o ponto de dados mais recente será exibido assim que todos os dados forem publicados no intervalo de agregação correspondente. Cada vez que você atualiza a exibição, o ponto de dados mais recente pode ser alterado à medida que novos dados dentro desse período de agregação são publicados. Se você usar uma estatística cumulativa, como Sum (Soma) ou Sample Count (Contagem de Amostras), o uso de dados dinâmicos pode resultar em uma queda no final do gráfico.

Você pode optar por habilitar dados dinâmicos para um painel inteiro ou para widgets individuais no painel.

Como escolher se deseja usar dados dinâmicos em todo o seu painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Para ativar ou desativar permanentemente os dados em tempo real para todos os widgets do painel, faça o seguinte:

   a. Escolha Actions (Ações), Settings (Configurações), Bulk update live data (Atualizar dados dinâmicos em lote).

   b. Escolha Live Data on (Dados em tempo real ativados) ou Live Data off (Dados em tempo real desativados), e escolha Set (Definir).
4. Para substituir temporariamente as configurações de dados dinâmicos de cada widget, escolha Actions (Ações). Então, em Overrides (Substituições), ao lado de Live data (Dados em tempo real), proceda de uma destas formas:

   - Escolha On (Ativados) para ativar temporariamente os dados em tempo real para todos os widgets.
   - Escolha Off (Desativados) para desativar temporariamente os dados em tempo real para todos os widgets.
   - Escolha Do not override (Não substituir) para preservar a configuração de dados dinâmicos de cada widget.

Para escolher se deseja usar dados em tempo real em um único widget

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Selecione um widget e escolha Actions (Ações), Edit (Editar).
4. Escolha a guia Graph options (Opções de gráfico).
5. Marque ou desmarque a caixa de seleção em Tempo real (Live Data).

Visualizar um painel animado

É possível um painel animado que reproduz dados de métrica do CloudWatch capturados ao longo do tempo. Isso pode ajudar a ver tendências, fazer apresentações ou analisar problemas depois que eles ocorrem.

Os widgets animados no painel incluem: widgets de linha, widgets de área empilhada, widgets de números e widgets de explorador de métricas. Grafos de pizza, gráficos de barras, widgets de texto e widgets de logs são exibidos no painel, mas não são animados.

Para visualizar um painel animado

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha o nome para o painel.
4. Selecione Actions (Ações), Replay dashboard (Repetir painel).
5. (Opcional) Por padrão, ao iniciar a animação, ela aparece como uma janela deslizante. Se quiser que a animação apareça como uma animação ponto a ponto, escolha o ícone de lupa enquanto a animação estiver pausada e redefina o zoom.
6. Para iniciar a animação, escolha o botão Play. Também é possível escolher os botões para trás e para frente para mover para outros pontos no tempo.
7. (Opcional) Para alterar a janela de tempo da animação, escolha o calendário e selecione o período.
8. Para alterar a velocidade da animação, escolha Auto speed (Velocidade automática) e selecione a nova velocidade.
9. Ao concluir, escolha Exit animate (Sair da animação).

Adicionar um painel do CloudWatch a sua lista de favoritos

No console do CloudWatch, você pode adicionar painéis, alarmes e grupos de logs a uma lista de favoritos. A lista de favoritos pode ser acessada no painel de navegação. O procedimento a seguir descreve como adicionar um painel à lista de favoritos.

Para adicionar um painel à lista de favoritos

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Na lista de painéis, selecione o símbolo de asterisco ao lado do nome do painel que deseja marcar como favorito.

    - (Opcional) Também é possível marcar um painel como favorito selecionando um painel na lista e escolhendo o símbolo de asterisco próximo ao nome do painel.

4. Para acessar a lista de favoritos, escolha Favorites and recents (Favoritos e recentes) no painel de navegação. O menu tem duas colunas. Uma contém seus painéis, alarmes e grupos de logs favoritos, enquanto a outra contém os painéis, alarmes e grupos de logs que você visitou recentemente.

> **Tip**
>
> É possível marcar painéis como favoritos, bem como alarmes e grupos de logs, no menu Favorites and recents (Favoritos e recentes) no painel de navegação do console do CloudWatch. Na coluna Recently visited (Visitados recentemente), mova o ponteiro do mouse sobre o painel que deseja marcar como favorito e escolha o símbolo de asterisco próximo a ele.

Altere a configuração de substituição de período ou atualize o intervalo para o painel do CloudWatch

Você pode especificar como as configurações de período de gráficos adicionadas a este painel são retidas ou modificadas.

Quando um período automático ou um intervalo de tempo persistente é aplicado a um widget, o intervalo de tempo geral do gráfico pode afetar os períodos que você definiu.

- Se o intervalo de tempo for de um dia ou menos, as configurações de período não serão alteradas.
- Se o intervalo de tempo estiver entre um dia e três dias, os períodos definidos como abaixo de cinco minutos serão alterados para 5 minutos.
- Se o intervalo de tempo for superior a três dias, os períodos definidos para menos de uma hora serão alterados para uma hora.

As etapas apresentadas a seguir explicam como usar o console para alterar as opções de substituição de período. Também é possível alterá-las ao usar o campo `period0verride` na estrutura em JSON do painel. Para obter mais informações, consulte Dashboard Body Overall Structure.

Para alterar as opções de substituição de período

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Escolha Ações.
3. Em Period override (Substituição de período), selecione uma das seguintes opções:

   - Escolha Auto (Automático) para fazer com que o período das métricas em cada gráfico adapte-se automaticamente para o intervalo de tempo do painel.
   - Escolha Do not override (Não substituir) para garantir que a configuração de período de cada gráfico seja sempre obedecida.
   - Escolha uma das outras opções para fazer com que os gráficos adicionados ao painel sempre adaptem esse horário escolhido como sua configuração de período.

   A Period override (Substituição de período) sempre reverte para Auto (Automático) quando o painel é fechado ou o navegador é atualizado. Configurações diferentes para Period override (Substituição de período) não podem ser salvas.

Você pode alterar a frequência com que os dados no painel do CloudWatch são atualizados.

Para alterar o intervalo de atualização do painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. No menu Refresh options (Opções de atualização) (canto superior direito), escolha 10 Seconds (10 segundos), 1 Minute (1 minuto), 2 Minutes (2 minutos), 5 Minutes (5 minutos) ou 15 Minutes (15 minutos).

Alterar o formato do período ou do fuso horário de um painel do CloudWatch

Você pode alterar o intervalo de tempo para exibir os dados do painel em minutos, horas, dias ou semanas. Você também pode alterar o formato de hora para exibir os dados do painel no horário UTC ou local. A hora local é o fuso horário especificado no sistema operacional do computador.

> **Note**
>
> Se você criar um painel com gráficos que contenham 100 ou mais métricas de alta resolução, recomendamos definir o intervalo de tempo para não mais de 1 hora. Para ter mais informações, consulte Métricas de alta resolução.

> **Note**
>
> Se o intervalo de tempo de um painel for menor do que o período usado para um widget no painel, ocorrerá o seguinte:
>
> - O widget é modificado para exibir a quantidade de dados correspondente a um período completo para esse widget, mesmo que esse período seja maior do que o intervalo de tempo do painel. Isso garante que haja pelo menos um ponto de dados no gráfico.
> - A hora de início do período para esse ponto de dados é ajustada retroativamente para garantir que pelo menos um ponto de dados possa ser exibido.

New console

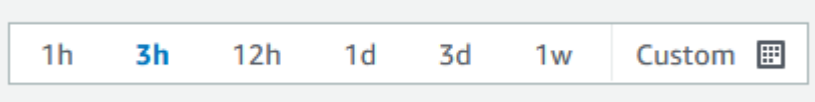
Para alterar o intervalo de tempo do painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Na tela do painel de controle, siga um destes procedimentos:
   - Na área superior do painel, selecione um dos intervalos de tempo predefinidos. Eles variam de 1 hora a 1 semana (1h, 3h, 12h, 1d ou 1w).
   - Como alternativa, você pode escolher uma das seguintes opções de intervalo de tempo personalizado:
     - Escolha Custom (Personalizado) e, em seguida, escolha a guia Relative (Relativo). Escolha um intervalo de tempo de 1 minuto a 15 meses.
     - Escolha Custom (Personalizado) e, em seguida, a guia Absolute (Absoluto). Use o calendário ou os campos de texto para especificar o intervalo de tempo.

> **Tip**
>
> Se o período de agregação estiver definido como Auto (Automático), quando você alterar o intervalo de tempo de um gráfico, o CloudWatch poderá alterar o período. Para definir o período manualmente, escolha a lista suspensa Actions (Ações) e, em seguida, escolha Period override (Substituição de período).

Para alterar o formato de fuso horário do painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Na área superior do painel, escolha Personalizado.



4. No canto superior direito da caixa que aparece, selecione o menu suspenso e escolha UTC ou Local time (Hora local).
5. Escolha Aplicar.

Old console

Para alterar o intervalo de tempo do painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. Na tela do painel de controle, siga um destes procedimentos:
   - Na área superior do painel, selecione um dos intervalos de tempo predefinidos. Eles variam de 1 hora a 1 semana (1h,3h,12h,1d,3d ou1w).
   - Como alternativa, você pode escolher uma das seguintes opções de intervalo de tempo personalizado:
     - Escolha a lista suspensa custom (personalizado) e, em seguida, escolha a guia Relative (Relativo). Selecione um dos intervalos predefinidos, que variam de 1 minuto a 15 meses.

- Escolha a lista suspensa custom (personalizado) e, em seguida, escolha a guia Absolute (Absoluto). Use o calendário ou os campos de texto para especificar o intervalo de tempo.

> **Tip**
>
> Se o período de agregação estiver definido como Auto (Automático), quando você alterar o intervalo de tempo de um gráfico, o CloudWatch poderá alterar o período. Para definir o período manualmente, escolha a lista suspensa Actions (Ações) e, em seguida, escolha Period override (Substituição de período).

Para alterar o formato de fuso horário do painel

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e escolha um painel.
3. No canto superior direito da tela do painel, escolha o menu suspenso Custom (Personalizado).
4. No canto superior direito da caixa que aparece, selecione UTC ou Local timezone (Fuso horário local) no menu suspenso.

Usar métricas do Amazon CloudWatch

Métricas são dados sobre a performance de seus sistemas. Por padrão, muitos serviços fornecem métricas gratuitas para recursos (como instâncias do Amazon EC2, volumes do Amazon EBS e instâncias de banco de dados do Amazon RDS). Você também pode habilitar o monitoramento detalhado de alguns recursos, como instâncias do Amazon EC2 ou publicar suas próprias métricas de aplicações. O Amazon CloudWatch pode carregar todas as métricas em sua conta (métricas de recursos da AWS e métricas de aplicações que você fornecer) para pesquisa, criação de gráficos e alarmes.

Os dados das métricas são mantidos por 15 meses, permitindo que você visualize os dados mais recentes e dados históricos.

Para gráficos de métricas no console, você pode usar o CloudWatch Metrics Insights, um mecanismo de consulta SQL de alta performance que pode ser usado para identificar tendências e padrões em todas as suas métricas em tempo real.

Conteúdo

Monitoramento básico e monitoramento detalhado

O CloudWatch fornece duas categorias de monitoramento: monitoramento básico e monitoramento detalhado.

Muitos serviços da AWS oferecem monitoramento básico ao publicar um conjunto padrão de métricas para o CloudWatch sem cobrança para os clientes. Quando você começa a usar um desses Serviços da AWS, o monitoramento básico é habilitado automaticamente por padrão. Para obter uma lista dos serviços que oferecem monitoramento básico, consulte Produtos da AWS que publicam métricas do CloudWatch.

O monitoramento detalhado é oferecido apenas por alguns serviços. Essa modalidade também gera cobranças. Para usá-lo em um serviço da AWS, você deve optar por ativá-lo. Para obter mais informações sobre a definição de preço, consulte Definição de preço do Amazon CloudWatch.

As opções do monitoramento detalhado diferem com base nos serviços que o oferecem. Por exemplo, o monitoramento detalhado do Amazon EC2 fornece métricas mais frequentes, publicadas em intervalos de 1 minuto, em vez dos intervalos de 5 minutos usados no monitoramento básico do Amazon EC2. O monitoramento detalhado para o Amazon S3 e o Amazon Managed Streaming for Apache Kafka significa métricas mais refinadas.

Em diferentes serviços da AWS, o monitoramento detalhado também tem nomes diferentes. Por exemplo, no Amazon EC2, ele é chamado de monitoramento detalhado, no AWS Elastic Beanstalk ele é chamado de monitoramento aprimorado. No Amazon S3, ele é chamado de métricas de solicitação.

O uso de monitoramento detalhado para o Amazon EC2 ajuda a gerenciar melhor seus recursos do Amazon EC2, permitindo que você encontre tendências e atue com mais rapidez. Para o Amazon S3, as métricas de solicitação estão disponíveis em intervalos de 1 minuto a fim de ajudar você a identificar e agir rapidamente em problemas operacionais. No Amazon MSK, ao habilitar o monitoramento de nível `PER_BROKER`, `PER_TOPIC_PER_BROKER` ou `PER_TOPIC_PER_PARTITION`, você obtém métricas adicionais que fornecem maior visibilidade.

A tabela a seguir lista os serviços que oferecem monitoramento detalhado. Ela também inclui links para a documentação dos respectivos serviços, explicando mais sobre o monitoramento detalhado e fornecendo instruções sobre como ativá-lo.

Serviço	Documentação
Amazon API Gateway	Dimensões para métricas do API Gateway

Serviço	Documentação
Amazon CloudFront	Visualizar métricas adicionais de distribuição do CloudFront
Amazon EC2	Habilitar ou desabilitar o monitoramento detalhado para suas instâncias
Elastic Beanstalk	Monitoramento e relatório de integridade aprimorados
Amazon Kinesis Data Streams	Métricas aprimoradas de fragmento
Amazon MSK	Métricas do Amazon MSK para monitorar com o CloudWatch
Amazon S3	Métricas de solicitação do Amazon S3 no CloudWatch

Serviço	Documentação
Amazon SES	Colete métricas de monitoramento detalhadas do CloudWatch usando a publicação de eventos do Amazon SES.

Além disso, o CloudWatch oferece soluções de monitoramento prontas para uso com métricas mais detalhadas e painéis pré-criados para alguns serviços da AWS, conforme mostrado na tabela a seguir.

Serviço	Documentação de recursos
Lambda	Lambda Insights
Amazon ECS	Container Insights para Amazon ECS
Amazon EKS	Container Insights para Amazon EKS e Kubernetes

Consulte suas métricas com o CloudWatch Metrics Insights

O CloudWatch Metrics Insights é um poderoso mecanismo de consulta SQL de alta performance que você pode usar para consultar suas métricas em escala. É possível identificar tendências e padrões em todas as métricas do CloudWatch em tempo real.

Você também pode definir alarmes em qualquer consulta ao Metrics Insights que retorne uma única série temporal. Isso pode ser especialmente útil para criar alarmes que monitorem métricas agregadas em toda uma frota da sua infraestrutura ou das suas aplicações. Crie o alarme uma vez e ele se ajustará dinamicamente à medida que recursos forem adicionados ou removidos da frota.

Você pode realizar uma consulta do CloudWatch Metrics Insights no console com o editor de consultas do CloudWatch Metrics Insights. Você também pode realizar uma consulta do CloudWatch Metrics Insights com a AWS CLI ou um SDK da AWS executando `GetMetricData` ou `PutDashboard`. Não há cobrança pelas consultas que você executa com o editor de consultas do CloudWatch Metrics Insights. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

Com o editor de consultas do CloudWatch Metrics Insights, você pode escolher entre uma variedade de consultas de amostra pré-criadas e também criar suas próprias consultas. Ao criar suas consultas, você pode usar uma visualização do construtor para pesquisar as métricas e dimensões atuais. Como alternativa, use uma visualização de editor para escrever consultas manualmente.

Você também pode usar linguagem natural para criar consultas do CloudWatch Metrics Insights. Para fazer isso, faça perguntas ou descreva os dados que você está procurando. Esse recurso assistido por IA gera uma consulta com base na sua solicitação e fornece uma explicação detalhada de como a consulta funciona. Para obter mais informações, consulte Use natural language to generate and update CloudWatch Metrics Insights queries.

Com o Metrics Insights, você pode executar consultas em escala. Com a cláusula GROUP BY, você pode agrupar as métricas em tempo real em séries temporais separadas por valor de dimensão específico. Como as consultas do Metrics Insights incluem a capacidade ORDER BY, você pode usar o Metrics Insights para fazer consultas do tipo “N principais”. Por exemplo, consultas do tipo “N principais” podem verificar milhões de métricas na sua conta e retornar as dez instâncias que consomem mais CPU. Isso pode ajudar você a identificar e corrigir problemas de latência nas suas aplicações.

Tópicos

- Inferência SQL
- Consultas de exemplo do Metrics Insights
- Limites do Metrics Insights
- Glossário do Metrics Insights
- Solução de problemas do métricas do Metrics Insights

Criar consultas

Você pode executar uma consulta do CloudWatch Metrics Insights usando o console do CloudWatch, a AWS CLI ou os AWS SDKs. As consultas executadas no console são oferecidas gratuitamente. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

Para obter mais informações sobre como usar os AWS SDKs para executar uma consulta do Metrics Insights, consulte GetMetricData.

Para executar uma consulta usando o console do CloudWatch, siga estas etapas:

Para consultar suas métricas usando o Metrics Insights

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), All metrics (Todas as métricas).
3. Escolha a guia Queries (Consultas).
4. (Opcional) Para executar uma consulta de amostra pré-criada, escolha Add query (Adicionar consulta) e selecione a consulta a ser executada. Se estiver contente com essa consulta, poderá ignorar o restante deste procedimento. Caso contrário, você pode escolher Editor para editar a consulta de exemplo e, em seguida, escolher Run (Executar) para executar a consulta modificada.
5. Para criar sua própria consulta, você pode usar as visualizações Builder (Criador), Editor e também uma combinação de ambos. Você pode alternar entre as duas visualizações a qualquer momento e ver o trabalho em andamento em ambas.

   No Builder (Criador), você pode procurar e selecionar o namespace da métrica, o nome da métrica, o filtro, o grupo e as opções de ordem. Para cada uma dessas opções, o criador de consultas oferece uma lista de possíveis seleções do seu ambiente para escolher.

Na visualização de Editor, você pode começar a gravar sua consulta. À medida que você digita, o editor oferece sugestões com base nos caracteres digitados até o momento.

6. Quando você estiver contente com sua consulta escolha Run (Executar).
7. (Opcional) Outra maneira de editar uma consulta que você tenha representada em gráfico é escolher a guia Graphed metrics (Métricas em gráficos) e escolher o ícone de edição ao lado da fórmula de consulta na coluna Details (Detalhes).
8. (Opcional) Para remover uma consulta do gráfico, escolha Graphed metrics (Métricas em gráficos) e escolha o ícone X no lado direito da linha que exibe a consulta.

Componentes e sintaxe de consulta do Metrics Insights

A sintaxe do CloudWatch Metrics Insights é a seguinte.

```
SELECT FUNCTION(metricName)
FROM namespace | SCHEMA(...)
[ WHERE labelKey OPERATOR labelValue [AND ... ] ]
[ GROUP BY labelKey [ , ... ] ]
[ ORDER BY FUNCTION() [ DESC | ASC ] ]
[ LIMIT number ]
```

As cláusulas possíveis em uma consulta do Metrics Insights são as seguintes. Nenhuma das palavras-chave diferencia maiúsculas de minúsculas, mas os identificadores como os nomes de métricas, namespaces e dimensões sim.

SELECT

Obrigatório. Especifica a função a ser usada para agregar observações em cada bucket de tempo (determinado pelo período fornecido). Também especifica o nome da métrica a ser consultada.

Os valores válidos para FUNCTION são `AVG`, `COUNT`, `MAX`, `MIN` e `SUM`.

- `AVG` calcula a média das observações correspondidas pela consulta.
- `COUNT` retorna a contagem das observações correspondidas pela consulta.
- `MAX` retorna o valor máximo das observações correspondidas pela consulta.
- `MIN` retorna o valor mínimo das observações correspondidas pela consulta.
- `SUM` calcula a soma das observações correspondidas pela consulta.

FROM

Obrigatório. Especifica a fonte da métrica. Você pode especificar o namespace da métrica que contém a métrica a ser consultada ou uma função de tabela SCHEMA. Exemplos de namespaces de métrica incluem `"AWS/EC2"`, `"AWS/Lambda"` e namespaces de métrica criados para suas métricas personalizadas.

Namespaces de métrica que incluem / ou qualquer outro caractere que não seja letra, número ou sublinhado devem ser cercados por aspas duplas. Para ter mais informações, consulte Quando é necessário usar aspas ou caracteres de escape?.

SCHEMA

Uma função de tabela opcional que pode ser usada dentro de uma cláusula FROM. Use SCHEMA para reduzir o escopo dos resultados da consulta apenas para as métricas que correspondem exatamente a uma lista de dimensões ou para métricas que não têm dimensões.

Se você usar uma cláusula SCHEMA, ela deve conter pelo menos um argumento, e o primeiro deles deve ser o namespace de métrica que está sendo consultado. Se você especificar SCHEMA só com o argumento de namespace, os resultados são reduzidos para somente as métricas que não têm dimensões.

Se você especificar SCHEMA com argumentos adicionais, os argumentos adicionais após o argumento de namespace devem ser chaves de rótulo. As chaves de rótulo devem ser nomes de dimensão. Se você especificar uma ou mais chaves de rótulo, os resultados serão definidos apenas para as métricas que têm esse conjunto exato de dimensões. A ordem das chaves de rótulo não tem importância.

Por exemplo:

- SELECT AVG(CPUUtilization) FROM "AWS/EC2" corresponde a todas as métricas `CPUUtilization` do namespace `AWS/EC2`, independentemente de suas dimensões, e retorna uma única série temporal agregada.
- SELECT AVG(CPUUtilization) FROM SCHEMA("AWS/EC2") corresponde apenas às métricas de `CPUUtilization` no namespace `AWS/EC2` que não têm nenhuma dimensão definida.
- SELECT AVG(CPUUtilization) FROM SCHEMA("AWS/EC2", InstanceId) corresponde apenas às métricas `CPUUtilization` que foram relatadas ao CloudWatch com exatamente uma dimensão, `InstanceId`.
- SELECT SUM(RequestCount) FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone) corresponde apenas às métricas `RequestCount` que foram relatadas ao

CloudWatch pelo `AWS/ApplicationELB` com exatamente duas dimensões, `LoadBalancer` e `AvailabilityZone`.

WHERE

Opcional. Filtra os resultados apenas para as métricas que correspondem à expressão especificada usando valores de rótulo específicos para uma ou mais chaves de rótulo. Por exemplo, WHERE InstanceType = 'c3.4xlarge' filtra os resultados somente para os tipos de instância `c3.4xlarge`, e WHERE InstanceType != 'c3.4xlarge' filtra os resultados para todos os tipos de instância, exceto `c3.4xlarge`.

Ao executar uma consulta em uma conta de monitoramento, você pode usar `WHERE AWS.AccountId` para limitar os resultados somente à conta que você especificar. Por exemplo, `WHERE AWS.AccountId=444455556666` só consulta métricas da conta `444455556666`. Para limitar sua consulta somente a métricas na própria conta de monitoramento, use `WHERE AWS.AccountId=CURRENT_ACCOUNT_ID()`.

Os valores do rótulo devem sempre ser incluídos com aspas simples.

Operadores compatíveis

A cláusula WHERE é compatível com os seguintes operadores:

- = O valor do rótulo deve corresponder à string especificada.
- ! = O valor do rótulo não deve ser correspondente à string especificada.
- AND Ambas as condições especificadas devem ser verdadeiras para serem correspondentes. Você pode usar várias palavras-chave AND para especificar duas ou mais condições.

GROUP BY

Opcional. Agrupa os resultados da consulta em várias séries temporais, cada uma correspondendo a um valor diferente para a chave de rótulo ou chaves especificadas. Por exemplo, o uso de `GROUP BY InstanceId` retorna uma série temporal diferente para cada valor de `InstanceId`. O uso do `GROUP BY ServiceName, Operation` cria uma série temporal diferente para cada combinação possível de valores de `ServiceName` e `Operation`.

Com uma cláusula GROUP BY, os resultados são ordenados alfabeticamente em ordem crescente por padrão usando a sequência de rótulos especificada na cláusula GROUP BY. Para alterar a ordem dos resultados, adicione uma cláusula ORDER BY à sua consulta.

Ao executar uma consulta em uma conta de monitoramento, você pode usar `GROUP BY AWS.AccountId` para agrupar os resultados com base nas contas das quais eles se originam.

> **Note**
>
> Se algumas das métricas correspondentes não especificarem uma chave de rótulo específica na cláusula GROUP BY, é retornado um grupo nulo denominado `Other`. Por exemplo, se você especificar `GROUP BY ServiceName, Operation` e algumas das métricas retornadas não incluem `ServiceName` como uma dimensão, essas métricas são exibidas com `Other` como o valor para `ServiceName`.

ORDER BY

Opcional. Especifica a ordem a ser usada para a série temporal retornada, se a consulta retornar mais de uma série temporal. A ordem é baseada nos valores encontrados pela FUNCTION que você especifica na cláusula ORDER BY. A FUNCTION é usada para calcular um único valor escalar de cada série temporal retornada, e esse valor é usado para determinar a ordem.

Você também especifica se deve-se usar a ordem crescente ASC ou decrescente DESC. Se você omitir isso, o padrão será crescente ASC.

Por exemplo, adicionar uma cláusula `ORDER BY MAX() DESC` ordena os resultados pelo ponto de dados máximo observado dentro do intervalo de tempo em ordem decrescente: o que significa que a série temporal que tem o ponto de dados máximo mais alto é retornada primeiro.

As funções válidas para serem usadas em uma cláusula ORDER BY são `AVG()`, `COUNT()`, `MAX()`, `MIN()` e `SUM()`.

Se você usar uma cláusula ORDER BY com uma cláusula LIMIT, a consulta resultante é uma consulta “Top N”. ORDER BY também é útil para consultas que podem retornar um grande número de métricas, porque cada consulta não pode retornar mais de 500 séries temporais. Se uma consulta corresponder a mais de 500 séries temporais e você usar uma cláusula ORDER BY, as séries temporais são classificadas e, em seguida, as 500 que aparecem primeiro na ordem de classificação são as que são retornadas.

LIMIT

Opcional. Limita o número de séries temporais retornadas pela consulta ao valor que você especificar. O valor máximo que você pode especificar é 500, e uma consulta que não especifica um LIMIT também pode retornar até 500 séries temporais.

O uso de uma cláusula LIMIT com uma cláusula ORDER BY fornece uma consulta “Top N”.

Quando é necessário usar aspas ou caracteres de escape?

Em uma consulta, os valores de rótulo devem sempre estar cercados por aspas simples. Por exemplo, SELECT MAX(CPUUtilization) FROM "AWS/EC2" WHERE AutoScalingGroupName = 'my-production-fleet'.

Namespaces de métricas, nomes de métricas e chaves de rótulo que contenham caracteres diferentes de letras, números e sublinhado (_) devem ser cercados por aspas duplas. Por exemplo, SELECT MAX("My.Metric").

Se um deles contiver aspas duplas ou aspas simples (como `Bytes"Input"`), você deve fazer o escape de cada aspa com uma barra invertida, como em SELECT AVG("Bytes\"Input\"").

Se um namespace de métrica, nome da métrica ou chave de rótulo contiver uma palavra que seja palavra-chave reservada no Metrics Insights, eles também deverão estar entre aspas duplas. Por exemplo, se você tiver uma métrica chamada `LIMIT`, você pode usar `SELECT AVG("LIMIT")`. Também é válido colocar qualquer namespace, nome da métrica ou rótulo entre aspas duplas, mesmo que não inclua uma palavra-chave reservada.

(Para obter uma lista completa de palavras-chave reservadas, consulte Palavras-chave reservadas.

Construir uma consulta avançada passo a passo

Esta seção ilustra a criação de um exemplo completo que usa todas as cláusulas possíveis, passo a passo.

Começamos com a seguinte consulta, que agrega todas as métricas `RequestCount` do Application Load Balancer coletadas com ambas as dimensões `LoadBalancer` e `AvailabilityZone`.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
```

Agora, se quisermos ver métricas somente de um balanceador de carga específico, podemos adicionar uma cláusula WHERE para limitar as métricas retornadas somente para aquelas em que o valor da dimensão `LoadBalancer` é `app/load-balancer-1`.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
```

A consulta anterior agrega as métricas `RequestCount` de todas as zonas de disponibilidade para esse balanceador de carga em uma série temporal. Se quisermos ver séries temporais diferentes para cada zona de disponibilidade, podemos adicionar uma cláusula GROUP BY.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
```

Em seguida, podemos querer ordenar esses resultados para ver os valores mais altos primeiro. A seguinte cláusula ORDER BY ordena a série temporal em ordem decrescente pelo valor máximo relatado por cada série temporal durante o intervalo de tempo da consulta:

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
ORDER BY MAX() DESC
```

Finalmente, se estivermos interessados principalmente em um tipo de consulta "Top N", podemos usar uma cláusula LIMIT. Este exemplo final limita os resultados apenas às séries temporais com os cinco maiores valores `MAX`.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
ORDER BY MAX() DESC
LIMIT 5
```

Exemplos de consultas entre contas

Esses exemplos são válidos quando executados em uma conta configurada como conta de monitoramento na observabilidade entre contas do CloudWatch.

O exemplo a seguir pesquisa todas as instâncias do Amazon EC2 na conta de origem 123456789012 e retorna a média.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
```

```
WHERE AWS.AccountId ='123456789012'
```

O exemplo a seguir consulta a métrica `CPUUtilization` no `AWS/EC2` em todas as contas de origem vinculadas e agrupa os resultados por ID da conta e tipo de instância.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
GROUP BY AWS.AccountId, InstanceType
```

O exemplo a seguir consulta a `CPUUtilization` na própria conta de monitoramento.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
WHERE AWS.AccountId = CURRENT_ACCOUNT_ID()
```

Palavras-chave reservadas

A seguir, veja as palavras-chave reservadas no CloudWatch Metrics Insights. Se alguma dessas palavras estiver em um namespace, nome da métrica ou chave de rótulo em uma consulta, você deverá colocá-las entre aspas duplas. As palavras-chave reservadas não diferenciam letras maiúsculas.

```
"ABORT" "ABORTSESSION" "ABS" "ABSOLUTE" "ACCESS" "ACCESSIBLE" "ACCESS_LOCK" "ACCOUNT"
 "ACOS" "ACOSH" "ACTION" "ADD" "ADD_MONTHS"
"ADMIN" "AFTER" "AGGREGATE" "ALIAS" "ALL" "ALLOCATE" "ALLOW" "ALTER" "ALTERAND" "AMP"
 "ANALYSE" "ANALYZE" "AND" "ANSIDATE" "ANY" "ARE" "ARRAY",
"ARRAY_AGG" "ARRAY_EXISTS" "ARRAY_MAX_CARDINALITY" "AS" "ASC" "ASENSITIVE" "ASIN"
 "ASINH" "ASSERTION" "ASSOCIATE" "ASUTIME" "ASYMMETRIC" "AT",
"ATAN" "ATAN2" "ATANH" "ATOMIC" "AUDIT" "AUTHORIZATION" "AUX" "AUXILIARY" "AVE"
 "AVERAGE" "AVG" "BACKUP" "BEFORE" "BEGIN" "BEGIN_FRAME" "BEGIN_PARTITION",
"BETWEEN" "BIGINT" "BINARY" "BIT" "BLOB" "BOOLEAN" "BOTH" "BREADTH" "BREAK" "BROWSE"
 "BT" "BUFFERPOOL" "BULK" "BUT" "BY" "BYTE" "BYTEINT" "BYTES" "CALL",
"CALLED" "CAPTURE" "CARDINALITY" "CASCADE" "CASCADED" "CASE" "CASESPECIFIC" "CASE_N"
 "CAST" "CATALOG" "CCSID" "CD" "CEIL" "CEILING" "CHANGE" "CHAR",
"CHAR2HEXINT" "CHARACTER" "CHARACTERS" "CHARACTER_LENGTH" "CHARS" "CHAR_LENGTH" "CHECK"
 "CHECKPOINT" "CLASS" "CLASSIFIER" "CLOB" "CLONE" "CLOSE" "CLUSTER",
"CLUSTERED" "CM" "COALESCE" "COLLATE" "COLLATION" "COLLECT" "COLLECTION" "COLLID"
 "COLUMN" "COLUMN_VALUE" "COMMENT" "COMMIT" "COMPLETION" "COMPRESS" "COMPUTE",
"CONCAT" "CONCURRENTLY" "CONDITION" "CONNECT" "CONNECTION" "CONSTRAINT" "CONSTRAINTS"
 "CONSTRUCTOR" "CONTAINS" "CONTAINSTABLE" "CONTENT" "CONTINUE" "CONVERT",
"CONVERT_TABLE_HEADER" "COPY" "CORR" "CORRESPONDING" "COS" "COSH" "COUNT" "COVAR_POP"
 "COVAR_SAMP" "CREATE" "CROSS" "CS" "CSUM" "CT" "CUBE" "CUME_DIST",
```

```
"CURRENT" "CURRENT_CATALOG" "CURRENT_DATE" "CURRENT_DEFAULT_TRANSFORM_GROUP"
 "CURRENT_LC_CTYPE" "CURRENT_PATH" "CURRENT_ROLE" "CURRENT_ROW" "CURRENT_SCHEMA",
"CURRENT_SERVER" "CURRENT_TIME" "CURRENT_TIMESTAMP" "CURRENT_TIMEZONE"
 "CURRENT_TRANSFORM_GROUP_FOR_TYPE" "CURRENT_USER" "CURRVAL" "CURSOR" "CV" "CYCLE"
 "DATA",
"DATABASE" "DATABASES" "DATABLOCKSIZE" "DATE" "DATEFORM" "DAY" "DAYS" "DAY_HOUR"
 "DAY_MICROSECOND" "DAY_MINUTE" "DAY_SECOND" "DBCC" "DBINFO" "DEALLOCATE" "DEC",
"DECFLOAT" "DECIMAL" "DECLARE" "DEFAULT" "DEFERRABLE" "DEFERRED" "DEFINE" "DEGREES"
 "DEL" "DELAYED" "DELETE" "DENSE_RANK" "DENY" "DEPTH" "DEREF" "DESC" "DESCRIBE",
"DESCRIPTOR" "DESTROY" "DESTRUCTOR" "DETERMINISTIC" "DIAGNOSTIC" "DIAGNOSTICS"
 "DICTIONARY" "DISABLE" "DISABLED" "DISALLOW" "DISCONNECT" "DISK" "DISTINCT",
"DISTINCTROW" "DISTRIBUTED" "DIV" "DO" "DOCUMENT" "DOMAIN" "DOUBLE" "DROP" "DSSIZE"
 "DUAL" "DUMP" "DYNAMIC" "EACH" "ECHO" "EDITPROC" "ELEMENT" "ELSE" "ELSEIF",
"EMPTY" "ENABLED" "ENCLOSED" "ENCODING" "ENCRYPTION" "END" "END-EXEC" "ENDING"
 "END_FRAME" "END_PARTITION" "EQ" "EQUALS" "ERASE" "ERRLVL" "ERROR" "ERRORFILES",
"ERRORTABLES" "ESCAPE" "ESCAPED" "ET" "EVERY" "EXCEPT" "EXCEPTION" "EXCLUSIVE" "EXEC"
 "EXECUTE" "EXISTS" "EXIT" "EXP" "EXPLAIN" "EXTERNAL" "EXTRACT" "FALLBACK
"FALSE" "FASTEXPORT" "FENCED" "FETCH" "FIELDPROC" "FILE" "FILLFACTOR" "FILTER" "FINAL"
 "FIRST" "FIRST_VALUE" "FLOAT" "FLOAT4" "FLOAT8" "FLOOR"
"FOR" "FORCE" "FOREIGN" "FORMAT" "FOUND" "FRAME_ROW" "FREE" "FREESPACE" "FREETEXT"
 "FREETEXTTABLE" "FREEZE" "FROM" "FULL" "FULLTEXT" "FUNCTION"
"FUSION" "GE" "GENERAL" "GENERATED" "GET" "GIVE" "GLOBAL" "GO" "GOTO" "GRANT" "GRAPHIC"
 "GROUP" "GROUPING" "GROUPS" "GT" "HANDLER" "HASH"
"HASHAMP" "HASHBAKAMP" "HASHBUCKET" "HASHROW" "HAVING" "HELP" "HIGH_PRIORITY" "HOLD"
 "HOLDLOCK" "HOUR" "HOURS" "HOUR_MICROSECOND" "HOUR_MINUTE"
"HOUR_SECOND" "IDENTIFIED" "IDENTITY" "IDENTITYCOL" "IDENTITY_INSERT" "IF" "IGNORE"
 "ILIKE" "IMMEDIATE" "IN" "INCLUSIVE" "INCONSISTENT" "INCREMENT"
"INDEX" "INDICATOR" "INFILE" "INHERIT" "INITIAL" "INITIALIZE" "INITIALLY" "INITIATE"
 "INNER" "INOUT" "INPUT" "INS" "INSENSITIVE" "INSERT" "INSTEAD"
"INT" "INT1" "INT2" "INT3" "INT4" "INT8" "INTEGER" "INTEGERDATE" "INTERSECT"
 "INTERSECTION" "INTERVAL" "INTO" "IO_AFTER_GTIDS" "IO_BEFORE_GTIDS"
"IS" "ISNULL" "ISOBID" "ISOLATION" "ITERATE" "JAR" "JOIN" "JOURNAL" "JSON_ARRAY"
 "JSON_ARRAYAGG" "JSON_EXISTS" "JSON_OBJECT" "JSON_OBJECTAGG"
"JSON_QUERY" "JSON_TABLE" "JSON_TABLE_PRIMITIVE" "JSON_VALUE" "KEEP" "KEY" "KEYS"
 "KILL" "KURTOSIS" "LABEL" "LAG" "LANGUAGE" "LARGE" "LAST"
"LAST_VALUE" "LATERAL" "LC_CTYPE" "LE" "LEAD" "LEADING" "LEAVE" "LEFT" "LESS" "LEVEL"
 "LIKE" "LIKE_REGEX" "LIMIT" "LINEAR" "LINENO" "LINES"
"LISTAGG" "LN" "LOAD" "LOADING" "LOCAL" "LOCALE" "LOCALTIME" "LOCALTIMESTAMP" "LOCATOR"
 "LOCATORS" "LOCK" "LOCKING" "LOCKMAX" "LOCKSIZE" "LOG"
"LOG10" "LOGGING" "LOGON" "LONG" "LONGBLOB" "LONGTEXT" "LOOP" "LOWER" "LOW_PRIORITY"
 "LT" "MACRO" "MAINTAINED" "MAP" "MASTER_BIND"
"MASTER_SSL_VERIFY_SERVER_CERT" "MATCH" "MATCHES" "MATCH_NUMBER" "MATCH_RECOGNIZE"
 "MATERIALIZED" "MAVG" "MAX" "MAXEXTENTS" "MAXIMUM" "MAXVALUE"
```

```
"MCHARACTERS" "MDIFF" "MEDIUMBLOB" "MEDIUMINT" "MEDIUMTEXT" "MEMBER" "MERGE" "METHOD"
 "MICROSECOND" "MICROSECONDS" "MIDDLEINT" "MIN" "MINDEX"
"MINIMUM" "MINUS" "MINUTE" "MINUTES" "MINUTE_MICROSECOND" "MINUTE_SECOND" "MLINREG"
 "MLOAD" "MLSLABEL" "MOD" "MODE" "MODIFIES" "MODIFY"
"MODULE" "MONITOR" "MONRESOURCE" "MONSESSION" "MONTH" "MONTHS" "MSUBSTR" "MSUM"
 "MULTISET" "NAMED" "NAMES" "NATIONAL" "NATURAL" "NCHAR" "NCLOB"
"NE" "NESTED_TABLE_ID" "NEW" "NEW_TABLE" "NEXT" "NEXTVAL" "NO" "NOAUDIT" "NOCHECK"
 "NOCOMPRESS" "NONCLUSTERED" "NONE" "NORMALIZE" "NOT" "NOTNULL"
"NOWAIT" "NO_WRITE_TO_BINLOG" "NTH_VALUE" "NTILE" "NULL" "NULLIF" "NULLIFZERO" "NULLS"
 "NUMBER" "NUMERIC" "NUMPARTS" "OBID" "OBJECT" "OBJECTS"
"OCCURRENCES_REGEX" "OCTET_LENGTH" "OF" "OFF" "OFFLINE" "OFFSET" "OFFSETS" "OLD"
 "OLD_TABLE" "OMIT" "ON" "ONE" "ONLINE" "ONLY" "OPEN" "OPENDATASOURCE"
"OPENQUERY" "OPENROWSET" "OPENXML" "OPERATION" "OPTIMIZATION" "OPTIMIZE"
 "OPTIMIZER_COSTS" "OPTION" "OPTIONALLY" "OR" "ORDER" "ORDINALITY" "ORGANIZATION"
"OUT" "OUTER" "OUTFILE" "OUTPUT" "OVER" "OVERLAPS" "OVERLAY" "OVERRIDE" "PACKAGE" "PAD"
 "PADDED" "PARAMETER" "PARAMETERS" "PART" "PARTIAL" "PARTITION"
"PARTITIONED" "PARTITIONING" "PASSWORD" "PATH" "PATTERN" "PCTFREE" "PER" "PERCENT"
 "PERCENTILE" "PERCENTILE_CONT" "PERCENTILE_DISC" "PERCENT_RANK" "PERIOD" "PERM"
"PERMANENT" "PIECESIZE" "PIVOT" "PLACING" "PLAN" "PORTION" "POSITION" "POSITION_REGEX"
 "POSTFIX" "POWER" "PRECEDES" "PRECISION" "PREFIX" "PREORDER"
"PREPARE" "PRESERVE" "PREVVAL" "PRIMARY" "PRINT" "PRIOR" "PRIQTY" "PRIVATE"
 "PRIVILEGES" "PROC" "PROCEDURE" "PROFILE" "PROGRAM" "PROPORTIONAL"
"PROTECTION" "PSID" "PTF" "PUBLIC" "PURGE" "QUALIFIED" "QUALIFY" "QUANTILE" "QUERY"
 "QUERYNO" "RADIANS" "RAISERROR" "RANDOM" "RANGE" "RANGE_N" "RANK"
"RAW" "READ" "READS" "READTEXT" "READ_WRITE" "REAL" "RECONFIGURE" "RECURSIVE" "REF"
 "REFERENCES" "REFERENCING" "REFRESH" "REGEXP" "REGR_AVGX" "REGR_AVGY"
"REGR_COUNT" "REGR_INTERCEPT" "REGR_R2" "REGR_SLOPE" "REGR_SXX" "REGR_SXY" "REGR_SYY"
 "RELATIVE" "RELEASE" "RENAME" "REPEAT" "REPLACE" "REPLICATION"
"REPOVERRIDE" "REQUEST" "REQUIRE" "RESIGNAL" "RESOURCE" "RESTART" "RESTORE" "RESTRICT"
 "RESULT" "RESULT_SET_LOCATOR" "RESUME" "RET" "RETRIEVE" "RETURN"
"RETURNING" "RETURNS" "REVALIDATE" "REVERT" "REVOKE" "RIGHT" "RIGHTS" "RLIKE" "ROLE"
 "ROLLBACK" "ROLLFORWARD" "ROLLUP" "ROUND_CEILING" "ROUND_DOWN"
"ROUND_FLOOR" "ROUND_HALF_DOWN" "ROUND_HALF_EVEN" "ROUND_HALF_UP" "ROUND_UP" "ROUTINE"
 "ROW" "ROWCOUNT" "ROWGUIDCOL" "ROWID" "ROWNUM" "ROWS" "ROWSET"
"ROW_NUMBER" "RULE" "RUN" "RUNNING" "SAMPLE" "SAMPLEID" "SAVE" "SAVEPOINT" "SCHEMA"
 "SCHEMAS" "SCOPE" "SCRATCHPAD" "SCROLL" "SEARCH" "SECOND" "SECONDS"
"SECOND_MICROSECOND" "SECQTY" "SECTION" "SECURITY" "SECURITYAUDIT" "SEEK" "SEL"
 "SELECT" "SEMANTICKEYPHRASETABLE" "SEMANTICSIMILARITYDETAILSTABLE"
"SEMANTICSIMILARITYTABLE" "SENSITIVE" "SEPARATOR" "SEQUENCE" "SESSION" "SESSION_USER"
 "SET" "SETRESRATE" "SETS" "SETSESSRATE" "SETUSER" "SHARE" "SHOW"
"SHUTDOWN" "SIGNAL" "SIMILAR" "SIMPLE" "SIN" "SINH" "SIZE" "SKEW" "SKIP" "SMALLINT"
 "SOME" "SOUNDEX" "SOURCE" "SPACE" "SPATIAL" "SPECIFIC" "SPECIFICTYPE"
"SPOOL" "SQL" "SQLEXCEPTION" "SQLSTATE" "SQLTEXT" "SQLWARNING" "SQL_BIG_RESULT"
 "SQL_CALC_FOUND_ROWS" "SQL_SMALL_RESULT" "SQRT" "SS" "SSL" "STANDARD"
```

```
"START" "STARTING" "STARTUP" "STAT" "STATE" "STATEMENT" "STATIC" "STATISTICS" "STAY"
 "STDDEV_POP" "STDDEV_SAMP" "STEPINFO" "STOGROUP" "STORED" "STORES"
"STRAIGHT_JOIN" "STRING_CS" "STRUCTURE" "STYLE" "SUBMULTISET" "SUBSCRIBER" "SUBSET"
 "SUBSTR" "SUBSTRING" "SUBSTRING_REGEX" "SUCCEEDS" "SUCCESSFUL"
"SUM" "SUMMARY" "SUSPEND" "SYMMETRIC" "SYNONYM" "SYSDATE" "SYSTEM" "SYSTEM_TIME"
 "SYSTEM_USER" "SYSTIMESTAMP" "TABLE" "TABLESAMPLE" "TABLESPACE" "TAN"
"TANH" "TBL_CS" "TEMPORARY" "TERMINATE" "TERMINATED" "TEXTSIZE" "THAN" "THEN"
 "THRESHOLD" "TIME" "TIMESTAMP" "TIMEZONE_HOUR" "TIMEZONE_MINUTE" "TINYBLOB"
"TINYINT" "TINYTEXT" "TITLE" "TO" "TOP" "TRACE" "TRAILING" "TRAN" "TRANSACTION"
 "TRANSLATE" "TRANSLATE_CHK" "TRANSLATE_REGEX" "TRANSLATION" "TREAT"
"TRIGGER" "TRIM" "TRIM_ARRAY" "TRUE" "TRUNCATE" "TRY_CONVERT" "TSEQUAL" "TYPE" "UC"
 "UESCAPE" "UID" "UNDEFINED" "UNDER" "UNDO" "UNION" "UNIQUE"
"UNKNOWN" "UNLOCK" "UNNEST" "UNPIVOT" "UNSIGNED" "UNTIL" "UPD" "UPDATE" "UPDATETEXT"
 "UPPER" "UPPERCASE" "USAGE" "USE" "USER" "USING" "UTC_DATE"
"UTC_TIME" "UTC_TIMESTAMP" "VALIDATE" "VALIDPROC" "VALUE" "VALUES" "VALUE_OF"
 "VARBINARY" "VARBYTE" "VARCHAR" "VARCHAR2" "VARCHARACTER" "VARGRAPHIC"
"VARIABLE" "VARIADIC" "VARIANT" "VARYING" "VAR_POP" "VAR_SAMP" "VCAT" "VERBOSE"
 "VERSIONING" "VIEW" "VIRTUAL" "VOLATILE" "VOLUMES" "WAIT" "WAITFOR"
"WHEN" "WHENEVER" "WHERE" "WHILE" "WIDTH_BUCKET" "WINDOW" "WITH" "WITHIN"
 "WITHIN_GROUP" "WITHOUT" "WLM" "WORK" "WRITE" "WRITETEXT" "XMLCAST" "XMLEXISTS"
"XMLNAMESPACES" "XOR" "YEAR" "YEARS" "YEAR_MONTH" "ZEROFILL" "ZEROIFNULL" "ZONE"
```

Criar alarmes em consultas ao Metrics Insights

Você pode criar alarmes nas consultas ao Metrics Insights. Isso ajuda você a ter alarmes que rastreiam vários recursos sem ser necessário atualizá-los posteriormente. A consulta captura os novos recursos e os recursos que sofrem alteações. Por exemplo, é possível criar um alarme que monitore a utilização de CPU da sua frota, e o alarme avaliará automaticamente as novas instâncias que você iniciar após criar o alarme.

Em uma conta de monitoramento configurada para observabilidade entre contas do CloudWatch, os alarmes do Metrics Insights podem observar recursos nas contas de origem e na própria conta de monitoramento. Para obter mais informações sobre como limitar suas consultas de alarme a uma conta específica ou agrupar os resultados por ID da conta, consulte as seções `WHERE` e `GROUP BY` em Componentes e sintaxe de consulta do Metrics Insights.

Sumário

Criar um alarme do Metrics Insights

Para criar um alarme em uma consulta ao Metrics Insights usando o console

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), All metrics (Todas as métricas).
3. Escolha a guia Queries (Consultas).
4. (Opcional) Para executar uma consulta de amostra pré-criada, escolha Add query (Adicionar consulta) e selecione a consulta a ser executada. Caso contrário, você pode escolher Editor para editar a consulta de exemplo e, em seguida, escolher Run (Executar) para executar a consulta modificada.
5. Para criar sua própria consulta, você pode usar as visualizações Builder (Compilador), Editor e também uma combinação dos dois. Você pode alternar entre as duas visualizações a qualquer momento e ver o trabalho em andamento em ambas.

   No Builder (Criador), você pode procurar e selecionar o namespace da métrica, o nome da métrica, o filtro, o grupo e as opções de ordem. Para cada uma dessas opções, o criador de consultas oferece uma lista de possíveis seleções do seu ambiente para escolher.

   Na visualização de Editor, você pode começar a gravar sua consulta. À medida que você digita, o editor oferece sugestões com base nos caracteres digitados até o momento.

   > **Important**
   >
   > Para definir um alarme em uma consulta ao Metrics Insights, a consulta deve retornar uma única série temporal. Se ela contiver uma instrução GROUP BY, a instrução deverá ser encapsulada em uma expressão matemática métrica que retorne apenas uma série temporal como resultado final da expressão.

6. Quando você estiver contente com sua consulta escolha Run (Executar).
7. Selecione Criar alarme.
8. Em Conditions (Condições), especifique o seguinte:

   a. Em Whenever ***metric*** is (Sempre que a métrica for), especifique se a métrica deve ser maior que, menor que ou igual ao limite. Em than... (que...), especifique o valor limite.

   b. Escolha Additional configuration (Configuração adicional). Em Datapoints to alarm (Pontos de dados para alarme), especifique quantos períodos de avaliação (pontos de

dados) devem estar no estado `ALARM` para disparar o alarme. Se os dois valores forem correspondentes, você criará um alarme que passa para o estado `ALARM` se esses períodos consecutivos estiverem violando.

Para criar um alarme M de N, especifique um número menor para o primeiro valor que especificar para o segundo valor. Para ter mais informações, consulte Avaliar um alarme.

c. Para o Missing data treatment (Tratamento de dados ausentes), escolha como deseja que o alarme se comporte quando alguns pontos de dados estiverem ausentes. Para ter mais informações, consulte Configurar como os alarmes do CloudWatch tratam dados ausentes.

9. Escolha Próximo.
10. Em Notification (Notificação), selecione um tópico do SNS para notificar quando o alarme estiver no estado `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

    Para que o alarme envie várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

    Para que o alarme não envie notificações, escolha Remove (Remover).
11. Para que o alarme execute ações do Auto Scaling, do EC2 ou do Systems Manager, escolha o botão apropriado, o estado do alarme e a ação a ser executada. Os alarmes só poderão executar ações do Systems Manager ao entrarem no estado ALARM. Para obter mais informações sobre ações do Systems Manager, consulte Configurar o CloudWatch para criar OpsItems a partir de alarmes e Criação de incidentes.

    > **Note**
    >
    > Para criar um alarme que executa uma ação do SSM Incident Manager, é necessário ter determinadas permissões. Para obter mais informações, consulte Exemplos de políticas baseadas em identidade para o AWSSystems Manager Incident Manager.

12. Quando terminar, escolha Next (Próximo).
13. Digite um nome e uma descrição para o alarme. O nome deve conter somente caracteres ASCII. Em seguida, escolha Próximo.
14. Em Preview and create (Visualizar e criar), confirme se as informações e condições são o que você deseja e escolha Create alarm (Criar alarme).

Para criar um alarme em uma consulta ao Metrics Insights usando a AWS CLI

- Use o comando `put-metric-alarm` e especifique uma consulta ao Metrics Insights no parâmetro `metrics`. Por exemplo, o comando a seguir define um alarme que entra no estado ALARM (ALARME) se alguma das instâncias ultrapassar 50% de utilização da CPU.

```
aws cloudwatch put-metric-alarm --alarm-name Metrics-Insights-alarm --
evaluation-periods 1 --comparison-operator GreaterThanThreshold --metrics
 '[{"Id":"m1","Expression":"SELECT MAX(CPUUtilization) FROM SCHEMA(\"AWS/EC2\",
 InstanceId)", "Period":60}]' --threshold 50
```

Casos de dados parciais

Se a consulta ao Metrics Insights usada para o alarme corresponder a mais de 10.000 métricas, o alarme será avaliado com base nas primeiras 10.000 métricas encontradas pela consulta. Isso significa que o alarme está sendo avaliado com base em dados parciais.

Você pode usar os métodos a seguir para descobrir se um alarme do Metrics Insights está avaliando seu estado de alarme com base em dados parciais:

- No console, se você escolher um alarme para ver a página Details (Detalhes), a mensagem Evaluation warning: Not evaluating all data (Aviso da avaliação: nem todos os dados estão sendo avaliados) será exibida nessa página.
- Você vê o valor `PARTIAL_DATA` no campo `EvaluationState` quando usar o comando describe-alarms da AWS CLI ou a API DescribeAlarms.

Os alarmes também publicam eventos no Amazon EventBridge quando ele entra no estado de dados parciais, para que você possa criar uma regra do EventBridge para observar esses eventos. Nesses eventos, o campo `evaluationState` tem o valor `PARTIAL_DATA`. Veja um exemplo a seguir.

```
{
    "version": "0",
    "id": "12345678-3bf9-6a09-dc46-12345EXAMPLE",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-11-08T11:26:05Z",
    "region": "us-east-1",
    "resources": [
```

```
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:my-alarm-name"
    ],
    "detail": {
        "alarmName": "my-alarm-name",
        "state": {
            "value": "ALARM",
            "reason": "Threshold Crossed: 3 out of the last 3 datapoints [20000.0
 (08/11/22 11:25:00), 20000.0 (08/11/22 11:24:00), 20000.0 (08/11/22 11:23:00)] were
 greater than the threshold (0.0) (minimum 1 datapoint for OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2022-11-08T11:26:05.399+0000\",\"startDate\":\"2022-11-08T11:23:00.000+0000\",
\"period\":60,\"recentDatapoints\":[20000.0,20000.0,20000.0],\"threshold\":0.0,
\"evaluatedDatapoints\":[{\"timestamp\":\"2022-11-08T11:25:00.000+0000\",\"value
\":20000.0}]}",
            "timestamp": "2022-11-08T11:26:05.401+0000",
            "evaluationState": "PARTIAL_DATA"
        },
        "previousState": {
            "value": "INSUFFICIENT_DATA",
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2022-11-08T11:25:51.227+0000"
        },
        "configuration": {
            "metrics": [
                {
                    "id": "m2",
                    "expression": "SELECT SUM(PartialDataTestMetric) FROM
 partial_data_test",
                    "returnData": true,
                    "period": 60
                }
            ]
        }
    }
}
```

Se a consulta para o alarme incluir uma instrução GROUP BY que retorne inicialmente mais de 500 séries temporais, o alarme será avaliado com base nas primeiras 500 séries temporais encontradas pela consulta. Porém, se você usar uma cláusula ORDER BY, todas as séries temporais que a consulta encontrar serão classificadas, e as 500 com os valores mais altos ou mais baixos, de acordo com sua cláusula ORDER BY, serão usadas para avaliar o alarme.

Usar consultas do Metrics Insights com matemática métrica

Você pode usar uma consulta do Metrics Insights como entrada para uma função matemática métrica. Para obter mais informações sobre matemática métrica, consulte Usar matemática de métricas.

Uma consulta do Metrics Insights que não inclui uma cláusula GROUP BY retorna uma única série temporal. Portanto, seus resultados retornados podem ser usados com qualquer função matemática métrica que tenha uma única série temporal como entrada.

Uma consulta do Metrics Insights que inclui uma cláusula GROUP BY retorna várias séries temporais. Portanto, seus resultados retornados podem ser usados com qualquer função matemática métrica que tenha uma matriz de séries temporais como entrada.

Por exemplo, a seguinte consulta retorna o número total de bytes baixados para cada bucket na região, como uma matriz de séries temporais:

```
SELECT SUM(BytesDownloaded)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
```

Em um gráfico no console ou em uma operação GetMetricData, os resultados desta consulta são q1. Essa consulta retorna o resultado em bytes. Portanto, se você quiser ver o resultado como MB, você pode usar a seguinte função matemática:

```
q1/1024/1024
```

Usar linguagem natural para gerar e atualizar as consultas do CloudWatch Metrics Insights

Esse recurso está em versão de pré-lançamento nas regiões Leste dos EUA (N. da Virgínia) e Oeste dos EUA (Óregon) e Pacífico Asiático (Tóquio) para o CloudWatch e está sujeito a alterações.

O CloudWatch é compatível com um recurso de consulta em linguagem natural para ajudar você a gerar e atualizar consultas para o CloudWatch Metrics Insights e o CloudWatch Logs Insights.

Com esse recurso, você pode fazer perguntas ou descrever os dados do CloudWatch que está procurando em inglês simples. O recurso de linguagem natural gera uma consulta com base em uma solicitação que você envia e fornece uma explicação detalhada de como a consulta funciona. Você também pode atualizar a consulta para investigar melhor seus dados.

Dependendo do ambiente, você pode inserir solicitações, como “Qual instância do Amazon Elastic Compute Cloud tem a maior saída de rede?” e “Mostre-me as dez principais tabelas do Amazon DynamoDB por leituras consumidas”.

Para gerar uma consulta do CloudWatch Metrics Insights com esse recurso, abra o editor de consultas do CloudWatch Metrics Insights na visualização do construtor ou do editor e escolha Gerar consulta.

> **Important**
>
> Para usar o recurso de consulta em linguagem natural, você deve usar a política CloudWatchFullAccess, CloudWatchReadOnlyAccess, CloudWatchFullAccessV2, AdministratorAccess ou ReadOnlyAccess.
> Você também pode incluir a ação `cloudwatch:GenerateQuery` em uma política nova ou atual gerenciada pelo cliente ou em uma política em linha.

Consultas de exemplo

Os exemplos nesta seção descrevem como gerar e atualizar consultas usando o recurso de linguagem natural.

> **Note**
>
> Para obter mais informações sobre o editor de consultas e a sintaxe do CloudWatch Metrics Insights, consulte Componentes e sintaxe de consulta do CloudWatch Metrics Insights.

Exemplo: gerar uma consulta em linguagem natural

Para gerar uma consulta usando linguagem natural, insira uma solicitação e escolha Gerar nova consulta. Este exemplo mostra uma consulta que executa uma pesquisa básica.

Prompt

Veja a seguir um exemplo de uma solicitação que direciona o recurso para pesquisar as dez principais tabelas do DynamoDB que consomem a maior capacidade de leitura.

```
Show top 10 DynamoDB Tables by consumed reads
```

Consulta

Veja a seguir um exemplo de consulta que o recurso de linguagem natural gera com base na solicitação. Observe como a solicitação aparece em um comentário antes da consulta. Depois da consulta, você pode ler uma explicação que descreve como a consulta funciona.

```
# Show top 10 DynamoDB Tables by consumed reads
SELECT SUM("ConsumedReadCapacityUnits")
FROM "AWS/DynamoDB"
GROUP BY TableName
ORDER BY SUM() DESC
LIMIT 10
# This query selects the sum of consumed read capacity units for each DynamoDB table,
 groups the results by table name, orders the results from highest to lowest read
 capacity consumption, and limits the results to the top 10 tables.
```

> **Note**
>
> Para desativar o surgimento da solicitação e a explicação de como a consulta funciona, use o ícone de engrenagem no editor.

Exemplo: atualizar uma consulta em linguagem natural

Você pode atualizar uma consulta editando a solicitação inicial e escolhendo Atualizar consulta.

Solicitação atualizada

O exemplo a seguir mostra uma versão atualizada da solicitação anterior. Em vez de uma solicitação que pesquisa as dez principais tabelas do DynamoDB que consomem a maior capacidade de leitura, essa solicitação agora direciona o recurso para classificar os resultados pelo número de bytes retornados.

```
Sort by bytes returned instead
```

Consulta atualizada

Veja a seguir um exemplo da consulta atualizada. Observe como a solicitação atualizada aparece em um comentário antes da consulta atualizada. Depois da consulta, você pode ler uma explicação que descreve como a consulta original foi atualizada.

```
# Sort by bytes returned instead
SELECT SUM("ReturnedBytes")
FROM "AWS/DynamoDB"
GROUP BY TableName
ORDER BY SUM() DESC
LIMIT 10
# This query modifies the original query to select the sum of returned bytes instead
 of consumed read capacity units, and orders the results from highest to lowest sum of
 returned bytes, limiting the results to the top 10 tables.
```

Optar por não usar seus dados para melhorar o serviço

Os dados das solicitações em linguagem natural que você fornece para treinar o modelo de IA e gerar consultas relevantes são usados exclusivamente para fornecer e manter seu serviço. Esses dados podem ser usados para melhorar a qualidade do CloudWatch Metrics Insights. Sua confiança e privacidade, além da segurança do seu conteúdo, são nossas maiores prioridades. Para obter mais informações, consulte Termos de Serviço da AWS e AWS responsible AI policy.

Você pode se recusar a ter seu conteúdo usado para desenvolver ou melhorar a qualidade das consultas em linguagem natural ao criar uma política de rejeição de serviços de IA. Para recusar a coleta de dados de todos os recursos de IA do CloudWatch, incluindo a geração de consultas, você deve criar uma política de recusa para o CloudWatch. Para obter mais informações, consulte Políticas de exclusão dos serviços de IA no Guia do usuário do AWS Organizations.

Inferência SQL

O CloudWatch Metrics Insights utiliza vários mecanismos para inferir a intenção de determinada consulta SQL.

Tópicos

- Intervalo de tempo
- Projeção de campos
- Agregação global ORDER BY

Intervalo de tempo

Os pontos de dados de séries temporais resultantes de uma consulta são agrupados em intervalos de tempo com base no período solicitado. Para agregar valores no SQL padrão, deve-se definir uma cláusula GROUP BY explícita para coletar todas as observações de um determinado período juntas. Como essa é a forma padrão de consultar dados de séries temporais, o CloudWatch Metrics Insights infere o intervalo de tempo sem a necessidade de expressar uma cláusula GROUP BY explícita.

Por exemplo, quando uma consulta é realizada com um período de um minuto, todas as observações pertencentes a esse minuto até o próximo (excluído) são acumuladas até a hora de início do bucket de tempo. Isso torna as instruções SQL do Metrics Insights mais concisas e menos detalhadas.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

A consulta anterior retorna uma única série temporal (pares de valores de timestamp), representando a utilização média da CPU de todas as instâncias do Amazon EC2. Supondo que o período solicitado seja de um minuto, cada ponto de dados retornado representa a média de todas as observações medidas dentro do intervalo específico de um minuto (incluindo a hora de início e excluindo a hora de término). O timestamp relacionado ao ponto de dados específico é a hora de início do bucket

Projeção de campos

As consultas do Metrics Insights sempre retornam a projeção de timestamp. Não é necessário especificar uma coluna timestamp na cláusula SELECT para obter o timestamp de cada valor de ponto de dados correspondente. Para obter detalhes sobre como o timestamp é calculado, consulte Intervalo de tempo.

Ao usar GROUP BY, cada nome de grupo também é inferido e projetado no resultado, para que você possa agrupar a série temporal retornada.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
```

A consulta anterior retorna uma série temporal para cada instância do Amazon EC2. Cada série temporal é rotulada após o valor do ID da instância.

Agregação global ORDER BY

Ao usar a ORDER BY, a FUNCTION() infere por qual função agregada você deseja ordenar os valores dos pontos de dados das métricas consultadas. A operação agregada é executada em todos os pontos de dados correspondentes de cada série temporal individual na janela de tempo consultada.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
ORDER BY MAX()
LIMIT 10
```

A consulta anterior retorna a utilização da CPU para cada instância do Amazon EC2, limitando o conjunto de resultados a 10 entradas. Os resultados são ordenados com base no valor máximo da série temporal individual na janela de tempo solicitada. A cláusula ORDER BY é aplicada antes de LIMIT, para que se calcule o ordenamento com relação a mais de dez séries temporais.

Consultas de exemplo do Metrics Insights

Esta seção contém exemplos de consultas úteis do CloudWatch Metrics Insights que você pode copiar e usar diretamente ou copiar e modificar no editor de consultas. Alguns desses exemplos já estão disponíveis no console, e você pode acessá-los escolhendo Add query (Adicionar consulta) na visualização de Metrics (Métricas).

Exemplos do Application Load Balancer

Total de solicitações em todos os balanceadores de carga

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer)
```

Os dez balanceadores de carga mais ativos

```
SELECT MAX(ActiveConnectionCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer)
GROUP BY LoadBalancer
ORDER BY SUM() DESC
LIMIT 10
```

Exemplos de uso de API da AWS

As 20 principais APIs da AWS pelo número de chamadas em sua conta

```
SELECT COUNT(CallCount)
FROM SCHEMA("AWS/Usage", Class, Resource, Service, Type)
WHERE Type = 'API'
GROUP BY Service, Resource
ORDER BY COUNT() DESC
LIMIT 20
```

APIs do CloudWatch classificadas por chamadas

```
SELECT COUNT(CallCount)
FROM SCHEMA("AWS/Usage", Class, Resource, Service, Type)
WHERE Type = 'API' AND Service = 'CloudWatch'
GROUP BY Resource
ORDER BY COUNT() DESC
```

Exemplos do DynamoDB

As dez principais tabelas por leituras consumidas

```
SELECT SUM(ProvisionedWriteCapacityUnits)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

As dez principais tabelas por bytes retornados

```
SELECT SUM(ReturnedBytes)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

As dez principais tabelas por erros do usuário

```
SELECT SUM(UserErrors)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Exemplos do Amazon Elastic Block Store

Os dez principais volumes do Amazon EBS por bytes gravados

```
SELECT SUM(VolumeWriteBytes)
FROM SCHEMA("AWS/EBS", VolumeId)
GROUP BY VolumeId
ORDER BY SUM() DESC
LIMIT 10
```

Tempo médio de gravação do volume do Amazon EBS

```
SELECT AVG(VolumeTotalWriteTime)
FROM SCHEMA("AWS/EBS", VolumeId)
```

Exemplos do Amazon EC2

Utilização da CPU por instâncias EC2 classificadas pela mais alta

```
SELECT AVG(CPUUtilization)
  FROM SCHEMA("AWS/EC2", InstanceId)
  GROUP BY InstanceId
  ORDER BY AVG() DESC
```

Utilização média da CPU em toda a frota

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

As dez principais instâncias por maior utilização da CPU

```
SELECT MAX(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
ORDER BY MAX() DESC
LIMIT 10
```

Nesse caso, o atendente do CloudWatch está coletando uma métrica **`CPUUtilization`** por aplicação. Esta consulta filtra a média dessa métrica por um nome de aplicação específico.

```
SELECT AVG(CPUUtilization)
FROM "AWS/CWAgent"
WHERE ApplicationName = 'eCommerce'
```

Exemplos do Amazon Elastic Container Service

Utilização média da CPU em todos os clusters do ECS

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/ECS", ClusterName)
```

Os dez principais clusters por utilização de memória

```
SELECT AVG(MemoryUtilization)
FROM SCHEMA("AWS/ECS", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC
LIMIT 10
```

Os dez principais serviços por utilização da CPU

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/ECS", ClusterName, ServiceName)
GROUP BY ClusterName, ServiceName
ORDER BY AVG() DESC
LIMIT 10
```

Os dez principais serviços por tarefas em execução (Container Insights)

```
SELECT AVG(RunningTaskCount)
FROM SCHEMA("ECS/ContainerInsights", ClusterName, ServiceName)
GROUP BY ClusterName, ServiceName
ORDER BY AVG() DESC
LIMIT 10
```

Exemplos do Amazon Elastic Kubernetes Service Container Insights

Utilização média da CPU em todos os clusters EKS

```
SELECT AVG(pod_cpu_utilization)
```

```
FROM SCHEMA("ContainerInsights", ClusterName)
```

Os dez principais clusters por utilização de nó de CPU

```
SELECT AVG(node_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC LIMIT 10
```

Os dez principais clusters por utilização de memória de pod

```
SELECT AVG(pop_memory_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC LIMIT 10
```

Os dez principais nós por utilização da CPU

```
SELECT AVG(node_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName, NodeName)
GROUP BY ClusterName, NodeName
ORDER BY AVG() DESC LIMIT 10
```

Os dez principais pods por utilização de memória

```
SELECT AVG(pod_memory_utilization)
FROM SCHEMA("ContainerInsights", ClusterName, PodName)
GROUP BY ClusterName, PodName
ORDER BY AVG() DESC LIMIT 10
```

Exemplos do EventBridge

As dez principais regras por invocações

```
SELECT SUM(Invocations)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

As dez principais regras por invocações que falharam

```
SELECT SUM(FailedInvocations)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

As dez principais regras por regras correspondentes

```
SELECT SUM(MatchedEvents)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Exemplos do Kinesis

Os dez principais fluxos por bytes gravados

```
SELECT SUM("PutRecords.Bytes")
FROM SCHEMA("AWS/Kinesis", StreamName)
GROUP BY StreamName
ORDER BY SUM() DESC LIMIT 10
```

Os dez principais fluxos pelos primeiros itens no fluxo

```
SELECT MAX("GetRecords.IteratorAgeMilliseconds")
FROM SCHEMA("AWS/Kinesis", StreamName)
GROUP BY StreamName
ORDER BY MAX() DESC LIMIT 10
```

Exemplos do Lambda

Funções do Lambda ordenadas por número de invocações

```
SELECT SUM(Invocations)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY SUM() DESC
```

As dez principais funções do Lambda por runtime mais longo

```
SELECT AVG(Duration)
```

```
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY MAX() DESC
LIMIT 10
```

As dez principais funções do Lambda por contagem de erros

```
SELECT SUM(Errors)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY SUM() DESC
LIMIT 10
```

Exemplos do CloudWatch Logs

Os dez principais grupos de logs por eventos recebidos

```
SELECT SUM(IncomingLogEvents)
FROM SCHEMA("AWS/Logs", LogGroupName)
GROUP BY LogGroupName
ORDER BY SUM() DESC LIMIT 10
```

Os dez principais grupos de logs por bytes gravados

```
SELECT SUM(IncomingBytes)
FROM SCHEMA("AWS/Logs", LogGroupName)
GROUP BY LogGroupName
ORDER BY SUM() DESC LIMIT 10
```

Exemplos do Amazon RDS

As dez principais instâncias do Amazon RDS pela maior utilização da CPU

```
SELECT MAX(CPUUtilization)
FROM SCHEMA("AWS/RDS", DBInstanceIdentifier)
GROUP BY DBInstanceIdentifier
ORDER BY MAX() DESC
LIMIT 10
```

Os dez principais clusters do Amazon RDS por gravações

```
SELECT SUM(WriteIOPS)
FROM SCHEMA("AWS/RDS", DBClusterIdentifier)
GROUP BY DBClusterIdentifier
ORDER BY MAX() DESC
LIMIT 10
```

Exemplos do Amazon Simple Storage Service para PHP

Latência média por bucket

```
SELECT AVG(TotalRequestLatency)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
ORDER BY AVG() DESC
```

Os dez principais buckets por bytes baixados

```
SELECT SUM(BytesDownloaded)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
ORDER BY SUM() DESC
LIMIT 10
```

Exemplos do Amazon Simple Notification Service

Total de mensagens publicadas por tópicos do SNS

```
SELECT SUM(NumberOfMessagesPublished)
FROM SCHEMA("AWS/SNS", TopicName)
```

Os dez principais tópicos por mensagens publicadas

```
SELECT SUM(NumberOfMessagesPublished)
FROM SCHEMA("AWS/SNS", TopicName)
GROUP BY TopicName
ORDER BY SUM() DESC
LIMIT 10
```

Os dez principais tópicos por falhas na entrega de mensagens

```
SELECT SUM(NumberOfNotificationsFailed)
FROM SCHEMA("AWS/SNS", TopicName)
GROUP BY TopicName
ORDER BY SUM() DESC
LIMIT 10
```

Exemplos do Amazon SQS

As dez principais filas por número de mensagens visíveis

```
SELECT AVG(ApproximateNumberOfMessagesVisible)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY AVG() DESC
LIMIT 10
```

As dez principais filas mais ativas

```
SELECT SUM(NumberOfMessagesSent)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY SUM() DESC
LIMIT 10
```

As dez principais filas por idade da primeira mensagem

```
SELECT AVG(ApproximateAgeOfOldestMessage)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY AVG() DESC
LIMIT 10
```

Limites do Metrics Insights

No momento, o CloudWatch Metrics Insights tem os seguintes limites:

- Atualmente, você pode consultar apenas as três horas mais recentes de dados.
- Uma consulta única não pode processar mais de 10.000 métricas. Isto significa que se as cláusulas SELECT, FROM eWHERE correspondem a mais de 10.000 métricas, a consulta processa apenas as primeiras 10.000 entre as métricas encontradas.

- Uma consulta única não pode retornar mais de 500 séries temporais. Isso significa que, se a consulta puder retornar mais de 500 métricas, nem todas as métricas serão retornadas nos resultados da consulta. Se você usar uma cláusula ORDER BY, todas as métricas que estão sendo processadas serão classificadas, e as 500 que têm os valores mais altos ou mais baixos de acordo com sua cláusula ORDER BY serão retornadas.

  Se você não incluir uma cláusula ORDER BY, não poderá controlar quais 500 métricas correspondentes serão retornadas.
- Você pode ter até 200 alarmes do Metrics Insights por região.
- O Metrics Inights não é compatível com dados de alta resolução, que são dados métricos relatados com uma granularidade de menos de um minuto. Se você solicitar dados de alta resolução, a solicitação não falhará, mas a saída será agregada com a granularidade de um minuto.
- Cada operação GetMetricData pode ter apenas uma consulta. No entanto, você pode ter vários widgets em um painel para que cada um inclua uma consulta.

Glossário do Metrics Insights

rótulo

No Metrics Insights, um rótulo é um par de chave-valor usado na definição do escopo de uma consulta para que ela retorne um determinado conjunto de dados ou na definição de critérios pelos quais os resultados da consulta devem ser separados em séries temporais distintas. Uma chave de rótulo é semelhante ao nome de uma coluna no SQL. Atualmente, os rótulos devem ser dimensões métricas do CloudWatch.

observação

Uma observação é um valor registrado para uma determinada métrica em um determinado momento.

Solução de problemas do métricas do Metrics Insights

Os resultados incluem “Outros”, mas não tenho esse valor como uma dimensão

Isso significa que a consulta inclui uma cláusula GROUP BY que especifica uma chave de rótulo que não está sendo usada em algumas das métricas retornadas pela consulta. Nesse caso, um grupo nulo denominado `Other` é retornado. As métricas que não incluem essa chave de rótulo

provavelmente são métricas agregadas que retornam valores agregados em todos os valores dessa chave de rótulo.

Por exemplo, suponha que tenhamos a seguinte consulta:

```
SELECT AVG(Faults)
FROM MyCustomNamespace
GROUP BY Operation, ServiceName
```

Se algumas das métricas retornadas não incluírem `ServiceName` como uma dimensão, elas são exibidas tendo `Other` como o valor para `ServiceName`.

Para evitar ver “Outros” em seus resultados, use SCHEMA na sua cláusula FROM, como no seguinte exemplo:

```
SELECT AVG(Faults)
FROM SCHEMA(MyCustomNamespace, Operation)
GROUP BY Operation, ServiceName
```

Isso limita os resultados retornados apenas às métricas que têm ambas as dimensões `Operation` e `ServiceName`.

O timestamp mais antigo no meu gráfico tem um valor de métrica menor do que os outros

O CloudWatch Metrics Insights atualmente suporta apenas as últimas três horas de dados. Quando você faz um gráfico com um período maior que um minuto, pode haver casos em que o ponto de dados mais antigo difere do valor esperado. Isso ocorre porque as consultas do Metrics Insights retornam somente as 3 horas mais recentes de dados. Nesse caso, o ponto de dados mais antigo na consulta retorna somente as observações que foram medidas dentro do limite das últimas três horas, em vez de retornar todas as observações dentro do período desse ponto de dados.

Use o explorador de métricas para monitorar recursos a partir de suas etiquetas e propriedades

O explorador de métricas é uma ferramenta baseada em etiquetas que permite filtrar, agregar e visualizar suas métricas por etiquetas e propriedades de recursos, para melhorar a observabilidade de seus serviços. Isso oferece uma experiência de solução de problemas flexível e dinâmica, para que você crie vários grafos por vez e use esses grafos para criar painéis de integridade da aplicação.

As visualizações do explorador de métricas são dinâmicas. Portanto, se um recurso correspondente for criado depois de criar um widget do explorador de métricas e adicioná-lo a um painel do CloudWatch, o novo recurso será exibido automaticamente no widget do explorador.

Por exemplo, se todas as suas instâncias de produção do EC2 tiverem a etiqueta **`production`**, você pode usar o explorador de métricas para filtrar e agregar métricas de todas essas instâncias para entender sua integridade e performance. Se uma nova instância com uma etiqueta correspondente for criada posteriormente, ela será adicionada automaticamente ao widget do explorador de métricas.

> **Note**
>
> O Metrics Explorer fornece uma experiência em um ponto no tempo. Os recursos que tenham sido encerrados ou que não existam mais com a propriedade ou a tag que você especificou não serão exibidos na visualização. Contudo, você ainda pode encontrar as métricas desses recursos nas visualizações de métricas do CloudWatch.

Com o explorador de métricas, você pode escolher como agregar métricas dos recursos que correspondem aos critérios e se deseja exibi-las todas em um único grafo ou em grafos diferentes dentro de um widget do explorador de métricas.

O explorador de métricas contém modelos que você pode usar para ver grafos de visualização úteis com um clique, e também é possível estender esses modelos para criar widgets de explorador de métricas completamente personalizados.

O gerenciador de métricas é compatível com métricas emitidas pela AWS e métricas do EC2 que são publicadas pelo agente do CloudWatch, inclusive métricas de memória, disco e CPU. Para usar o explorador de métricas a fim de ver as métricas publicadas pelo agente do CloudWatch, talvez seja necessário atualizar o arquivo de configuração do agente do CloudWatch. Para obter mais informações, consulte Configuração do agente do CloudWatch para o explorador de métricas

Para criar uma visualização com o explorador de métricas e, opcionalmente, adicioná-la a um painel, siga estas etapas.

Para criar uma visualização com o explorador de métricas

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Explorer.
3. Faça um dos seguintes procedimentos:

- Para usar um modelo, selecione-o na caixa que exibe Empty Explorer (Explorador vazio).

  Dependendo do modelo, o explorador pode exibir grafos de métricas imediatamente. Se isso não acontecer, escolha uma ou mais etiquets ou propriedades da lista From (De), e os dados deverão aparecer. Senão, use as opções na parte superior da página para exibir um intervalo de tempo mais longo nos grafos.
- Para criar uma visualização personalizada, em Metrics (Métricas), escolha uma única métrica ou todas as métricas disponíveis de um serviço.

  Se preferir, você pode repetir essa etapa para adicionar mais métricas.

4. Para cada métrica selecionada, o CloudWatch exibe a estatística que será usada imediatamente após o nome da métrica. Para alterar isso, escolha o nome da estatística e selecione a estatística desejada.
5. Em From (De), escolha uma etiqueta ou uma propriedade do recurso para filtrar seus resultados.

   Depois de fazer isso, você pode, opcionalmente, repetir essa etapa para escolher mais etiquetas ou propriedades do recurso.

   Se você escolher vários valores da mesma propriedade, como dois tipos de instância do EC2, o explorador exibirá todos os recursos que correspondem à propriedade escolhida. É tratado como uma operação OR.

   Se você escolher propriedades ou etiquetas diferentes, como a etiqueta **`Production`** e o tipo de instância M5, somente os recursos que correspondem a todas essas seleções serão exibidos. É tratado como uma operação AND.
6. (Opcional) Em Aggregate by (Agregar por), escolha uma estatística a ser usada para agregar as métricas. Em seguida, ao lado de for (para), escolha como agregar a métrica na lista. É possível agregar todos os recursos que são exibidos no momento ou agregar por uma única etiqueta ou propriedade do recurso.

   Dependendo de como você escolher agregar, o resultado pode ser uma única série temporal ou várias séries temporais.
7. Em Split by (Dividir por), você pode optar por dividir um único gráfico com várias séries temporais em vários gráficos. A divisão pode ser feita por critérios variados, que você escolhe em Split by (Dividir por).
8. Em Graph options (Opções de grafos), é possível refinar o grafo alterando o período, o tipo de grafo, o posicionamento da legenda e o layout.

9. Para adicionar esta visualização como um widget a um painel do CloudWatch, escolha Add to dashboard (Adicionar ao painel).

Configuração do agente do CloudWatch para o explorador de métricas

Para habilitar o explorador de métricas a detectar as métricas do EC2 publicadas pelo agente do CloudWatch, verifique se o arquivo de configuração do agente do CloudWatch contém os seguintes valores:

- Na seção `metrics`, verifique se o parâmetro `aggregation_dimensions` contém `["InstanceId"]`. Também pode conter outras dimensões.
- Na seção `metrics`, verifique se o parâmetro `append_dimensions` contém uma linha `{"InstanceId":"${aws:InstanceId}"}`. Também pode conter outras linhas.
- Na seção `metrics`, dentro da seção `metrics_collected`, verifique as seções para cada tipo de recurso que você deseja que o explorador de métricas detecte, como as seções `cpu`, `disk` e `memory`. Confira se cada uma dessas seções contém um `"resources": [ "*"] line.`.
- Na seção `cpu` da seção `metrics_collected`, verifique se existe uma linha `"totalcpu": true`.
- Você deve usar o namespace `CWAgent` padrão para as métricas coletadas pelo agente do CloudWatch, em vez de um namespace personalizado.

As configurações na lista anterior fazem com que o agente do CloudWatch publique métricas agregadas para discos, CPUs e outros recursos que podem ser representados no explorador de métricas para todas as instâncias que o utilizam.

Essas configurações republicarão as métricas que você configurou anteriormente para serem publicadas com várias dimensões, adicionando aos custos de métrica.

Para obter mais informações sobre como editar as configurações no arquivo de configuração do agente do CloudWatch, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

Usar fluxos de métricas

Você pode usar fluxos de métricas para transmitir continuamente as métricas do CloudWatch para um destino de sua preferência, com entrega quase em tempo real e baixa latência. Entre os destinos

compatíveis estão os destinos da AWS, como o Amazon Simple Storage Service e vários destinos de provedores de serviços de terceiros.

Há três cenários principais de uso para os fluxos de métricas do CloudWatch:

- Configuração personalizada com o Firehose: crie um fluxo de métricas e direcione-o para um fluxo de entrega do Amazon Data Firehose que envie suas métricas do CloudWatch para onde você deseja. Você pode transmiti-las para um data lake, como o Amazon S3, ou para qualquer destino ou endpoint compatível com o Firehose, incluindo provedores externos. Os formatos JSON, OpenTelemetry 1.0.0 e OpenTelemetry 0.7.0 são compatíveis nativamente ou você pode configurar transformações no fluxo de entrega do Firehose para converter os dados em um formato diferente, como o Parquet. Com um fluxo de métricas, você pode atualizar continuamente os dados de monitoramento ou combinar esses dados de métricas do CloudWatch com dados de faturamento e performance para criar conjuntos de dados avançados. Em seguida, você pode usar ferramentas como o Amazon Athena para obter insights sobre otimização de custos, performance de recursos e utilização de recursos.
- Configuração rápida do S3: crie o fluxo para o Amazon Simple Storage Service com um processo de configuração rápida. Por padrão, o CloudWatch cria os recursos necessários para o fluxo. Os formatos JSON, OpenTelemetry 1.0.0 e OpenTelemetry 0.7.0 são compatíveis.
- Configuração rápida do parceiro da AWS: o CloudWatch oferece uma experiência de configuração rápida para alguns parceiros externos. Você pode usar provedores de serviços terceirizados para monitorar, solucionar problemas e analisar suas aplicações usando os dados transmitidos do CloudWatch. Quando você usa o fluxo de trabalho de configuração rápida de parceiros, é necessário fornecer apenas um URL de destino e uma chave de API para o destino, e o CloudWatch cuida do restante da configuração. A configuração rápida de parceiros está disponível para os seguintes provedores externos:

  - Datadog
  - Dynatrace
  - New Relic
  - Splunk Observability Cloud
  - SumoLogic

É possível transmitir todas as suas métricas do CloudWatch ou usar filtros para transmitir somente métricas especificadas. Cada fluxo de métrica pode conter até 1000 filtros que incluem ou excluem

namespaces de métrica ou métricas específicas. Um único fluxo de métricas pode ter somente filtros de inclusão ou exclusão, mas não ambos.

Depois que um fluxo de métrica é criado, se novas métricas forem criadas que correspondam aos filtros no local, as novas métricas serão incluídas no fluxo automaticamente.

Não há limite para o número de fluxos de métrica por conta ou por região e não há limite para o número de atualizações de métricas que estão sendo transmitidas.

Cada fluxo pode usar os formatos JSON, OpenTelemetry 1.0.0 ou OpenTelemetry 0.7.0. Você pode editar o formato de saída de um fluxo de métrica a qualquer momento, por exemplo, para atualizar do OpenTelemetry 0.7.0 para o OpenTelemetry 1.0.0. Para obter mais informações sobre formatos de saída, consulte Formatos de saída de fluxos de métricas.

Para fluxos de métricas em contas de monitoramento, escolha se você deseja incluir métricas das contas de origem vinculadas a essa conta de monitoramento. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

Os fluxos de métrica sempre incluem as estatísticas `Minimum`, `Maximum`, `SampleCount` e `Sum`. Você também pode optar por incluir estatísticas adicionais mediante um custo adicional. Para ter mais informações, consulte Estatísticas que podem ser transmitidas.

O preço dos fluxos de métrica é baseado no número de atualizações de métricas. Você também vai gerar cobranças do Firehose para o fluxo de entrega usado no fluxo de métrica. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

Tópicos

Configurar um fluxo de métricas

Use as etapas nas seções a seguir para configurar um fluxo de métricas do CloudWatch.

Depois que um fluxo de métricas é criado, o tempo necessário para que os dados de métrica sejam exibidos no destino depende das configurações de buffer no fluxo de entrega do Firehose. O buffer é expresso em tamanho máximo da carga útil ou tempo máximo de espera, o que for atingido primeiro. Se estes forem definidos com os valores mínimos (60 segundos, 1 MB), a latência esperada será dentro de 3 minutos, se os namespaces do CloudWatch selecionados tiverem atualizações de métricas ativas.

Em um fluxo de métricas do CloudWatch, os dados são enviados a cada minuto. Os dados podem chegar ao destino final fora de ordem. Todas as métricas nos namespaces especificados são enviadas no fluxo de métricas, com a exceção de métricas com um carimbo de data/hora com mais de dois dias de idade.

Para cada combinação de nome da métrica e namespace que você transmitir, todas as combinações de dimensão desse nome da métrica e namespace são transmitidas.

Para fluxos de métricas em contas de monitoramento, escolha se você deseja incluir métricas das contas de origem vinculadas a essa conta de monitoramento. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

Para criar e gerenciar um fluxo de métricas, você deve estar conectado a uma conta que tenha a política CloudWatchFullAccess e a permissão `iam:PassRole` ou uma conta que tenha a seguinte lista de permissões:

- `iam:PassRole`
- `cloudwatch:PutMetricStream`
- `cloudwatch:DeleteMetricStream`
- `cloudwatch:GetMetricStream`
- `cloudwatch:ListMetricStreams`
- `cloudwatch:StartMetricStreams`
- `cloudwatch:StopMetricStreams`

Para que o CloudWatch configure a função do IAM necessária para fluxos de métricas, você também deve ter as permissões `iam:CreateRole` e `iam:PutRolePolicy`.

> **Important**
>
> Um usuário com `cloudwatch:PutMetricStream` tem acesso aos dados métricos do CloudWatch que estão sendo transmitidos, mesmo que não tenha a permissão `cloudwatch:GetMetricData`.

Tópicos

Configuração personalizada com o Firehose

Use esse método para criar um fluxo de métricas e direcioná-lo a um fluxo de entrega do Amazon Data Firehose que envia suas métricas do CloudWatch para onde você deseja. Você pode transmiti-las para um data lake, como o Amazon S3, ou para qualquer destino ou endpoint compatível com o Firehose, incluindo provedores externos.

Os formatos JSON, OpenTelemetry 1.0.0 e OpenTelemetry 0.7.0 são compatíveis nativamente ou você pode configurar transformações no fluxo de entrega do Firehose para converter os dados em um formato diferente, como o Parquet. Com um fluxo de métricas, você pode atualizar continuamente os dados de monitoramento ou combinar esses dados de métricas do CloudWatch com dados de faturamento e performance para criar conjuntos de dados avançados. Em seguida, você pode usar ferramentas como o Amazon Athena para obter insights sobre otimização de custos, performance de recursos e utilização de recursos.

Você pode usar o console do CloudWatch, a AWS CLI, o AWS CloudFormation ou o AWS Cloud Development Kit (AWS CDK) para configurar um fluxo de métricas.

O fluxo de entrega do Firehose usado para o fluxo de métricas deve estar na mesma conta e na mesma região em que o fluxo de métricas está configurado. Para obter a funcionalidade entre regiões, é possível configurar o fluxo de entrega do Firehose para transmitir a um destino final que esteja em outra conta ou região.

Console do CloudWatch

Esta seção descreve como usar o console do CloudWatch para configurar um fluxo de métricas usando o Firehose.

Para configurar um fluxo de métricas personalizado usando o Firehose

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), Streams (Fluxos). Em seguida, escolha Create metric stream (Criar fluxo de métrica).
3. (Opcional) Se estiver conectado a uma conta configurada como uma conta de monitoramento na observabilidade entre contas do CloudWatch, você poderá escolher se deseja incluir métricas de contas de origem vinculadas nesse fluxo de métricas. Para incluir métricas de contas de origem, escolha Include source account metrics (Incluir métricas da conta de origem).
4. Escolha Configuração personalizada com Firehose.
5. Em Selecionar o fluxo do Kinesis Data Firehose, selecione o fluxo de entrega do Firehose a ser usado. Ele deve estar na mesma conta. O formato padrão para essa opção é OpenTelemetry 0.7.0, mas é possível alterar o formato posteriormente nesse procedimento.

   Em seguida, selecione o fluxo de entrega do Firehose a ser usado em Selecionar o fluxo de entrega do Firehose.
6. (Opcional) Você pode escolher Selecionar perfil de serviço existente para usar um perfil do IAM existente em vez de fazer com que o CloudWatch crie um novo para você.
7. (Opcional) Para alterar o formato de saída do formato padrão para seu cenário, escolha Change output format (Alterar o formato de saída). Os formatos compatíveis são JSON, OpenTelemetry 1.0.0 e OpenTelemetry 0.7.0.
8. Em Métricas a serem transmitidas, escolha Todas as métricas ou Selecionar métricas.

   Se você escolher Todas as métricas, todas as métricas dessa conta serão incluídas no fluxo.

   Avalie bem se deseja transmitir todas as métricas, pois quanto mais métricas forem transmitidas, maiores serão as cobranças do fluxo de métrica.

   Se você escolgher Selecionar métrica, faça o seguinte:

   - Para transmitir a maioria dos namespaces métricos, escolha Excluir e selecione os namespaces ou métricas a serem excluídos. Ao especificar um namespace em Excluir, você também pode selecionar algumas métricas específicas desse namespace para excluir.

Se você optar por excluir um namespace, mas não selecionar métricas nesse namespace, todas as métricas desse namespace serão excluídas.

- Para incluir apenas alguns namespaces de métrica ou métricas no fluxo de métricas, escolha Incluir e selecione os namespaces ou métricas que deseja incluir. Se você optar por incluir um namespace, mas não selecionar métricas nesse namespace, todas as métricas desse namespace serão incluídas.

9. (Opcional) Para transmitir estatísticas adicionais para algumas dessas métricas além de Mínimo, Máximo, SampleCount e Soma, selecione Adicionar estatísticas adicionais. Escolha Add recommended metrics (Adicionar métricas recomendadas) para adicionar algumas estatísticas comumente usadas ou selecione manualmente o namespace e o nome da métrica para os quais deseja transmitir estatísticas adicionais. Em seguida, selecione as estatísticas adicionais que deseja transmitir.

   Em seguida, para escolher outro grupo de métricas para o qual transmitir um conjunto diferente de estatísticas adicionais, escolha Add additional statistics (Adicionar outras estatísticas). Cada métrica pode incluir até 20 estatísticas adicionais, e um fluxo de métrica pode ter até 100 métricas com estatísticas adicionais.

   A transmissão de estatísticas adicionais gera mais cobranças. Para ter mais informações, consulte Estatísticas que podem ser transmitidas.

   Para ver as definições das estatísticas adicionais, consulte Definições de estatísticas do CloudWatch.

10. (Opcional) Personalize o nome do novo fluxo de métricas em Metric stream name (Nome do fluxo de métricas).
11. Escolha Create metric stream (Criar filtro de métrica).

AWS CLI ou API da AWS

Use as etapas a seguir para criar um fluxo de métricas do CloudWatch.

Para usar a AWS CLI ou a API da AWSpara criar um fluxo de métricas

1. Se estiver transmitindo para o Amazon S3, primeiro crie o bucket. Para mais informações, consulte Criar um bucket.
2. Crie o fluxo de entrega do Firehose. Para obter mais informações, consulte Criar um fluxo do Firehose.

3. Crie um perfil do IAM que permita que o CloudWatch grave no fluxo de entrega do Firehose. Para obter mais informações sobre o conteúdo dessa função, consulte Confiança entre o CloudWatch e o Firehose.
4. Use o comando `aws cloudwatch put-metric-stream` da CLI a API `PutMetricStream` para criar o fluxo de métricas do CloudWatch.

AWS CloudFormation

É possível usar o AWS CloudFormation para configurar um fluxo de métricas. Para obter mais informações, consulte AWS::CloudWatch::MetricStream.

Para usar o AWS CloudFormation para criar um fluxo de métricas

1. Se estiver transmitindo para o Amazon S3, primeiro crie o bucket. Para mais informações, consulte Criar um bucket.
2. Crie o fluxo de entrega do Firehose. Para obter mais informações, consulte Criar um fluxo do Firehose.
3. Crie um perfil do IAM que permita que o CloudWatch grave no fluxo de entrega do Firehose. Para obter mais informações sobre o conteúdo dessa função, consulte Confiança entre o CloudWatch e o Firehose.
4. Crie o fluxo no AWS CloudFormation. Para obter mais informações, consulte AWS::CloudWatch::MetricStream.

AWS Cloud Development Kit (AWS CDK)

É possível usar o AWS Cloud Development Kit (AWS CDK) para configurar um fluxo de métricas.

Para usar o AWS CDK para criar um fluxo de métricas

1. Se estiver transmitindo para o Amazon S3, primeiro crie o bucket. Para mais informações, consulte Criar um bucket.
2. Crie o fluxo de entrega do Firehose. Para obter mais informações, consulte Creating an Amazon Data Firehose Delivery Stream.
3. Crie um perfil do IAM que permita que o CloudWatch grave no fluxo de entrega do Firehose. Para obter mais informações sobre o conteúdo dessa função, consulte Confiança entre o CloudWatch e o Firehose.

4. Crie o fluxo de métricas. O recurso de fluxo de métricas está disponível no AWS CDK como construção de nível 1 (L1) chamado `CfnMetricStream`. Para obter mais informações, consulte Usar construções L1.

Use a configuração rápida do Amazon S3

O método Configuração rápida do S3 funcionará bem se você quiser configurar rapidamente um fluxo para o Amazon S3 e não precisar de qualquer transformação de formatação além dos formatos compatíveis JSON, OpenTelemetry 1.0.0 e OpenTelemetry 0.7.0. O CloudWatch criará todos os recursos necessários, incluindo o fluxo de entrega do Firehose e os perfis do IAM necessários. O formato padrão para essa opção é JSON, mas você pode alterar o formato ao configurar o fluxo.

Como alternativa, se você quiser que o formato final seja o formato Parquet ou Optimized Row Columnar (ORC), você deverá seguir as etapas em Configuração personalizada com o Firehose.

Console do CloudWatch

Esta seção descreve como usar o console do CloudWatch para configurar um fluxo de métricas do Amazon S3 usando a configuração rápida do S3.

Configurar um fluxo de métricas usando a configuração rápida do S3

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), Streams (Fluxos). Em seguida, escolha Create metric stream (Criar fluxo de métrica).
3. (Opcional) Se estiver conectado a uma conta configurada como uma conta de monitoramento na observabilidade entre contas do CloudWatch, você poderá escolher se deseja incluir métricas de contas de origem vinculadas nesse fluxo de métricas. Para incluir métricas de contas de origem, escolha Include source account metrics (Incluir métricas da conta de origem).
4. Selecione Configuração rápida do S3. O CloudWatch criará todos os recursos necessários, incluindo o fluxo de entrega do Firehose e os perfis do IAM necessários. O formato padrão para essa opção é JSON, mas é possível alterar o formato posteriormente neste procedimento.
5. (Opcional) Escolha Selecionar recursos existentes para usar um bucket do S3 existente ou perfis do IAM existentes em vez de fazer com que o CloudWatch crie novos recursos para você.
6. (Opcional) Para alterar o formato de saída do formato padrão para seu cenário, escolha Change output format (Alterar o formato de saída). Os formatos compatíveis são JSON, OpenTelemetry 1.0.0 e OpenTelemetry 0.7.0.

7. Em Métricas a serem transmitidas, escolha Todas as métricas ou Selecionar métricas.

   Se você escolher Todas as métricas, todas as métricas dessa conta serão incluídas no fluxo.

   Avalie bem se deseja transmitir todas as métricas, pois quanto mais métricas forem transmitidas, maiores serão as cobranças do fluxo de métrica.

   Se você escolgher Selecionar métrica, faça o seguinte:

   - Para transmitir a maioria dos namespaces métricos, escolha Excluir e selecione os namespaces ou métricas a serem excluídos. Ao especificar um namespace em Excluir, você também pode selecionar algumas métricas específicas desse namespace para excluir. Se você optar por excluir um namespace, mas não selecionar métricas nesse namespace, todas as métricas desse namespace serão excluídas.
   - Para incluir apenas alguns namespaces de métrica ou métricas no fluxo de métricas, escolha Incluir e selecione os namespaces ou métricas que deseja incluir. Se você optar por incluir um namespace, mas não selecionar métricas nesse namespace, todas as métricas desse namespace serão incluídas.

8. (Opcional) Para transmitir estatísticas adicionais para algumas dessas métricas além de Mínimo, Máximo, SampleCount e Soma, selecione Adicionar estatísticas adicionais. Escolha Add recommended metrics (Adicionar métricas recomendadas) para adicionar algumas estatísticas comumente usadas ou selecione manualmente o namespace e o nome da métrica para os quais deseja transmitir estatísticas adicionais. Em seguida, selecione as estatísticas adicionais que deseja transmitir.

   Em seguida, para escolher outro grupo de métricas para o qual transmitir um conjunto diferente de estatísticas adicionais, escolha Add additional statistics (Adicionar outras estatísticas). Cada métrica pode incluir até 20 estatísticas adicionais, e um fluxo de métrica pode ter até 100 métricas com estatísticas adicionais.

   A transmissão de estatísticas adicionais gera mais cobranças. Para ter mais informações, consulte Estatísticas que podem ser transmitidas.

   Para ver as definições das estatísticas adicionais, consulte Definições de estatísticas do CloudWatch.

9. (Opcional) Personalize o nome do novo fluxo de métricas em Metric stream name (Nome do fluxo de métricas).

10. Escolha Create metric stream (Criar filtro de métrica).

Configuração rápida de parceiros

O CloudWatch oferece uma experiência de configuração rápida para os parceiros externos a seguir. Para usar esse fluxo de trabalho, você precisa fornecer apenas um URL de destino e uma chave de API para seu destino. O CloudWatch cuida do restante da configuração, incluindo a criação do fluxo de entrega do Firehose e os perfis do IAM necessários.

> **Important**
>
> Antes de usar a configuração rápida do parceiro para criar um fluxo de métricas, é altamente recomendável que você leia a documentação desse parceiro, que está relacionada na lista a seguir.

- Datadog
- Dynatrace
- New Relic
- Splunk Observability Cloud
- SumoLogic

Quando você configura um fluxo de métricas para um desses parceiros, o fluxo é criado com algumas configurações padrão, conforme listado nas seções a seguir.

Tópicos

Configurar um fluxo de métricas usando a configuração rápida de parceiros

O CloudWatch oferece uma opção de configuração rápida para alguns parceiros terceirizados. Antes de iniciar as etapas desta seção, é necessário ter determinadas informações sobre o parceiro. Essas informações podem incluir um URL de destino e/ou uma chave de API para o destino de seu

parceiro. Você também deve ler a documentação no site do parceiro vinculado na seção anterior e os padrões para esse parceiro listados nas seções a seguir.

Para transmitir a um destino de terceiros que não seja compatível com a configuração rápida, você pode seguir as instruções em Siga as instruções em Configuração personalizada com o Firehose para configurar um fluxo usando o Firehose. Em seguida, envie essas métricas do Firehose ao destino final.

Usar a configuração rápida de parceiros para criar um fluxo de métricas para um provedor externo

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), Streams (Fluxos). Em seguida, escolha Create metric stream (Criar fluxo de métrica).
3. (Opcional) Se estiver conectado a uma conta configurada como uma conta de monitoramento na observabilidade entre contas do CloudWatch, você poderá escolher se deseja incluir métricas de contas de origem vinculadas nesse fluxo de métricas. Para incluir métricas de contas de origem, escolha Include source account metrics (Incluir métricas da conta de origem).
4. Escolha Configuração rápida do parceiro da Amazon Web Services
5. Selecione o nome do parceiro para o qual você deseja transmitir as métricas.
6. Para um URL do endpoint, insira o URL de destino.
7. Para Chave de acesso ou Chave de API, insira a chave de acesso para o parceiro. Nem todos os parceiros precisam de uma chave de acesso.
8. Em Métricas a serem transmitidas, escolha Todas as métricas ou Selecionar métricas.

   Se você escolher Todas as métricas, todas as métricas dessa conta serão incluídas no fluxo.

   Avalie bem se deseja transmitir todas as métricas, pois quanto mais métricas forem transmitidas, maiores serão as cobranças do fluxo de métrica.

   Se você escolgher Selecionar métrica, faça o seguinte:

   - Para transmitir a maioria dos namespaces métricos, escolha Excluir e selecione os namespaces ou métricas a serem excluídos. Ao especificar um namespace em Excluir, você também pode selecionar algumas métricas específicas desse namespace para excluir. Se você optar por excluir um namespace, mas não selecionar métricas nesse namespace, todas as métricas desse namespace serão excluídas.
   - Para incluir apenas alguns namespaces de métrica ou métricas no fluxo de métricas, escolha Incluir e selecione os namespaces ou métricas que deseja incluir. Se você optar por

incluir um namespace, mas não selecionar métricas nesse namespace, todas as métricas desse namespace serão incluídas.

9. (Opcional) Para transmitir estatísticas adicionais para algumas dessas métricas além de Mínimo, Máximo, SampleCount e Soma, selecione Adicionar estatísticas adicionais. Escolha Add recommended metrics (Adicionar métricas recomendadas) para adicionar algumas estatísticas comumente usadas ou selecione manualmente o namespace e o nome da métrica para os quais deseja transmitir estatísticas adicionais. Em seguida, selecione as estatísticas adicionais que deseja transmitir.

   Em seguida, para escolher outro grupo de métricas para o qual transmitir um conjunto diferente de estatísticas adicionais, escolha Add additional statistics (Adicionar outras estatísticas). Cada métrica pode incluir até 20 estatísticas adicionais, e um fluxo de métrica pode ter até 100 métricas com estatísticas adicionais.

   A transmissão de estatísticas adicionais gera mais cobranças. Para ter mais informações, consulte Estatísticas que podem ser transmitidas.

   Para ver as definições das estatísticas adicionais, consulte Definições de estatísticas do CloudWatch.

10. (Opcional) Personalize o nome do novo fluxo de métricas em Metric stream name (Nome do fluxo de métricas).
11. Escolha Create metric stream (Criar filtro de métrica).

Padrões de fluxo do Datadog

Os fluxos de configuração rápida de parceiros para o Datadog usam os seguintes padrões:

- Formato de saída: OpenTelemetry 0.7.0
- Codificação de conteúdo de fluxo do Firehose GZIP
- Opções de buffer de fluxo do Firehose Intervalo de 60 segundos, tamanho de 4 MBs
- Opção de nova tentativa de fluxo do Firehose Duração de 60 segundos

Quando você usa a configuração rápida de parceiros para criar um fluxo de métricas para o Datadog e transmite determinadas métricas, por padrão essas métricas incluem algumas estatísticas adicionais. O streaming de estatísticas adicionais pode incorrer em cobranças adicionais. Para

obter mais informações sobre estatísticas e suas cobranças, consulte Estatísticas que podem ser transmitidas.

A lista a seguir mostra as métricas que têm estatísticas adicionais transmitidas por padrão, se você optar por transmitir essas métricas. Você pode optar por desmarcar essas estatísticas adicionais antes de iniciar a transmissão.

- **`Duration`** no **`AWS/Lambda`**: p50, p80, p95, p99, p99.9
- **`PostRuntimeExtensionDuration`** no **`AWS/Lambda`**: p50, p99
- **`FirstByteLatency`** e **`TotalRequestLatency`** no **`AWS/S3`**: p50, p90, p95, p99, p99.9
- **`ResponseLatency`** no **`AWS/Polly`** e **`TargetResponseTime`** no AWS/ApplicationELB: p50, p90, p95, p99
- **`Latency`** e **`IntegrationLatency`** na **`AWS/ApiGateway`**: p90, p95, p99
- **`Latency`** e **`TargetResponseTime`** no **`AWS/ELB`**: p95, p99
- **`RequestLatency`** no **`AWS/AppRunner`**: p50, p95, p99
- **`ActivityTime`**, **`ExecutionTime`**, **`LambdaFunctionRunTime`**, **`LambdaFunctionScheduleTime`**, **`LambdaFunctionTime`**, **`ActivityRunTime`**, e **`ActivityScheduleTime`** nos **`AWS/States`**: p95, p99
- **`EncoderBitRate`**, **`ConfiguredBitRate`**, e **`ConfiguredBitRateAvailable`** no **`AWS/MediaLive`**: p90
- **`Latency`** no **`AWS/AppSync`**: p90

Padrões de fluxo do Dynatrace

Os fluxos de configuração rápida de parceiros para o Dynatrace usam os seguintes padrões:

- Formato de saída: OpenTelemetry 0.7.0
- Codificação de conteúdo de fluxo do Firehose GZIP
- Opções de buffer de fluxo do Firehose Intervalo de 60 segundos, tamanho de 5 MBs
- Opção de nova tentativa de fluxo do Firehose Duração de 600 segundos

Padrões de fluxo da New Relic

Os fluxos de configuração rápida de parceiros para a New Relic usam os seguintes padrões:

- Formato de saída: OpenTelemetry 0.7.0
- Codificação de conteúdo de fluxo do Firehose GZIP
- Opções de buffer de fluxo do Firehose Intervalo de 60 segundos, tamanho de 1 MB
- Opção de nova tentativa de fluxo do Firehose Duração de 60 segundos

Padrões de fluxo do Splunk Observability Cloud

Os fluxos de configuração rápida de parceiros para o Splunk Observability Cloud usam os seguintes padrões:

- Formato de saída: OpenTelemetry 0.7.0
- Codificação de conteúdo de fluxo do Firehose GZIP
- Opções de buffer de fluxo do Firehose Intervalo de 60 segundos, tamanho de 1 MB
- Opção de nova tentativa de fluxo do Firehose Duração de 300 segundos

Padrões de fluxo do Sumo Logic

Os fluxos de configuração rápida de parceiros para a Sumo Logic usam os seguintes padrões:

- Formato de saída: OpenTelemetry 0.7.0
- Codificação de conteúdo de fluxo do Firehose GZIP
- Opções de buffer de fluxo do Firehose Intervalo de 60 segundos, tamanho de 1 MB
- Opção de nova tentativa de fluxo do Firehose Duração de 60 segundos

Estatísticas que podem ser transmitidas

Os fluxos de métrica sempre incluem as seguintes estatísticas: `Minimum`, `Maximum`, `SampleCount` e `Sum`. Você também pode optar por incluir as seguintes estatísticas adicionais em um fluxo de métrica. Essa escolha é feita por métrica. Para mais informações sobre estas estatísticas, consulte Definições de estatísticas do CloudWatch.

- Valores de percentil como p95 ou p99 (para fluxos com formato JSON ou OpenTelemetry)
- Média aparada (somente para fluxos com o formato JSON)
- Média winsorizada (somente para fluxos com o formato JSON)
- Contagem aparada (somente para fluxos com o formato JSON)

- Soma aparada (somente para fluxos com o formato JSON)
- Classificação de percentil (somente para fluxos com o formato JSON)
- Média interquartil (somente para fluxos com o formato JSON)

A transmissão de estatísticas adicionais gera cobranças adicionais. A transmissão de uma a cinco dessas estatísticas adicionais para uma determinada métrica é cobrada como uma atualização de métrica adicional. Depois disso, cada conjunto adicional de até cinco dessas estatísticas é cobrado como outra atualização métrica.

Por exemplo, suponha que você esteja transmitindo as seis estatísticas adicionais a seguir para uma métrica: p95, p99, p99,9, média aparada, média winsorizada e soma aparada. Cada atualização dessa métrica é cobrada como três atualizações de métrica: uma para a atualização de métrica que inclui as estatísticas padrão, uma para as primeiras cinco estatísticas adicionais e uma para a sexta estatística adicional. A adição de outras quatro estatísticas, totalizando dez, não aumentaria a cobrança, mas uma décima primeira estatística adicional aumentaria.

Quando você especifica uma combinação de nome da métrica e namespace para transmitir estatísticas adicionais, todas as combinações de dimensão desse nome da métrica e namespace são transmitidas com as estatísticas adicionais.

Os fluxos de métricas do CloudWatch publicam uma nova métrica, `TotalMetricUpdate`, que reflete o número básico de atualizações métricas mais atualizações adicionais de métricas incorridas pela transmissão de estatísticas adicionais. Para ter mais informações, consulte Monitorar seus fluxos de métrica com métricas do CloudWatch.

Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

> **Note**
>
> Algumas métricas não são compatíveis com percentis. As estatísticas de percentil para essas métricas são excluídas do fluxo e não geram cobranças de fluxo de métrica. Um exemplo dessas estatísticas que não são compatíveis com percentis são algumas métricas no namespace `AWS/ECS`.

As estatísticas adicionais que você configurar só são transmitidas se corresponderem aos filtros do fluxo. Por exemplo, se você criar um fluxo que tenha apenas `EC2` e `RDS` nos filtros de inclusão e, em

seguida, sua configuração de estatísticas listar `EC2` e `Lambda`, o fluxo inclui métricas do `EC2` com estatísticas adicionais, métricas do `RDS` apenas com as estatísticas padrão e não inclui nenhuma estatística do `Lambda`.

Operação e manutenção do fluxo de métricas

Os fluxos de métricas estão sempre em um destes dois estados: Running (Em execeução) ou Stopped (Interrompido).

- Running: o fluxo de métricas está funcionando corretamente. Pode não haver dados de métrica transmitidos ao destino por causa dos filtros no fluxo.
- Stopped: o fluxo de métrica foi explicitamente interrompido por alguém, e não por causa de um erro. Pode ser útil interromper o fluxo para pausar temporariamente o fluxo de dados sem excluir o fluxo.

Se você interromper e reiniciar um fluxo de métricas, os dados de métrica que foram publicados no CloudWatch enquanto o fluxo de métricas foi interrompido não serão preenchidos para o fluxo de métricas.

Se você alterar o formato de saída de um fluxo de métricas, em certos casos, poderá ver uma pequena quantidade de dados métricos gravados no destino no formato antigo e no novo formato. Para evitar essa situação, é possível criar um fluxo de entrega do Firehose com a mesma configuração que o atual, depois mudar para o novo fluxo de entrega do Firehose e alterar o formato de saída ao mesmo tempo. Dessa forma, os registros do Kinesis com formato de saída diferente são armazenados no Amazon S3, em objetos separados. Posteriormente, você poderá direcionar o tráfego de volta ao fluxo de entrega original do Firehose e excluir o segundo fluxo de entrega.

Para visualizar, editar, interromper e iniciar seus fluxos de métricas

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), Streams (Fluxos).

   A lista de fluxos é exibida, e a coluna Status exibe se cada fluxo está sendo executado ou interrompido.
3. Para interromper ou iniciar um fluxo de métrica, selecione o fluxo e escolha Stop (Interromper) ou Start (Iniciar).
4. Para ver os detalhes sobre um fluxo de métrica, selecione o fluxo e escolha View details (Visualizar detalhes).

5. Para alterar o formato de saída do fluxo, os filtros, o fluxo de destino do Firehose ou os perfis, escolha Editar e faça as alterações desejadas.

   Se você alterar os filtros, poderá haver algumas lacunas nos dados de métrica durante a transição.

Monitorar seus fluxos de métrica com métricas do CloudWatch

Os fluxos de métrica emitem métricas do CloudWatch sobre sua integridade e operação no namespace `AWS/CloudWatch/MetricStreams`. São emitidas as métricas a seguir. Essas métricas são emitidas com uma dimensão `MetricStreamName` e sem dimensão. É possível usar as métricas sem dimensões para ver métricas agregadas de todos os fluxos de métricas. Você pode usar as métricas com a dimensão `MetricStreamName` para ver as métricas sobre apenas esse fluxo de métricas.

Para todas essas métricas, os valores são emitidos apenas para fluxos de métricas que estão no estado Running (Em execução).

Métrica	Descrição
`MetricUpdate`	As atualizações de métricas numéricas enviadas à transmissão de métricas. Se nenhuma atualização de métrica for transmitida durante um período, essa métrica não será emitida durante esse período. Se você interromper o fluxo de métricas, essa métrica deixará de ser emitida até que o fluxo de métricas seja iniciado outra vez. Estatística válida: `Sum` Unidades: nenhuma
`TotalMetricUpdate`	Isso é calculado como MetricUpdate + um número baseado em estatísticas adicionais que estão sendo transmitidas. Para cada combinação exclusiva de namespace e nome da métrica, a transmissão de 1 a 5 estatísticas adicionais acrescenta 1 a `TotalMetricUpdate` , a transmissão de 6 a 10 estatísticas adicionais acrescenta 2 a `TotalMetricUpdate` , e assim por diante.

Métrica	Descrição
	Estatística válida: `Sum` Unidades: nenhuma
`PublishErrorRate`	O número de erros irrecuperáveis que ocorrem ao inserir dados no fluxo de entrega do Firehose. Se não ocorrer nengum erro durante um período, essa métrica não será emitida durante esse período. Se você interromper o fluxo de métricas, essa métrica deixará de ser emitida até que o fluxo de métricas seja iniciado outra vez. Estatística válida: `Average` para ver a taxa de atualizações métricas que não podem ser gravadas. Esse valor deverá ser entre 0.0 e 1.0. Unidades: nenhuma

Confiança entre o CloudWatch e o Firehose

O fluxo de entrega do Firehose deve confiar no CloudWatch por meio de um perfil do IAM que tenha permissões de gravação no Firehose. Essas permissões podem ser limitadas ao único fluxo de entrega do Firehose usado pelo fluxo de métricas do CloudWatch. A função do IAM deve confiar no serviço principal `streams.metrics.cloudwatch.amazonaws.com`.

Se usar o console do CloudWatch para criar um fluxo de métricas, você poderá fazer com que o CloudWatch crie a função com as permissões corretas. Caso use outro método para criar um fluxo de métricas ou queira criar a própria função do IAM, ele deverá conter a seguinte política de permissões e política de confiança.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:firehose:region:account-id:deliverystream/*"
```

```
        }
    ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "streams.metrics.cloudwatch.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Os dados de métrica são transmitidos pelo CloudWatch ao fluxo de entrega do Firehose de destino em nome da origem que contém o recurso de fluxo de métricas.

Formatos de saída de fluxos de métricas

Os dados em um fluxo de métricas do CloudWatch podem estar no formato JSON ou no formato OpenTelemetry. Atualmente, os formatos OpenTelemetry 1.0.0 e 0.7.0 são compatíveis.

Sumário

Formato JSON

Em um fluxo de métricas do CloudWatch que usa o formato JSON, cada registro do Firehose contém vários objetos JSON separados por um caractere de nova linha (\n). Cada objeto contém um único ponto de dados de uma única métrica.

O formato JSON usado é totalmente compatível com o AWS Glue e com o Amazon Athena. Se você tiver um fluxo de entrega do Firehose e uma tabela do AWS Glue formatada corretamente, o formato poderá ser automaticamente transformado em Parquet ou em colunar de linha otimizado (ORC) antes de ser armazenado no S3. Para obter mais informações sobre como transformar o formato, consulte Converting Your Input Record Format in Firehose. Para obter mais informações sobre o formatos correto para AWS Glue, consulte Que esquema do AWS Glue devo usar para o formato de saída JSON?.

No formato JSON, os valores válidos para `unit` são os mesmos que para o valor de `unit` na estrutura da API `MetricDatum`. Para obter mais informações, consulte MetricDatum. O valor do campo `timestamp` está em milissegundos epoch, como `1616004674229`.

O exemplo a seguir é do formato. Neste exemplo, o JSON está formatado para facilitar a leitura, mas, na prática, todo o formato está em uma única linha.

```
{
    "metric_stream_name": "MyMetricStream",
    "account_id": "1234567890",
    "region": "us-east-1",
    "namespace": "AWS/EC2",
    "metric_name": "DiskWriteOps",
    "dimensions": {
        "InstanceId": "i-123456789012"
    },
    "timestamp": 1611929698000,
    "value": {
        "count": 3.0,
        "sum": 20.0,
        "max": 18.0,
        "min": 0.0,
        "p99": 17.56,
        "p99.9": 17.8764,
        "TM(25%:75%)": 16.43
    },
    "unit": "Seconds"
```

```
}
```

Que esquema do AWS Glue devo usar para o formato de saída JSON?

Veja a seguir um exemplo de uma representação JSON do `StorageDescriptor` para uma tabela do AWS Glue, que seria usada pelo Firehose. Para obter mais informações sobre `StorageDescriptor`, consulte StorageDescriptor.

```
{
  "Columns": [
    {
      "Name": "metric_stream_name",
      "Type": "string"
    },
    {
      "Name": "account_id",
      "Type": "string"
    },
    {
      "Name": "region",
      "Type": "string"
    },
    {
      "Name": "namespace",
      "Type": "string"
    },
    {
      "Name": "metric_name",
      "Type": "string"
    },
    {
      "Name": "timestamp",
      "Type": "timestamp"
    },
    {
      "Name": "dimensions",
      "Type": "map<string,string>"
    },
    {
      "Name": "value",
      "Type":
 "struct<min:double,max:double,count:double,sum:double,p99:double,p99.9:double>"
    },
```

```
    {
      "Name": "unit",
      "Type": "string"
    }
  ],
  "Location": "s3://my-s3-bucket/",
  "InputFormat": "org.apache.hadoop.mapred.TextInputFormat",
  "OutputFormat": "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
  "SerdeInfo": {
    "SerializationLibrary": "org.apache.hive.hcatalog.data.JsonSerDe"
  },
  "Parameters": {
    "classification": "json"
  }
}
```

O exemplo anterior serve para dados gravados no Amazon S3 no formato JSON. Substitua os valores nos campos a seguir pelos valores indicados para armazenar os dados no formato Parquet ou no formato colunar de linha otimizado (ORC).

- Parquet:
  - inputFormat: org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat
  - outputFormat: org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat
  - SerDeInfo.serializationLib: org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe
  - parameters.classification: parquet
- ORC:
  - inputFormat: org.apache.hadoop.hive.ql.io.orc.OrcInputFormat
  - outputFormat: org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat
  - SerDeInfo.serializationLib: org.apache.hadoop.hive.ql.io.orc.OrcSerde
  - parameters.classification: orc

Formato OpenTelemetry 1.0.0

> **Note**
>
> Com o formato OpenTelemetry 1.0.0, os atributos de métricas são codificados como uma lista de objetos `KeyValue` em vez do tipo `StringKeyValue` usado no formato 0.7.0. Como consumidor, essa é a única grande mudança entre os formatos 0.7.0 e 1.0.0. Um analisador

gerado nos arquivos de protocolo do 0.7.0 não analisará atributos de métricas codificados no formato 1.0.0. O mesmo acontece ao contrário: um analisador gerado nos arquivos de protocolo do 1.0.0 não analisará atributos de métricas codificados no formato 0.7.0.

OpenTelemetry é uma coleção de ferramentas, APIs e SDKs. Pode ser usada para instrumentar, gerar, coletar e exportar dados de telemetria (métricas, logs e rastreamentos) para análise. O OpenTelemetry faz parte da Cloud Native Computing Foundation. Para obter mais informações, consulte OpenTelemetry.

Para obter informações sobre a especificação completa do OpenTelemetry 1.0.0, consulte Release version 1.0.0.

Um registro do Kinesis pode conter uma ou mais estruturas de dados `ExportMetricsServiceRequest` do OpenTelemetry. Cada estrutura de dados começa com um cabeçalho com `UnsignedVarInt32`indicando o tamanho do registro em bytes. Cada `ExportMetricsServiceRequest` pode conter dados de várias métricas ao mesmo tempo.

Veja a seguir uma representação de string da mensagem da estrutura de dados `ExportMetricsServiceRequest` do OpenTelemetry. O OpenTelemetry serializa o uso do protocolo binário do Google Protocol Buffers, e não é legível por humanos.

```
resource_metrics {
  resource {
    attributes {
      key: "cloud.provider"
      value {
        string_value: "aws"
      }
    }
    attributes {
      key: "cloud.account.id"
      value {
        string_value: "123456789012"
      }
    }
    attributes {
      key: "cloud.region"
      value {
        string_value: "us-east-1"
      }
```

```
    }
    attributes {
      key: "aws.exporter.arn"
      value {
        string_value: "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
MyMetricStream"
      }
    }
  }
  scope_metrics {
    metrics {
      name: "amazonaws.com/AWS/DynamoDB/ConsumedReadCapacityUnits"
      unit: "NoneTranslated"
      summary {
        data_points {
          start_time_unix_nano: 60000000000
          time_unix_nano: 120000000000
          count: 1
          sum: 1.0
          quantile_values {
            value: 1.0
          }
          quantile_values {
            quantile: 0.95
            value: 1.0
          }
          quantile_values {
            quantile: 0.99
            value: 1.0
          }
          quantile_values {
            quantile: 1.0
            value: 1.0
          }
          attributes {
            key: "Namespace"
            value {
              string_value: "AWS/DynamoDB"
            }
          }
          attributes {
            key: "MetricName"
            value {
              string_value: "ConsumedReadCapacityUnits"
```

```
            }
          }
          attributes {
            key: "Dimensions"
            value {
              kvlist_value {
                values {
                  key: "TableName"
                  value {
                    string_value: "MyTable"
                  }
                }
              }
            }
          }
        }
        data_points {
          start_time_unix_nano: 70000000000
          time_unix_nano: 130000000000
          count: 2
          sum: 5.0
          quantile_values {
            value: 2.0
          }
          quantile_values {
            quantile: 1.0
            value: 3.0
          }
          attributes {
            key: "Namespace"
            value {
              string_value: "AWS/DynamoDB"
            }
          }
          attributes {
            key: "MetricName"
            value {
              string_value: "ConsumedReadCapacityUnits"
            }
          }
          attributes {
            key: "Dimensions"
            value {
              kvlist_value {
```

```
                values {
                  key: "TableName"
                  value {
                    string_value: "MyTable"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Objeto de nível superior para serializar dados de métrica do OpenTelemetry

`ExportMetricsServiceRequest` é o wrapper de nível mais alto para serializar uma carga útil do exportador do OpenTelemetry. Contém um ou mais `ResourceMetrics`.

```
message ExportMetricsServiceRequest {
  // An array of ResourceMetrics.
  // For data coming from a single resource this array will typically contain one
  // element. Intermediary nodes (such as OpenTelemetry Collector) that receive
  // data from multiple origins typically batch the data before forwarding further and
  // in that case this array will contain multiple elements.
  repeated opentelemetry.proto.metrics.v1.ResourceMetrics resource_metrics = 1;
}
```

`ResourceMetrics` é o objeto de nível superior para representar objetos MetricData.

```
// A collection of ScopeMetrics from a Resource.
message ResourceMetrics {
  reserved 1000;

  // The resource for the metrics in this message.
  // If this field is not set then no resource info is known.
  opentelemetry.proto.resource.v1.Resource resource = 1;

  // A list of metrics that originate from a resource.
  repeated ScopeMetrics scope_metrics = 2;

  // This schema_url applies to the data in the "resource" field. It does not apply
```

```
  // to the data in the "scope_metrics" field which have their own schema_url field.
  string schema_url = 3;
}
```

O objeto Resource

`Resource` é um objeto de par de valor que contém algumas informações sobre o recurso que gerou as métricas. Para métricas criadas pela AWS, a estrutura de dados contém o nome do recurso da Amazon (ARN) do recurso relacionado à métrica, como uma instância do EC2 ou um bucket do S3.

O objeto `Resource` contém um atributo chamado `attributes`, que armazena uma lista de pares de chave-valor.

- `cloud.account.id` contém o ID da conta
- `cloud.region` contém a Região
- `aws.exporter.arn` contém o ARN do fluxo de métricas
- `cloud.provider` é sempre `aws`.

```
// Resource information.
message Resource {
  // Set of attributes that describe the resource.
  // Attribute keys MUST be unique (it is not allowed to have more than one
  // attribute with the same key).
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 1;

  // dropped_attributes_count is the number of dropped attributes. If the value is 0,
 then
  // no attributes were dropped.
  uint32 dropped_attributes_count = 2;
}
```

O objeto ScopeMetrics

O campo `scope` não será preenchido. Preencheremos apenas o campo de métricas que estamos exportando.

```
// A collection of Metrics produced by an Scope.
message ScopeMetrics {
  // The instrumentation scope information for the metrics in this message.
  // Semantically when InstrumentationScope isn't set, it is equivalent with
```

```
  // an empty instrumentation scope name (unknown).
  opentelemetry.proto.common.v1.InstrumentationScope scope = 1;

  // A list of metrics that originate from an instrumentation library.
  repeated Metric metrics = 2;

  // This schema_url applies to all metrics in the "metrics" field.
  string schema_url = 3;
}
```

O objeto Metric

O objeto de métricas contém alguns metadados e um campo de dados `Summary` que contém uma lista de `SummaryDataPoint`.

Para fluxos de métricas, os metadados são:

- `name` será `amazonaws.com/`*`metric_namespace`*`/`*`metric_name`*
- `description` ficará em branco
- `unit` será preenchido mapeando a unidade do dado métrico para a variante que distingue maiúsculas e minúsculas do código unificado para unidades de medida. Para obter mais informações, consulte Conversões com formato OpenTelemetry 1.0.0 e Código unificado para unidades de medida.
- `type` será `SUMMARY`

```
message Metric {
  reserved 4, 6, 8;

  // name of the metric, including its DNS name prefix. It must be unique.
  string name = 1;

  // description of the metric, which can be used in documentation.
  string description = 2;

  // unit in which the metric value is reported. Follows the format
  // described by http://unitsofmeasure.org/ucum.html.
  string unit = 3;

  // Data determines the aggregation type (if any) of the metric, what is the
  // reported value type for the data points, as well as the relatationship to
```

```
  // the time interval over which they are reported.
  oneof data {
    Gauge gauge = 5;
    Sum sum = 7;
    Histogram histogram = 9;
    ExponentialHistogram exponential_histogram = 10;
    Summary summary = 11;
  }
}

message Summary {
  repeated SummaryDataPoint data_points = 1;
}
```

O objeto SummaryDataPoint

O objeto SummaryDataPoint contém o valor de um único ponto de dados em uma série temporal em uma métrica DoubleSummary.

```
// SummaryDataPoint is a single data point in a timeseries that describes the
// time-varying values of a Summary metric.
message SummaryDataPoint {
  reserved 1;

  // The set of key/value pairs that uniquely identify the timeseries from
  // where this point belongs. The list may be empty (may contain 0 elements).
  // Attribute keys MUST be unique (it is not allowed to have more than one
  // attribute with the same key).
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 7;

  // StartTimeUnixNano is optional but strongly encouraged, see the
  // the detailed comments above Metric.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 start_time_unix_nano = 2;

  // TimeUnixNano is required, see the detailed comments above Metric.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 time_unix_nano = 3;
```

```
  // count is the number of values in the population. Must be non-negative.
  fixed64 count = 4;

  // sum of the values in the population. If count is zero then this field
  // must be zero.
  //
  // Note: Sum should only be filled out when measuring non-negative discrete
  // events, and is assumed to be monotonic over the values of these events.
  // Negative events *can* be recorded, but sum should not be filled out when
  // doing so.  This is specifically to enforce compatibility w/ OpenMetrics,
  // see: https://github.com/OpenObservability/OpenMetrics/blob/main/specification/
OpenMetrics.md#summary
  double sum = 5;

  // Represents the value at a given quantile of a distribution.
  //
  // To record Min and Max values following conventions are used:
  // - The 1.0 quantile is equivalent to the maximum value observed.
  // - The 0.0 quantile is equivalent to the minimum value observed.
  //
  // See the following issue for more context:
  // https://github.com/open-telemetry/opentelemetry-proto/issues/125
  message ValueAtQuantile {
    // The quantile of a distribution. Must be in the interval
    // [0.0, 1.0].
    double quantile = 1;

    // The value at the given quantile of a distribution.
    //
    // Quantile values must NOT be negative.
    double value = 2;
  }

  // (Optional) list of values at different quantiles of the distribution calculated
  // from the current snapshot. The quantiles must be strictly increasing.
  repeated ValueAtQuantile quantile_values = 6;

  // Flags that apply to this specific data point.  See DataPointFlags
  // for the available flags and their meaning.
  uint32 flags = 8;
}
```

Para ter mais informações, consulte Conversões com formato OpenTelemetry 1.0.0.

Conversões com formato OpenTelemetry 1.0.0

O CloudWatch executa algumas transformações para colocar os dados do CloudWatch no formato OpenTelemetry.

Converter namespace, nome da métrica e dimensões

Esses atributos são pares chave-valor codificados no mapeamento.

- Um atributo tem a chave `Namespace` e seu valor é o namespace da métrica
- Um atributo tem a chave `MetricName` e seu valor é o nome da métrica
- Um par tem a chave `Dimensions` e seu valor é uma lista aninhada de pares de chave/valor. Cada par nessa lista é mapeado para uma dimensão de métrica do CloudWatch, em que a chave do par é o nome da dimensão e seu valor é o valor da dimensão.

Converter Average, Sum, SampleCount, Min e Max

O ponto de dados Summary permite que o CloudWatch exporte todas essas estatísticas usando um ponto de dados.

- `startTimeUnixNano` contém o `startTime` do CloudWatch
- `timeUnixNano` contém o `endTime` do CloudWatch
- `sum` contém a estatística Sum.
- `count` contém a estatística SampleCount.
- `quantile_values` contém dois objetos `valueAtQuantile.value`:
  - `valueAtQuantile.quantile = 0.0` com `valueAtQuantile.value = Min value`
  - `valueAtQuantile.quantile = 0.99` com `valueAtQuantile.value = p99 value`
  - `valueAtQuantile.quantile = 0.999` com `valueAtQuantile.value = p99.9 value`
  - `valueAtQuantile.quantile = 1.0` com `valueAtQuantile.value = Max value`

Os recursos que consomem o fluxo de métrica podem calcular a estatística Average como Sum/SampleCount.

Converter unidades

As unidades do CloudWatch são mapeadas para a variante que diferencia maiúsculas de minúsculas do Código unificado para unidades de medida, conforme exibido na tabela a seguir. Para obter mais informações, consulte Código unificado para unidades de medida.

CloudWatch	OpenTelemetry
Segundo	s
Segundo ou segundos	s
Microssegundos	us
Milissegundos	ms
Bytes	By
Kilobytes	kBy
Megabytes	MBy
Gigabytes	GBy
Terabytes	TBy
Bits	bit
Kilobits	kbit
Megabits	MBit
Gigabits	Gbit
Terabits	Tbit
Percentual	%
Contagem	{Count}
Nenhum	1

As unidades que são combinadas com uma barra são mapeadas aplicando-se a conversão do OpenTelemetry de ambas as unidades. Por exemplo, bytes/segundo é mapeado para by/s.

Como analisar mensagens do OpenTelemetry 1.0.0

Esta seção fornece informações para ajudar você a começar a analisar o OpenTelemetry 1.0.0.

Primeiro, é necessário obter associações específicas de idioma que permitem analisar mensagens do OpenTelemetry 1.0.0 no idioma de sua preferência.

Para obter associações específicas de idioma

- As etapas dependem do idioma de sua preferência.
  - Para usar o Java, adicione a seguinte dependência Maven ao projeto Java: OpenTelemetry Java >> 0.14.1.
  - Para utilizar qualquer outro idioma, siga estas etapas:

    a. Verifique se seu idioma é compatível, conferindo a lista em Gerar suas classes.

    b. Instale o compilador Protobuf seguindo as etapas em Baixar buffers de protocolo.

    c. Baixe as definições do OpenTelemetry 0.7.0 ProtoBuf em Release version 1.0.0.

    d. Confirme se você está na pasta-raiz das definições do OpenTelemetry 0.7.0 ProtoBuf. Depois, crie uma pasta `src` e execute o comando para gerar ligações específicas do idioma. Para obter mais informações, consulte Gerar suas classes.

       Veja a seguir um exemplo de como gerar associações Javascript.

```
protoc --proto_path=./ --js_out=import_style=commonjs,binary:src \
opentelemetry/proto/common/v1/common.proto \
opentelemetry/proto/resource/v1/resource.proto \
opentelemetry/proto/metrics/v1/metrics.proto \
opentelemetry/proto/collector/metrics/v1/metrics_service.proto
```

A seção a seguir contém exemplos de como usar as vinculações específicas de idioma que você pode criar usando as instruções anteriores.

Java

```
package com.example;
```

```
import io.opentelemetry.proto.collector.metrics.v1.ExportMetricsServiceRequest;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public class MyOpenTelemetryParser {

    public List<ExportMetricsServiceRequest> parse(InputStream inputStream) throws
 IOException {
        List<ExportMetricsServiceRequest> result = new ArrayList<>();

        ExportMetricsServiceRequest request;
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        while ((request =
 ExportMetricsServiceRequest.parseDelimitedFrom(inputStream)) != null) {
            // Do whatever we want with the parsed message
            result.add(request);
        }

        return result;
    }
}
```

Javascript

Este exemplo pressupõe que a pasta raiz com as ligações geradas seja `./`

O argumento de dados da função `parseRecord` pode ser de um destes tipos:

- `Uint8Array` é o ideal
- `Buffer` ideal no nó
- `Array.number` inteiros de 8 bits

```
const pb = require('google-protobuf')
```

```
const pbMetrics =
    require('./opentelemetry/proto/collector/metrics/v1/metrics_service_pb')

function parseRecord(data) {
    const result = []

    // Loop until we've read all the data from the buffer
    while (data.length) {
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        const reader = new pb.BinaryReader(data)
        const messageLength = reader.decoder_.readUnsignedVarint32()
        const messageFrom = reader.decoder_.cursor_
        const messageTo = messageFrom + messageLength

        // Extract the current `ExportMetricsServiceRequest` message to parse
        const message = data.subarray(messageFrom, messageTo)

        // Parse the current message using the ProtoBuf library
        const parsed =
            pbMetrics.ExportMetricsServiceRequest.deserializeBinary(message)

        // Do whatever we want with the parsed message
        result.push(parsed.toObject())

        // Shrink the remaining buffer, removing the already parsed data
        data = data.subarray(messageTo)
    }

    return result
}
```

Python

É necessário ler os delimitadores `var-int` você mesmo ou usar os métodos internos `_VarintBytes(size)` e `_DecodeVarint32(buffer, position)`. Estes retornam a posição no buffer logo após os bytes de tamanho. O lado de leitura constrói um novo buffer que está limitado a ler apenas os bytes da mensagem.

```
size = my_metric.ByteSize()
f.write(_VarintBytes(size))
f.write(my_metric.SerializeToString())
msg_len, new_pos = _DecodeVarint32(buf, 0)
msg_buf = buf[new_pos:new_pos+msg_len]
request = metrics_service_pb.ExportMetricsServiceRequest()
request.ParseFromString(msg_buf)
```

Go

Use `Buffer.DecodeMessage()`.

C#

Use `CodedInputStream`. Essa classe é capaz de ler mensagens delimitadas por tamanho.

C++

As funções descritas em `google/protobuf/util/delimited_message_util.h` podem ler mensagens delimitadas por tamanho.

Outras linguagens

Para outros idiomas, consulte Baixar buffers de protocolo.

Ao implementar o analisador, considere que um registro do Kinesis pode conter vários mensagens de buffers de protocolo `ExportMetricsServiceRequest`, todas delas começando com um cabeçalho contendo `UnsignedVarInt32`, que indica o tamanho do registro em bytes.

Formato OpenTelemetry 0.7.0

OpenTelemetry é uma coleção de ferramentas, APIs e SDKs. Pode ser usada para instrumentar, gerar, coletar e exportar dados de telemetria (métricas, logs e rastreamentos) para análise. O OpenTelemetry faz parte da Cloud Native Computing Foundation. Para obter mais informações, consulte OpenTelemetry.

Para obter informações sobre a especificação completa do OpenTelemetry 0.7.0, consulte a versão v0.7.0.

Um registro do Kinesis pode conter uma ou mais estruturas de dados `ExportMetricsServiceRequest` do OpenTelemetry. Cada estrutura de dados começa

com um cabeçalho com `UnsignedVarInt32`indicando o tamanho do registro em bytes. Cada `ExportMetricsServiceRequest` pode conter dados de várias métricas ao mesmo tempo.

Veja a seguir uma representação de string da mensagem da estrutura de dados `ExportMetricsServiceRequest` do OpenTelemetry. O OpenTelemetry serializa o uso do protocolo binário do Google Protocol Buffers, e não é legível por humanos.

```
resource_metrics {
  resource {
    attributes {
      key: "cloud.provider"
      value {
        string_value: "aws"
      }
    }
    attributes {
      key: "cloud.account.id"
      value {
        string_value: "2345678901"
      }
    }
    attributes {
      key: "cloud.region"
      value {
        string_value: "us-east-1"
      }
    }
    attributes {
      key: "aws.exporter.arn"
      value {
        string_value: "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
MyMetricStream"
      }
    }
  }
  instrumentation_library_metrics {
    metrics {
      name: "amazonaws.com/AWS/DynamoDB/ConsumedReadCapacityUnits"
      unit: "1"
      double_summary {
        data_points {
          labels {
            key: "Namespace"
```

```
          value: "AWS/DynamoDB"
        }
        labels {
          key: "MetricName"
          value: "ConsumedReadCapacityUnits"
        }
        labels {
          key: "TableName"
          value: "MyTable"
        }
        start_time_unix_nano: 1604948400000000000
        time_unix_nano: 1604948460000000000
        count: 1
        sum: 1.0
        quantile_values {
          quantile: 0.0
          value: 1.0
        }
        quantile_values {
          quantile: 0.95
          value: 1.0
        }
        quantile_values {
          quantile: 0.99
          value: 1.0
        }
        quantile_values {
          quantile: 1.0
          value: 1.0
        }
      }
      data_points {
        labels {
          key: "Namespace"
          value: "AWS/DynamoDB"
        }
        labels {
          key: "MetricName"
          value: "ConsumedReadCapacityUnits"
        }
        labels {
          key: "TableName"
          value: "MyTable"
        }
```

```
          start_time_unix_nano: 1604948460000000000
          time_unix_nano: 1604948520000000000
          count: 2
          sum: 5.0
          quantile_values {
            quantile: 0.0
            value: 2.0
          }
          quantile_values {
            quantile: 1.0
            value: 3.0
          }
        }
      }
    }
  }
}
```

Objeto de nível superior para serializar dados de métrica do OpenTelemetry

`ExportMetricsServiceRequest` é o wrapper de nível mais alto para serializar uma carga útil do exportador do OpenTelemetry. Contém um ou mais `ResourceMetrics`.

```
message ExportMetricsServiceRequest {
  // An array of ResourceMetrics.
  // For data coming from a single resource this array will typically contain one
  // element. Intermediary nodes (such as OpenTelemetry Collector) that receive
  // data from multiple origins typically batch the data before forwarding further and
  // in that case this array will contain multiple elements.
  repeated opentelemetry.proto.metrics.v1.ResourceMetrics resource_metrics = 1;
}
```

`ResourceMetrics` é o objeto de nível superior para representar objetos MetricData.

```
// A collection of InstrumentationLibraryMetrics from a Resource.
message ResourceMetrics {
  // The resource for the metrics in this message.
  // If this field is not set then no resource info is known.
  opentelemetry.proto.resource.v1.Resource resource = 1;

  // A list of metrics that originate from a resource.
  repeated InstrumentationLibraryMetrics instrumentation_library_metrics = 2;
```

```
}
```

O objeto Resource

`Resource` é um objeto de par de valor que contém algumas informações sobre o recurso que gerou as métricas. Para métricas criadas pela AWS, a estrutura de dados contém o nome do recurso da Amazon (ARN) do recurso relacionado à métrica, como uma instância do EC2 ou um bucket do S3.

O objeto `Resource` contém um atributo chamado `attributes`, que armazena uma lista de pares de chave-valor.

- `cloud.account.id` contém o ID da conta
- `cloud.region` contém a Região
- `aws.exporter.arn` contém o ARN do fluxo de métricas
- `cloud.provider` é sempre `aws`.

```
// Resource information.
message Resource {
  // Set of labels that describe the resource.
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 1;

  // dropped_attributes_count is the number of dropped attributes. If the value is 0,
  // no attributes were dropped.
  uint32 dropped_attributes_count = 2;
}
```

O objeto InstrumentationLibraryMetrics

O campo instrumentation_library não será preenchido. Preencheremos apenas o campo de métricas que estamos exportando.

```
// A collection of Metrics produced by an InstrumentationLibrary.
message InstrumentationLibraryMetrics {
  // The instrumentation library information for the metrics in this message.
  // If this field is not set then no library info is known.
  opentelemetry.proto.common.v1.InstrumentationLibrary instrumentation_library = 1;
  // A list of metrics that originate from an instrumentation library.
  repeated Metric metrics = 2;
}
```

O objeto Metric

O objeto de métrica contém um campo de dados `DoubleSummary` que contém uma lista de `DoubleSummaryDataPoint`.

```
message Metric {
  // name of the metric, including its DNS name prefix. It must be unique.
  string name = 1;

  // description of the metric, which can be used in documentation.
  string description = 2;

  // unit in which the metric value is reported. Follows the format
  // described by http://unitsofmeasure.org/ucum.html.
  string unit = 3;

  oneof data {
    IntGauge int_gauge = 4;
    DoubleGauge double_gauge = 5;
    IntSum int_sum = 6;
    DoubleSum double_sum = 7;
    IntHistogram int_histogram = 8;
    DoubleHistogram double_histogram = 9;
    DoubleSummary double_summary = 11;
  }
}

message DoubleSummary {
  repeated DoubleSummaryDataPoint data_points = 1;
}
```

O objeto MetricDescriptor

O objeto MetricDescriptor contém metadados. Para obter mais informações, consulte o metrics.proto no GitHub.

Para fluxos de métricas, o MetricDescriptor tem o seguinte conteúdo:

- O `name` será `amazonaws.com/`*`metric_namespace`*`/`*`metric_name`*
- `description` ficará em branco.
- `unit` será preenchido mapeando a unidade do dado métrico para a variante que distingue maiúsculas e minúsculas do código unificado para unidades de medida. Para obter mais

informações, consulte Conversões com formato OpenTelemetry 0.7.0 e Código unificado para unidades de medida.

- O `type` será `SUMMARY`.

O objeto DoubleSummaryDataPoint

O objeto DoubleSummaryDataPoint contém o valor de um único ponto de dados em uma série temporal em uma métrica DoubleSummary.

```
// DoubleSummaryDataPoint is a single data point in a timeseries that describes the
// time-varying values of a Summary metric.
message DoubleSummaryDataPoint {
  // The set of labels that uniquely identify this timeseries.
  repeated opentelemetry.proto.common.v1.StringKeyValue labels = 1;

  // start_time_unix_nano is the last time when the aggregation value was reset
  // to "zero". For some metric types this is ignored, see data types for more
  // details.
  //
  // The aggregation value is over the time interval (start_time_unix_nano,
  // time_unix_nano].
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  //
  // Value of 0 indicates that the timestamp is unspecified. In that case the
  // timestamp may be decided by the backend.
  fixed64 start_time_unix_nano = 2;

  // time_unix_nano is the moment when this aggregation value was reported.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 time_unix_nano = 3;

  // count is the number of values in the population. Must be non-negative.
  fixed64 count = 4;

  // sum of the values in the population. If count is zero then this field
  // must be zero.
  double sum = 5;
```

```
  // Represents the value at a given quantile of a distribution.
  //
  // To record Min and Max values following conventions are used:
  // - The 1.0 quantile is equivalent to the maximum value observed.
  // - The 0.0 quantile is equivalent to the minimum value observed.
  message ValueAtQuantile {
    // The quantile of a distribution. Must be in the interval
    // [0.0, 1.0].
    double quantile = 1;

    // The value at the given quantile of a distribution.
    double value = 2;
  }

  // (Optional) list of values at different quantiles of the distribution calculated
  // from the current snapshot. The quantiles must be strictly increasing.
  repeated ValueAtQuantile quantile_values = 6;
}
```

Para ter mais informações, consulte Conversões com formato OpenTelemetry 0.7.0.

Conversões com formato OpenTelemetry 0.7.0

O CloudWatch executa algumas transformações para colocar os dados do CloudWatch no formato OpenTelemetry.

Converter namespace, nome da métrica e dimensões

Esses atributos são pares chave-valor codificados no mapeamento.

- Um par contém o namespace da métrica
- Um par contém o nome da métrica
- Para cada dimensão, o CloudWatch armazena o seguinte par: `metricDatum.Dimensions[i].`*`Name`*`, metricDatum.Dimensions[i].`*`Value`*

Converter Average, Sum, SampleCount, Min e Max

O ponto de dados Summary permite que o CloudWatch exporte todas essas estatísticas usando um ponto de dados.

- `startTimeUnixNano` contém o `startTime` do CloudWatch
- `timeUnixNano` contém o `endTime` do CloudWatch

- `sum` contém a estatística Sum.
- `count` contém a estatística SampleCount.
- `quantile_values` contém dois objetos `valueAtQuantile.value`:
  - `valueAtQuantile.quantile = 0.0` com `valueAtQuantile.value = Min value`
  - `valueAtQuantile.quantile = 0.99` com `valueAtQuantile.value = p99 value`
  - `valueAtQuantile.quantile = 0.999` com `valueAtQuantile.value = p99.9 value`
  - `valueAtQuantile.quantile = 1.0` com `valueAtQuantile.value = Max value`

Os recursos que consomem o fluxo de métrica podem calcular a estatística Average como Sum/SampleCount.

Converter unidades

As unidades do CloudWatch são mapeadas para a variante que diferencia maiúsculas de minúsculas do Código unificado para unidades de medida, conforme exibido na tabela a seguir. Para obter mais informações, consulte Código unificado para unidades de medida.

CloudWatch	OpenTelemetry
Segundo	s
Segundo ou segundos	s
Microsecond	us
Milissegundos	ms
Bytes	By
Kilobytes	kBy
Megabytes	MBy
Gigabytes	GBy
Terabytes	TBy
Bits	bit

CloudWatch	OpenTelemetry
Kilobits	kbit
Megabits	MBit
Gigabits	Gbit
Terabits	Tbit
Percentual	%
Contagem	{Count}
Nenhum	1

As unidades que são combinadas com uma barra são mapeadas aplicando-se a conversão do OpenTelemetry de ambas as unidades. Por exemplo, bytes/segundo é mapeado para by/s.

Como analisar mensagens OpenTelemetry 0.7.0

Esta seção fornece informações para ajudar você a começar a analisar o OpenTelemetry 0.7.0.

Primeiro, é necessário obter associações específicas de idioma, que permitem analisar mensagens do OpenTelemetry 0.7.0 no idioma de sua preferência.

Para obter associações específicas de idioma

- As etapas dependem do idioma de sua preferência.
  - Para usar o Java, adicione a seguinte dependência Maven ao projeto Java: OpenTelemetry Java >> 0.14.1.
  - Para utilizar qualquer outro idioma, siga estas etapas:

    a. Verifique se seu idioma é compatível, conferindo a lista em Gerar suas classes.

    b. Instale o compilador Protobuf seguindo as etapas em Baixar buffers de protocolo.

    c. Baixe as definições do OpenTelemetry 0.7.0 ProtoBuf em Release v0.7.0.

    d. Confirme se você está na pasta-raiz das definições do OpenTelemetry 0.7.0 ProtoBuf. Depois, crie uma pasta `src` e execute o comando para gerar ligações específicas do idioma. Para obter mais informações, consulte Gerar suas classes.

Veja a seguir um exemplo de como gerar associações Javascript.

```
protoc --proto_path=./ --js_out=import_style=commonjs,binary:src \
opentelemetry/proto/common/v1/common.proto \
opentelemetry/proto/resource/v1/resource.proto \
opentelemetry/proto/metrics/v1/metrics.proto \
opentelemetry/proto/collector/metrics/v1/metrics_service.proto
```

A seção a seguir contém exemplos de como usar as vinculações específicas de idioma que você pode criar usando as instruções anteriores.

Java

```
package com.example;

import io.opentelemetry.proto.collector.metrics.v1.ExportMetricsServiceRequest;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public class MyOpenTelemetryParser {

    public List<ExportMetricsServiceRequest> parse(InputStream inputStream) throws
 IOException {
        List<ExportMetricsServiceRequest> result = new ArrayList<>();

        ExportMetricsServiceRequest request;
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        while ((request =
 ExportMetricsServiceRequest.parseDelimitedFrom(inputStream)) != null) {
            // Do whatever we want with the parsed message
            result.add(request);
        }
```

```
        return result;
    }
}
```

Javascript

Este exemplo pressupõe que a pasta raiz com as ligações geradas seja `./`

O argumento de dados da função `parseRecord` pode ser de um destes tipos:

- `Uint8Array` é o ideal
- `Buffer` ideal no nó
- `Array.number` inteiros de 8 bits

```
const pb = require('google-protobuf')
const pbMetrics =
    require('./opentelemetry/proto/collector/metrics/v1/metrics_service_pb')

function parseRecord(data) {
    const result = []

    // Loop until we've read all the data from the buffer
    while (data.length) {
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        const reader = new pb.BinaryReader(data)
        const messageLength = reader.decoder_.readUnsignedVarint32()
        const messageFrom = reader.decoder_.cursor_
        const messageTo = messageFrom + messageLength

        // Extract the current `ExportMetricsServiceRequest` message to parse
        const message = data.subarray(messageFrom, messageTo)

        // Parse the current message using the ProtoBuf library
        const parsed =
            pbMetrics.ExportMetricsServiceRequest.deserializeBinary(message)
```

```
        // Do whatever we want with the parsed message
        result.push(parsed.toObject())

        // Shrink the remaining buffer, removing the already parsed data
        data = data.subarray(messageTo)
    }

    return result
}
```

Python

É necessário ler os delimitadores `var-int` você mesmo ou usar os métodos internos `_VarintBytes(size)` e `_DecodeVarint32(buffer, position)`. Estes retornam a posição no buffer logo após os bytes de tamanho. O lado de leitura constrói um novo buffer que está limitado a ler apenas os bytes da mensagem.

```
size = my_metric.ByteSize()
f.write(_VarintBytes(size))
f.write(my_metric.SerializeToString())
msg_len, new_pos = _DecodeVarint32(buf, 0)
msg_buf = buf[new_pos:new_pos+msg_len]
request = metrics_service_pb.ExportMetricsServiceRequest()
request.ParseFromString(msg_buf)
```

Go

Use `Buffer.DecodeMessage()`.

C#

Use `CodedInputStream`. Essa classe é capaz de ler mensagens delimitadas por tamanho.

C++

As funções descritas em `google/protobuf/util/delimited_message_util.h` podem ler mensagens delimitadas por tamanho.

Outras linguagens

Para outros idiomas, consulte Baixar buffers de protocolo.

Ao implementar o analisador, considere que um registro do Kinesis pode conter vários mensagens de buffers de protocolo `ExportMetricsServiceRequest`, todas delas começando com um cabeçalho contendo `UnsignedVarInt32`, que indica o tamanho do registro em bytes.

Solução de problemas

Caso não esteja vendo dados de métrica no destino final, confira o seguinte:

- Confira se o fluxo de métricas está no estado de execução. Para obter etapas sobre como usar o console do CloudWatch para isso, consulte Operação e manutenção do fluxo de métricas.
- Métricas publicadas há mais de dois dias não são transmitidas. Para determinar se uma determinada métrica será transmitida, exiba-a em um gráfico no console do CloudWatch e verifique há quanto tempo o último ponto de dados visível ocorreu. Se isso tiver ocorrido há mais de dois dias, ela não será coletada pelas transmissões de métricas.
- Confira as métricas emitidas pelo fluxo de métricas. No console do CloudWatch, em Metrics (Métricas), consulte o namespace AWS/CloudWatch/MetricStreams para as métricas MetricUpdate, TotalMetricUpdate e PublishErrorRate.
- Se a métrica PublishErrorRate for alta, confirme se o destino usado pelo fluxo de entrega do Firehose existe e se o perfil do IAM especificado na configuração do fluxo de métricas concede à entidade principal do serviço do `CloudWatch` permissões para gravar nele. Para ter mais informações, consulte Confiança entre o CloudWatch e o Firehose.
- Verifique se o fluxo de entrega do Firehose tem permissão para gravar no destino final.
- No console do Firehose, exiba o fluxo de entrega do Firehose usado para o fluxo de métricas e verifique a aba Monitoramento para ver se o fluxo de entrega do Firehose está recebendo dados.
- Confirme se configurou o fluxo de entrega do Firehose com os detalhes corretos.
- Confira todos os logs ou métricas disponíveis do destino final de gravação do fluxo de entrega do Firehose.
- Para obter informações mais detalhadas, habilite o registro de erros em log do CloudWatch Logs no fluxo de entrega do Firehose. Para obter mais informações, consulte Monitorar o Amazon Data Firehose usando o CloudWatch Logs.

Visualizar métricas disponíveis

As métricas são agrupadas primeiro pelo namespace e, em seguida, por várias combinações de dimensão dentro de cada namespace. Por exemplo, você pode visualizar todas as métricas do EC2,

as métricas do EC2 agrupadas por instância ou as métricas do EC2 agrupadas por grupo do Auto Scaling.

Somente os produtos da AWS que você está usando enviam métricas ao Amazon CloudWatch .

Para obter uma lista de produtos da AWS que enviam métricas ao CloudWatch, consulte Produtos da AWS que publicam métricas do CloudWatch. Nessa página, também é possível ver as métricas e as dimensões publicadas por cada um desses serviços.

> **Note**
>
> AS métricas que não tiverem novos pontos de dados nas últimas duas semanas não serão exibidas no console. Elas também não serão exibidas quando você digitar o nome da métrica ou os nomes de dimensão na caixa de pesquisa na guia Todas as métricas do console e não serão retornadas nos resultados de um comando list-metrics. A melhor maneira de recuperar essas métricas é com os comandos get-metric-data ou get-metric-statistics na AWS CLI.
> Se a métrica antiga que você deseja visualizar tiver uma métrica atual com dimensões semelhantes, você poderá visualizar essa métrica semelhante atual e depois escolher a guia Origem e alterar o nome da métrica e os campos de dimensão para os que deseja e também alterar o intervalo de tempo para um momento em que a métrica estava sendo relatada.

As etapas a seguir ajudam você a navegar pelos namespaces de métricas para encontrar e visualizar métricas. Você também pode pesquisar por métricas usando termos de pesquisa específicos. Para ter mais informações, consulte Procurar por métricas disponíveis.

Se você estiver procurando em uma conta configurada como uma conta de monitoramrento na observabilidade entre contas do CloudWatch, poderá visualizar as métricas das contas de origem vinculadas a essa conta de monitoramento. Quando as métricas das contas de origem são exibidas, o ID ou o rótulo da conta de origem também é exibido. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

Para visualizar as métricas disponíveis por namespace e dimensão usando o console

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), All metrics (Todas as métricas).
3. Selecione um namespace de métrica (por exemplo, EC2 ou Lambda).
4. Selecione uma dimensão de métrica, por exemplo, Per-Instance Metrics (Métricas por instância) ou By Function Name (Por nome de perfil).

5. A guia Browse (Procurar) exibe todas as métricas dessa dimensão no namespace. Ao lado de cada nome de métrica há um botão de informações que você pode escolher para visualizar um pop-up com a definição da métrica.

   Se essa for uma conta de monitoramento na observabilidade entre contas do CloudWatch, você poderá ver as métricas das contas de origem vinculadas e essa conta de monitoramento. As colunas Account label (Rótulo da conta) e Account id (ID da conta) na tabela mostram qual é a conta de origem de cada métrica.

   Você pode fazer o seguinte:

   a. Para classificar a tabela, use o cabeçalho da coluna.
   b. Para criar um gráfico de uma métrica, marque a caixa de seleção ao lado da métrica. Para selecionar todas as métricas, marque a caixa de seleção na linha de cabeçalho da tabela.
   c. Para filtrar por conta, escolha o rótulo da conta ou o ID da conta e escolha Add to search (Adicionar à pesquisa).
   d. Para filtrar por recurso, escolha o ID do recurso e, em seguida, escolha Adicionar à pesquisa.
   e. Para filtrar por métrica, selecione o nome da métrica e, em seguida, escolha Adicionar à pesquisa.
6. (Opcional) Para adicionar esse gráfico a um painel do CloudWatch, escolha Actions (Ações), Add to dashboard (Adicionar ao painel).

Para visualizar métricas disponíveis por namespace da conta, dimensão ou métrica usando a AWS CLI

Use o comando list-metrics para listar as métricas do CloudWatch. Para obter uma lista de namespaces, métricas e dimensões para todos os serviços que publicam métricas, consulte Produtos da AWS que publicam métricas do CloudWatch.

O exemplo de comando a seguir lista todas as métricas do Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

A seguir, um exemplo de saída.

```
{
```

```
  "Metrics" : [
    ...
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkOut"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "CPUUtilization"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkIn"
    },
    ...
  ]
}
```

Para relacionar todas as métricas disponíveis para um recurso especificado

O exemplo a seguir especifica o namespace `AWS/EC2` e a dimensão `InstanceId` para visualizar os resultados somente para a instância especificada.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
 Name=InstanceId,Value=i-1234567890abcdef0
```

Para relacionar uma métrica para todos os recursos

O exemplo a seguir especifica o namespace `AWS/EC2` e o nome de uma métrica para visualizar os resultados somente para a métrica especificada.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

Para recuperar métricas das contas de origem vinculadas na observabilidade entre contas do CloudWatch

O exemplo a seguir é executado em uma conta de monitoramento para recuperar as métricas da conta de monitoramento e de todas as contas de origem vinculadas. Se você não adicionar `--include-linked-accounts`, o comando só retornará as métricas da conta de monitoramento.

```
aws cloudwatch list-metrics --include-linked-accounts
```

Para recuperar métricas de uma conta de origem na observabilidade entre contas do CloudWatch

O exemplo a seguir é executado em uma conta de monitoramento para recuperar as métricas da conta de origem com o ID 111122223333.

```
aws cloudwatch list-metrics --include-linked-accounts --owning-account "111122223333"
```

Procurar por métricas disponíveis

Você pode pesquisar em todas as métricas da sua conta usando termos de pesquisa direcionados. Serão retornadas as métricas com resultados coincidentes no namespace, nome ou dimensões.

Se essa for uma conta de monitoramento na observabilidade entre contas do CloudWatch, você também pesquisará métricas das contas de origem vinculadas a essa conta de monitoramento.

> **Note**
>
> AS métricas que não tiverem novos pontos de dados nas últimas duas semanas não serão exibidas no console. Elas também não serão exibidas quando você digitar o nome da métrica ou os nomes de dimensão na caixa de pesquisa na guia Todas as métricas do console e não serão retornadas nos resultados de um comando list-metrics. A melhor maneira de recuperar essas métricas é com os comandos get-metric-data ou get-metric-statistics na AWS CLI.

Para procurar métricas disponíveis no CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. No campo de pesquisa da guia All metrics (Todas as métricas), insira um termo de pesquisa, como nome da métrica, namespace, ID da conta, rótulo da conta, nome ou valor da dimensão, ou nome do recurso. Isso mostra todos os namespaces com métricas com esse termo de pesquisa.

   Por exemplo, se você pesquisar **`volume`**, isso mostrará os namespaces que contêm métricas com esse termo no nome.

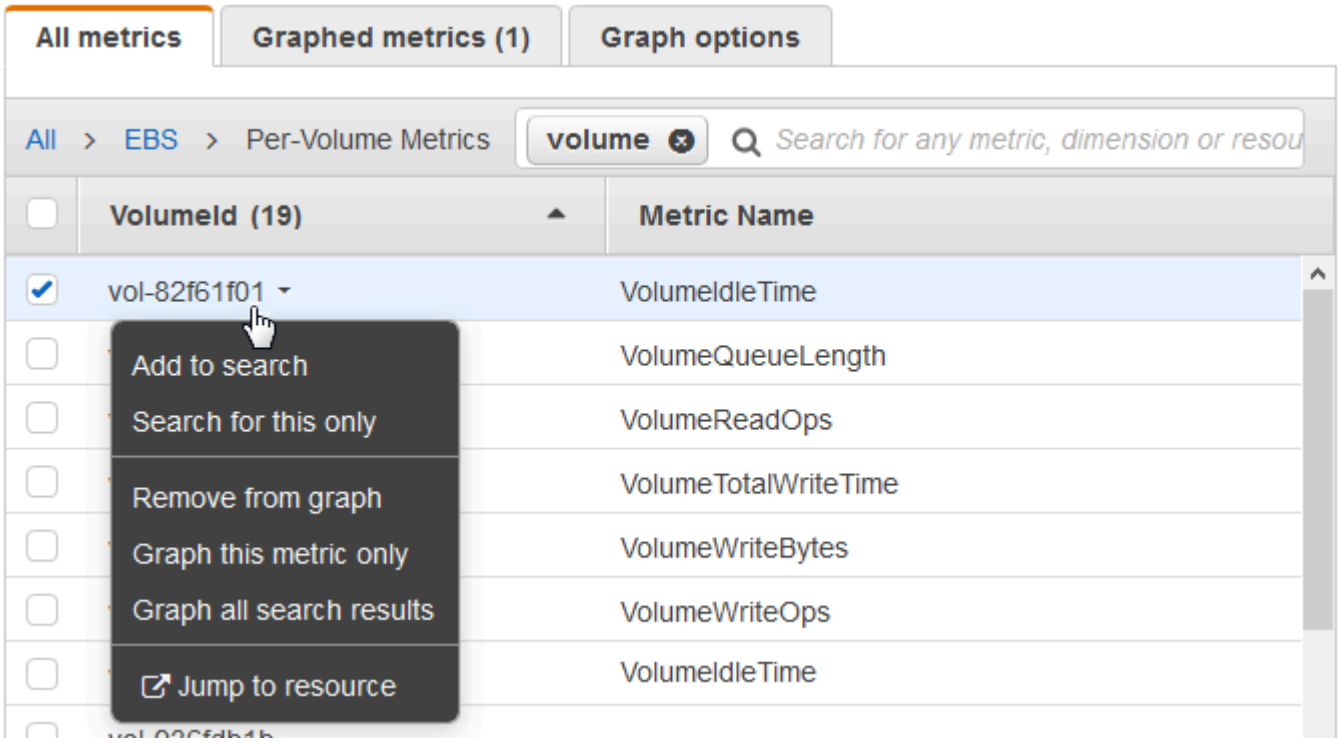
   Para obter mais informações sobre pesquisa, consulte Usar expressões de pesquisa em gráficos.
4. Para criar um gráfico de todos os resultados da pesquisa, selecione Graph search (Criar gráfico da pesquisa)

   ou

   Selecione um namespace para visualizar as métricas dele. Você pode, então, fazer o seguinte:

   a. Para criar um gráfico de uma ou mais métricas, marque a caixa de seleção ao lado de cada métrica. Para selecionar todas as métricas, marque a caixa de seleção na linha de cabeçalho da tabela.
   b. Para refinar a pesquisa, passe o mouse sobre o nome de uma métrica e selecione Add to search (Adicionar à pesquisa) ou Search for this only (Pesquisar apenas por isto).
   c. Para visualizar um dos recursos no console, escolha o ID do recurso e selecione Jump to resource (Pular para recurso).
   d. Para visualizar a ajuda para uma métrica, selecione o nome da métrica e escolha What is this? (O que é isso?).

   As métricas selecionadas aparecem no gráfico.



5. (Opcional) Selecione um dos botões na barra de pesquisa para editar a parte do termo de pesquisa.

Criar gráficos de métricas

Use o console do CloudWatch para criar gráficos de dados de métricas gerados por outros produtos da AWS. Isso torna mais eficiente visualizar a atividade da métrica em seus serviços. Os procedimentos a seguir descrevem como representar métricas em gráficos no CloudWatch.

Conteúdo

Criar um gráfico de uma métrica

É possível selecionar métricas e criar gráficos dos dados de métrica usando o console do CloudWatch.

O CloudWatch oferece suporte às seguintes estatísticas de métricas: `Average`, `Minimum`, `Maximum`, `Sum` e `SampleCount`. Para ter mais informações, consulte Estatísticas.

É possível visualizar seus dados em diferentes níveis de detalhes. Por exemplo, escolha uma exibição de um minuto, que pode ser útil durante a solução de problemas. Ou escolha uma visualização menos detalhada de uma hora. Isso pode ser útil ao visualizar um intervalo de tempo mais amplo (por exemplo, 3 dias) para que possa ver tendências ao longo do tempo. Para ter mais informações, consulte Períodos.

Se você estiver usando uma conta configurada como uma conta de monitoramento na observabilidade entre contas do CloudWatch, poderá gerar um gráfico das métricas das contas de origem vinculadas a essa conta de monitoramento. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

Criar um gráfico

Para criar um gráfico de uma métrica

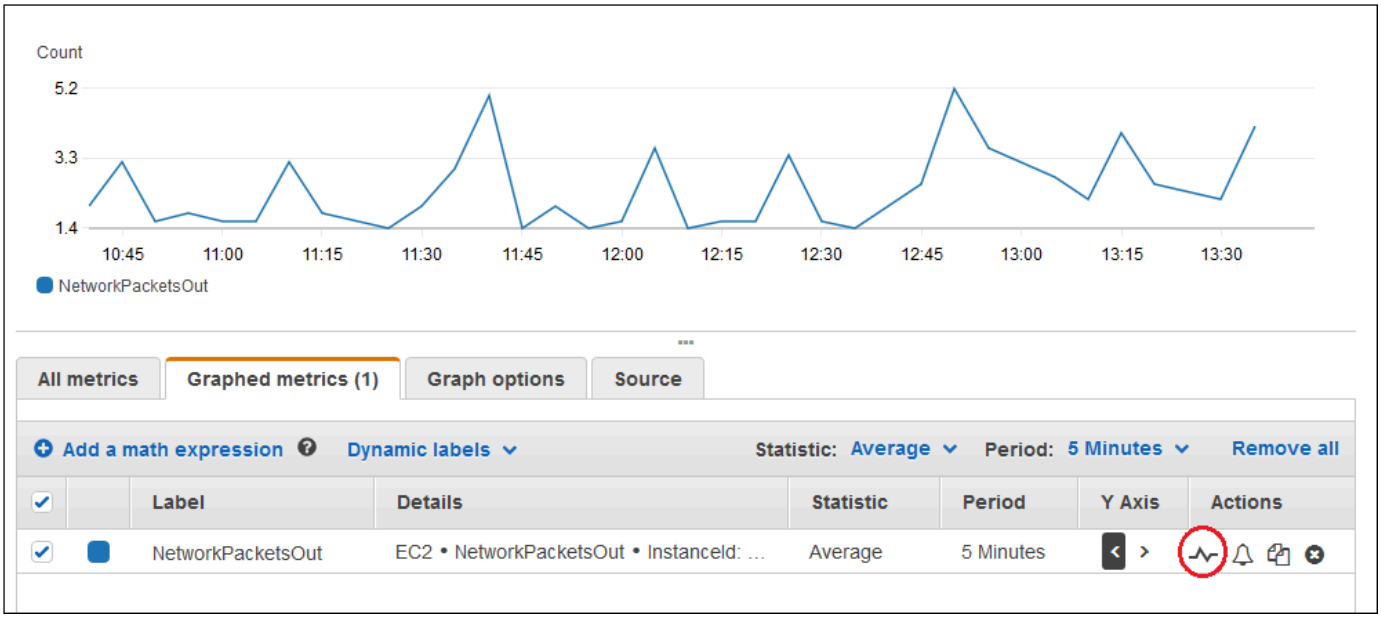
1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), All metrics (Todas as métricas).
3. Na guia Navegar, insira um termo no campo de pesquisa, como um nome de métrica, ID de conta ou nome de recurso.

   Por exemplo, se você pesquisar a métrica `CPUUtilization`, verá os namespaces e as dimensões com essa métrica.
4. Selecione um dos resultados da pesquisa para visualizar as métricas.
5. Para criar um gráfico de uma ou mais métricas, marque a caixa de seleção ao lado de cada métrica. Para selecionar todas as métricas, marque a caixa de seleção na linha de cabeçalho da tabela.
6. (Opcional) Para alterar o tipo de gráfico, selecione a guia Opções. Em seguida, será possível escolher entre um gráfico de linhas, gráfico de áreas empilhadas, exibição de números, gráfico de barras ou gráfico de pizza.
7. Escolha a guia Graphed metrics (Métricas em gráfico).

8. (Opcional) Para alterar a estatística usada no gráfico, escolha a nova estatística na lista Statistic (Estatística) ao lado do nome da métrica.

   Para obter mais informações sobre estatísticas do CloudWatch, consulte Definições de estatísticas do CloudWatch. Para obter mais informações sobre estatísticas de percentil Pxx, consulte Percentis.

9. (Opcional) Para adicionar uma faixa de detecção de anomalias que mostra os valores esperados para a métrica, escolha o ícone de detecção de anomalias em Actions (Ações), ao lado da métrica.



   O CloudWatch usa até duas semanas dos dados históricos recentes da métrica para calcular um modelo de valores esperados. Em seguida, exibe este intervalo de valores esperados como uma faixa no gráfico. O CloudWatch adiciona uma nova linha sob a métrica para exibir a expressão matemática da faixa de detecção de anomalias, rotulada ANOMALY_DETECTION_BAND. Se houver dados históricos recentes, você verá imediatamente uma faixa de detecção de anomalias, que é uma aproximação da faixa de detecção de anomalias gerada pelo modelo. Demora até 15 minutos para que a faixa de detecção de anomalias real apareça.

   Por padrão, o CloudWatch cria os limites superior e inferior da faixa de valores esperados com um valor padrão de 2 para o limite da faixa. Para alterar esse número, altere o valor ao final da fórmula em Details (Detalhes) para o segmento.

   - (Opcional) Escolha Edit model (Editar modelo) para alterar a forma como o modelo de detecção de anomalias é calculado. É possível excluir períodos passados e futuros para

que não sejam usados no treinamento para calcular o modelo. É fundamental excluir sistemas de eventos incomuns, como interrupções do sistema, implantações e feriados, dos dados de treinamento. Também é possível especificar o fuso horário a ser usado para o modelo para alterações de horário de verão.

Para ter mais informações, consulte Usar a detecção de anomalias do CloudWatch.

Para ocultar o modelo do gráfico, remova a marca de seleção da linha com a função `ANOMALY_DETECTION_BAND` ou escolha o ícone `X`. Para excluir completamente o modelo, escolha Edit model (Editar modelo), Delete model (Excluir modelo).

10. (Opcional) Conforme escolhe as métricas para o gráfico, especifique um rótulo dinâmico para aparecer na legenda do gráfico para cada métrica. Os rótulos dinâmicos exibem uma estatística sobre a métrica e são atualizados automaticamente quando o painel ou o gráfico é atualizado. Para adicionar um rótulo dinâmico, escolha Métricas em gráfico e Rótulos dinâmicos.

    Por padrão, os valores dinâmicos que você adicionar ao rótulo aparecerão no início do rótulo. Em seguida, você pode escolher o valor do Label (Rótulo) para a métrica para editar o rótulo. Para ter mais informações, consulte Usar rótulos dinâmicos.

11. Para obter mais informações sobre a métrica que está sendo usada no gráfico, passe o mouse sobre a legenda.

12. As anotações horizontais podem ajudar os usuários do gráfico a visualizar com mais eficiência quando uma métrica aumentou rapidamente para um determinado nível ou se a métrica está dentro de um intervalo predefinido. Para adicionar uma anotação horizontal, escolha a guia Opções e, em seguida, Adicionar anotação horizontal:

    a. Em Label (Rótulo), insira um rótulo para a anotação.

    b. Em Value (Valor), insira o valor da métrica em que a anotação horizontal aparece.

    c. Em Fill, especifique se o preenchimento do sombreamento deve ser usado com essa anotação. Por exemplo, selecione `Above` ou `Below` para a área correspondente ser preenchida. Se você especificar `Between`, outro campo `Value` será exibido e a área do gráfico entre os dois valores será preenchida.

    d. Em Axis (Eixos), especifique se os números em `Value` se referem à métrica associada ao eixo Y esquerdo ou direito, caso o gráfico inclua várias métricas.

    Você pode alterar a cor de preenchimento de uma anotação, escolhendo a cor no quadrado de cores na coluna à esquerda da anotação.

Repita essas etapas para adicionar várias anotações horizontais ao mesmo gráfico.

Para ocultar uma anotação, desmarque a caixa de seleção na coluna da esquerda para essa anotação.

Para excluir uma anotação, selecione x na coluna Actions (Ações).

13. Para obter uma URL para o gráfico, escolha Ações, Compartilhar. Copie o URL para salvar ou compartilhar.
14. Para adicionar o gráfico a um painel, escolha Ações, Adicionar ao painel.

Criar um gráfico de métricas com base em outra fonte de dados

Você pode criar um gráfico que exibe recursos de fontes de dados que não sejam o CloudWatch. Para obter mais informações sobre como criar conexões com essas outras fontes de dados, consulte Métricas de consulta de outras fontes de dados.

Como representar graficamente uma métrica com base em outra fonte de dados

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), All metrics (Todas as métricas).
3. Escolha a guia Consulta a várias fontes.
4. Em Fonte de dados, selecione a fonte de dados que deseja usar.

   Se você ainda não tiver criado uma conexão com a fonte de dados desejada, selecione Create and manage data sources e escolha Create and manage data sources. Para obter informações sobre o restante desse processo de criação de fonte de dados, consulte Conectar-se a uma fonte de dados pré-criada com um assistente.
5. O assistente ou editor de consultas solicita as informações necessárias para a consulta. O fluxo de trabalho é diferente para cada fonte de dados e é adaptado à fonte de dados. Por exemplo, para o Amazon Managed Service for Prometheus e fontes de dados do Prometheus, é exibida uma caixa do editor de consultas PromQL com um auxiliar de consulta.
6. Quando você terminar a estrutura da consulta, escolha Representar consulta graficamente.

   O gráfico é preenchido com métricas da consulta.
7. (Opcional) As anotações horizontais podem ajudar os usuários do gráfico a visualizar com mais eficiência quando uma métrica aumentou rapidamente para um determinado nível ou se

a métrica está dentro de um intervalo predefinido. Para adicionar uma anotação horizontal, escolha a guia Opções e, em seguida, Adicionar anotação horizontal:

a. Em Label (Rótulo), insira um rótulo para a anotação.

b. Em Value (Valor), insira o valor da métrica em que a anotação horizontal aparece.

c. Em Fill, especifique se o preenchimento do sombreamento deve ser usado com essa anotação. Por exemplo, selecione `Above` ou `Below` para a área correspondente ser preenchida. Se você especificar `Between`, outro campo `Value` será exibido e a área do gráfico entre os dois valores será preenchida.

d. Em Axis (Eixos), especifique se os números em `Value` se referem à métrica associada ao eixo Y esquerdo ou direito, caso o gráfico inclua várias métricas.

Você pode alterar a cor de preenchimento de uma anotação, escolhendo a cor no quadrado de cores na coluna à esquerda da anotação.

Repita essas etapas para adicionar várias anotações horizontais ao mesmo gráfico.

Para ocultar uma anotação, desmarque a caixa de seleção na coluna da esquerda para essa anotação.

Para excluir uma anotação, selecione x na coluna Actions (Ações).

8. (Opcional) Para adicionar esse gráfico a um painel, escolha Ações, Adicionar ao painel.

Atualizar um gráfico

Para atualizar um gráfico

1. Para alterar o nome do gráfico, escolha o ícone de lápis.
2. Para alterar o período, selecione um dos valores predefinidos ou escolha custom (personalizado). Para ter mais informações, consulte Modificar o formato de período ou fuso horário de um gráfico.
3. Para alterar a estatística, escolha a guia Métricas em gráfico. Escolha o cabeçalho da coluna ou um valor individual e escolha uma das estatísticas ou percentis predefinidos ou especifique um percentil personalizado (por exemplo, **`p95.45`**).
4. Para alterar o período, escolha a guia Métricas em gráfico. Escolha o cabeçalho da coluna ou um valor individual e, então, escolha um valor diferente.

5. Para adicionar uma anotação horizontal, selecione Graph options (Opções do gráfico) e Add horizontal annotation (Adicionar anotação horizontal):

   a. Em Label (Rótulo), insira um rótulo para a anotação.

   b. Em Value (Valor), insira o valor da métrica em que a anotação horizontal aparece.

   c. Em Fill, especifique se o preenchimento do sombreamento deve ser usado com essa anotação. Por exemplo, selecione `Above` ou `Below` para a área correspondente ser preenchida. Se você especificar `Between`, outro campo `Value` será exibido e a área do gráfico entre os dois valores será preenchida.

   d. Em Axis (Eixos), especifique se os números em `Value` se referem à métrica associada ao eixo Y esquerdo ou direito, caso o gráfico inclua várias métricas.

      Você pode alterar a cor de preenchimento de uma anotação, escolhendo a cor no quadrado de cores na coluna à esquerda da anotação.

   Repita essas etapas para adicionar várias anotações horizontais ao mesmo gráfico.

   Para ocultar uma anotação, desmarque a caixa de seleção na coluna da esquerda para essa anotação.

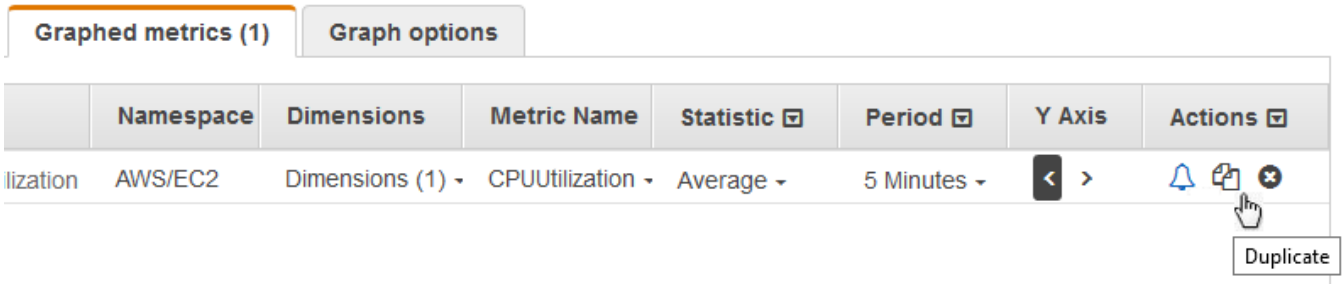
   Para excluir uma anotação, selecione x na coluna Actions (Ações).

6. Para alterar o intervalo de atualização, selecione Refresh options (Opções de atualização) e selecione Auto refresh (Atualizar automaticamente) ou escolha 1 Minute (1 minuto), 2 Minutes (2 minutos), 5 Minutes (5 minutos) ou 15 Minutes (15 minutos).

Duplicar uma métrica

Para duplicar uma métrica

1. Escolha a guia Graphed metrics (Métricas em gráfico).
2. Em Ações, selecione o ícone Duplicar.



3. Atualizar a métrica duplicada, conforme necessário.

Mesclar dois gráficos em um

É possível mesclar dois gráficos diferentes em um e, em seguida, o gráfico resultante mostrará as duas métricas. Isso pode ser útil se você já tiver métricas diferentes exibidas em gráficos diferentes e quiser combiná-las, ou se quiser criar facilmente um único gráfico com métricas de diferentes regiões.

Para mesclar um gráfico em outro, você usa a URL ou a fonte JSON do gráfico que deseja mesclar.

Para mesclar dois gráficos em um

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Abra o gráfico que você deseja mesclar em outro gráfico. Para fazer isso, é possível escolher Métricas, Todas as métricas e, em seguida, escolher uma métrica para representar graficamente. Ou é possível abrir um painel e, em seguida, abrir um dos gráficos no painel selecionando o gráfico e escolhendo Abrir em métricas no menu no canto superior direito do gráfico.
3. Depois de ter aberto um gráfico, faça um dos seguintes:
   - Copie o URL da barra do navegador.
   - Escolha a guia Origem e, em seguida, escolha Copiar.
4. Abra o gráfico que você deseja mesclar no gráfico anterior.
5. Quando você tiver o segundo gráfico aberto na visualização Métricas, escolha Ações, Mesclar gráfico.
6. Insira o URL ou o JSON que você copiou anteriormente e escolha Mesclar.
7. Os gráficos mesclados serão exibidos. O eixo y à esquerda é para o gráfico original e o eixo y à direita é para o gráfico que você mesclou nele.

> **Note**
>
> Se o gráfico em que você mesclou usa a função METRICS(), as métricas no gráfico que foi mesclado não serão incluídas no cálculo de METRICS() do gráfico mesclado.

8. Para adicionar o gráfico a um painel, escolha Ações, Adicionar a painel.

Usar rótulos dinâmicos

Você pode usar rótulos dinâmicos com os seus gráficos. Os rótulos dinâmicos adicionam um valor atualizado dinamicamente ao rótulo da métrica selecionada. Você pode adicionar uma ampla gama de valores aos rótulos, conforme mostrado nas tabelas a seguir.

O valor dinâmico mostrado no rótulo é derivado do intervalo de tempo mostrado atualmente no gráfico. A parte dinâmica do rótulo é atualizada automaticamente quando o painel ou o gráfico é atualizado.

Se você usar um rótulo dinâmico com uma expressão de pesquisa, o rótulo dinâmico será aplicado a todas as métricas retornadas pela pesquisa.

É possível usar o console do CloudWatch para adicionar um valor dinâmico a um rótulo, editar o rótulo, alterar a posição do valor dinâmico na coluna do rótulo e fazer outras personalizações.

Rótulos dinâmicos

Em um rótulo dinâmico, é possível usar os seguintes valores relacionados às propriedades da métrica:

Valor em tempo real do rótulo dinâmico	Descrição
${AVG}	A média dos valores no intervalo de tempo atualmente mostrado no gráfico.
${DATAPOINT_COUNT}	O número de pontos de dados no intervalo de tempo exibido atualmente no gráfico.

Valor em tempo real do rótulo dinâmico	Descrição
${FIRST}	Os valores de métrica mais antigos no intervalo de tempo mostrado atualmente no gráfico.
${FIRST_LAST_RANGE}	A diferença entre os valores de métrica dos pontos de dados mais antigos e mais recentes que são exibidos atualmente no gráfico.
${FIRST_LAST_TIME_RANGE}	O intervalo de tempo absoluto entre os pontos de dados mais antigos e mais recentes que são exibidos atualmente no gráfico.
${FIRST_TIME}	O carimbo de data/hora do ponto de dados mais antigo no intervalo de tempo exibido atualmente no gráfico.
${FIRST_TIME_RELATIVE}	A diferença de tempo absoluta entre agora e o carimbo de data/hora do ponto de dados mais antigo no intervalo de tempo exibido atualmente no gráfico.
${LABEL}	A representação do rótulo padrão de uma métrica.
${LAST}	Os valores de métrica mais recentes no intervalo de tempo mostrado atualmente no gráfico.
${LAST_TIME}	O carimbo de data/hora do ponto de dados mais rcente no intervalo de tempo exibido atualmente no gráfico.
${LAST_TIME_RELATIVE}	A diferença de tempo absoluta entre agora e o carimbo de data/hora do ponto de dados mais recente no intervalo de tempo exibido atualmente no gráfico.
${MAX}	O máximo dentre os valores no intervalo de tempo mostrado atualmente no gráfico.
${MAX_TIME}	O carimbo de data/hora do ponto de dados que tem o valor de métrica mais alto, dos pontos de dados exibidos atualmente no gráfico.

Valor em tempo real do rótulo dinâmico	Descrição
${MAX_TIME_RELATIVE}	A diferença de tempo absoluta entre agora e o carimbo de data/hora do ponto de dados com o valor mais alto, dos pontos de dados exibidos atualmente no gráfico.
${MIN}	O mínimo dentre os valores no intervalo de tempo mostrado atualmente no gráfico.
${MIN_MAX_RANGE}	A diferença em valores de métrica entre os pontos de dados com o valor mais alto e o valor mais baixo, dos pontos de dados exibidos atualmente no gráfico.
${MIN_MAX_TIME_RANGE}	O intervalo de tempo absoluto entre os pontos de dados com o valor mais alto e o valor mai baixo, dos pontos de dados exibidos atualmente no gráfico.
${MIN_TIME}	O carimbo de data/hora do ponto de dados que tem o valor de métrica mais baixo, dos pontos de dados exibidos atualmente no gráfico.
${MIN_TIME_RELATIVE}	A diferença de tempo absoluta entre agora e o carimbo de data/hora do ponto de dados com o valor mais baixo, dos pontos de dados exibidos atualmente no gráfico.
${PROP('AccountId')}	O ID da conta da AWS da métrica.
${PROP('AccountLabel')}	O rótulo especificado para a conta de origem que é a proprietária dessa métrica, na observabilidade entre contas do CloudWatch.
${PROP('Dim.*dimension_name*')}	O valor da dimensão especificada. Substitua *dimension_name* pelo nome da dimensão, com distinção entre maiúsculas e minúsculas.
${PROP('MetricName')}	O nome da métrica.
${PROP('Namespace')}	O namespace da métrica.

Valor em tempo real do rótulo dinâmico	Descrição
${PROP('Period')}	O período da métrica, em segundos.
${PROP('Region')}	A região da AWS onde a métrica será publicada.
${PROP('Stat')}	A estatística métrica que está sendo representada em gráficos.
${SUM}	A soma dos valores no intervalo de tempo mostrado atualmente no gráfico.

Por exemplo, suponha uma expressão de pesquisa **`SEARCH(' {AWS/Lambda, FunctionName} Errors ', 'Sum')`**, que encontra os `Errors` para cada uma de suas funções do Lambda. Se você definir o rótulo como `[max: ${MAX} Errors for Function Name ${LABEL}]`, o rótulo de cada métrica será [max: ***número*** de erros para a função chamada ***Nome***].

Você pode adicionar até seis valores dinâmicos a um rótulo. Você pode usar o espaço reservado `${LABEL}` somente uma vez em cada rótulo.

Modificar o formato de período ou fuso horário de um gráfico

Esta seção descreve como é possível modificar o formato de data, hora e fuso horário em um gráfico de métricas do CloudWatch. Ele também descreve como você pode ampliar um gráfico para aplicar um intervalo de tempo específico. Para obter informações sobre como criar um gráfico, leia Criar um gráfico de uma métrica.

> **Note**
>
> Se o intervalo de tempo de um painel for menor do que o período usado para um gráfico no painel, acontecerá o seguinte:
>
> - O gráfico é modificado para exibir a quantidade de dados correspondente a um período completo para esse widget, mesmo que esse período seja maior do que o intervalo de tempo do painel. Isso garante que haja pelo menos um ponto de dados no gráfico.
> - A hora de início do período para esse ponto de dados é ajustada retroativamente para garantir que pelo menos um ponto de dados possa ser exibido.

Definir um período relativo

New interface

Para especificar um intervalo de tempo relativo para um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas). No canto superior direito da tela, é possível selecionar um dos intervalos de período predefinidos, que variam de 1 hora a 1 semana (1h, 3h, 12h, 1d, 3d ou 1w). Como alternativa, você pode escolher Custom (Personalizado) para definir seu próprio intervalo de tempo.
3. Selecione Custom (Personalizado) e, em seguida, selecione a guia Relative (Relativo) no canto superior esquerdo da caixa. Você pode especificar um intervalo de tempo em Minutes (Minutos), Hours (Horas), Days (Dias), Weeks (Semanas), Months (Meses).
4. Depois de especificar um período, escolha Apply (Aplicar).

Original interface

Para especificar um intervalo de tempo relativo para um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas). No canto superior direito da tela, é possível selecionar um dos intervalos de período predefinidos, que variam de 1 hora a 1 semana (1h, 3h, 12h, 1d, 3d ou 1w). Como alternativa, você pode escolher custom (personalizado) para definir seu próprio intervalo de tempo.
3. Selecione custom (personalizado) e, em seguida, escolha a guia Relative (Relativo) no canto superior esquerdo da caixa. Você pode especificar um intervalo de tempo em Minutes (Minutos), Hours (Horas), Days (Dias), Weeks (Semanas) ou Months (Meses).

Definir um período absoluto

New interface

Para especificar um intervalo de tempo absoluto para um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas). No canto superior direito da tela, é possível selecionar um dos intervalos de período predefinidos, que variam de 1 hora a 1 semana (1h, 3h, 12h, 1d, 3d ou 1w). Como alternativa, você pode escolher Custom (Personalizado) para definir seu próprio intervalo de tempo.
3. Selecione Custom (Personalizado) e, em seguida, selecione a guia Absolute (Absoluto) no canto superior esquerdo da caixa. Use o seletor de calendário ou as caixas de campos de texto para especificar um intervalo de tempo.
4. Depois de especificar um período, escolha Apply (Aplicar).

Original interface

Para especificar um intervalo de tempo absoluto para um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas). No canto superior direito da tela, é possível selecionar um dos intervalos de período predefinidos, que variam de 1 hora a 1 semana (1h, 3h, 12h, 1d, 3d ou 1w). Como alternativa, você pode escolher custom (personalizado) para definir seu próprio intervalo de tempo.
3. Selecione custom (personalizado) e, em seguida, escolha a guia Absolute (Absoluto) no canto superior esquerdo da caixa. Use o seletor de calendário ou as caixas de campos de texto para especificar um intervalo de tempo.
4. Depois de especificar um período, escolha Apply (Aplicar).

Definir o formato de fuso horário

New interface

Para especificar o fuso horário de um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas). No canto superior direito da tela, é possível selecionar um dos intervalos de período predefinidos, que variam de 1 hora a 1 semana (1h, 3h, 12h, 1d, 3d ou 1w). Como alternativa, você pode escolher Custom (Personalizado) para definir seu próprio intervalo de tempo.
3. Selecione Custom (Personalizado) e, em seguida, escolha a lista suspensa no canto superior direito da caixa. Você pode alterar o fuso horário para UTC ou Local time zone (Fuso horário local).
4. Depois de fazer as alterações, escolha Apply (Aplicar).

Original interface

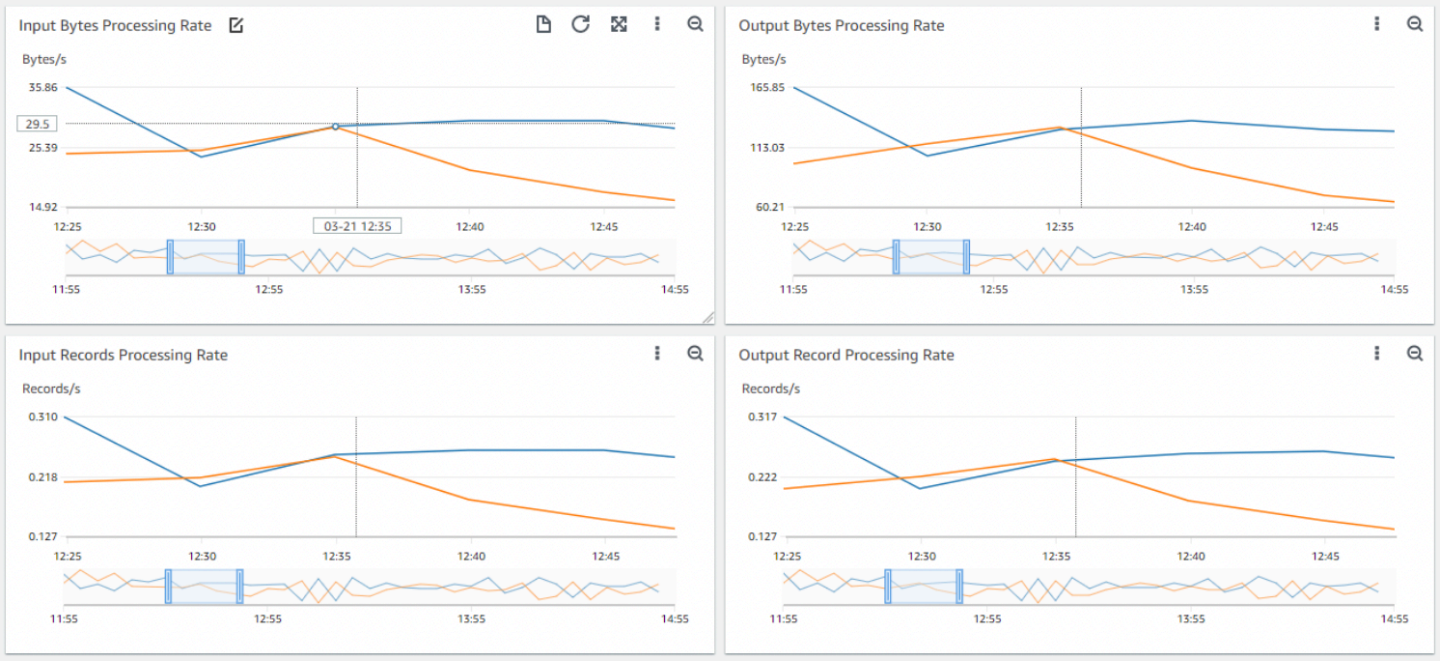
Para especificar o fuso horário de um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas). No canto superior direito da tela, é possível selecionar um dos intervalos de período predefinidos, que variam de 1 hora a 1 semana (1h, 3h, 12h, 1d, 3d ou 1w). Como alternativa, você pode escolher custom (personalizado) para definir seu próprio intervalo de tempo.
3. Selecione custom (personalizado) e, em seguida, escolha a lista suspensa no canto superior direito da caixa. Você pode alterar o fuso horário para UTC ou Local time zone (Fuso horário local).

Aumentar o zoom em um gráfico de linhas ou gráfico de áreas empilhadas

No console do CloudWatch, é possível usar o recurso de zoom de minimapa para focar em seções de gráficos de linhas e gráficos de área empilhada sem alterar entre as visualizações com zoom aumentado e zoom diminuído. Por exemplo, é possível usar o recurso de zoom de minimapa para

focar em um pico em um gráfico de linhas para comparar o pico com outras métricas no seu painel usando a mesma linha do tempo. Os procedimentos desta seção descrevem como usar o recurso de zoom.



Na imagem anterior, o recurso de zoom se concentra em um pico em um gráfico de linhas que está relacionado à taxa de processamento de bytes de entrada, além de mostrar outros gráficos de linha no painel que se concentram em seções da mesma linha do tempo.

New interface

Para ampliar um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas).
3. Escolha Browse (Procurar) e, em seguida, selecione uma ou mais métricas para adicionar ao gráfico.
4. Escolha Options (Opções) e selecione Line (Linhas) em Widget type (Tipo de widget).
5. Escolha e arraste a área do gráfico em que deseja focar e, em seguida, solte-a.
6. Para redefinir o zoom, escolha o ícone Redefinir zoom igual a uma lupa com um símbolo de menos (-) no interior.

Original interface

Para ampliar um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas).
3. Escolha All metrics (Todas as métricas) e, em seguida, selecione uma ou mais métricas para adicionar ao gráfico.
4. Selecione Graph options (Opções do gráfico). Em Widget type (Tipo de widget), selecione Line (Linhas).
5. Escolha e arraste a área do gráfico em que deseja focar e, em seguida, solte-a.
6. Para redefinir o zoom, escolha o ícone Redefinir zoom igual a uma lupa com um símbolo de menos (-) no interior.

> **Tip**
>
> Se você já criou um painel que contém um gráfico de linhas ou um gráfico de área empilhada, poderá ir até o painel e começar a usar o recurso de zoom.

Modificar o eixo Y de um gráfico

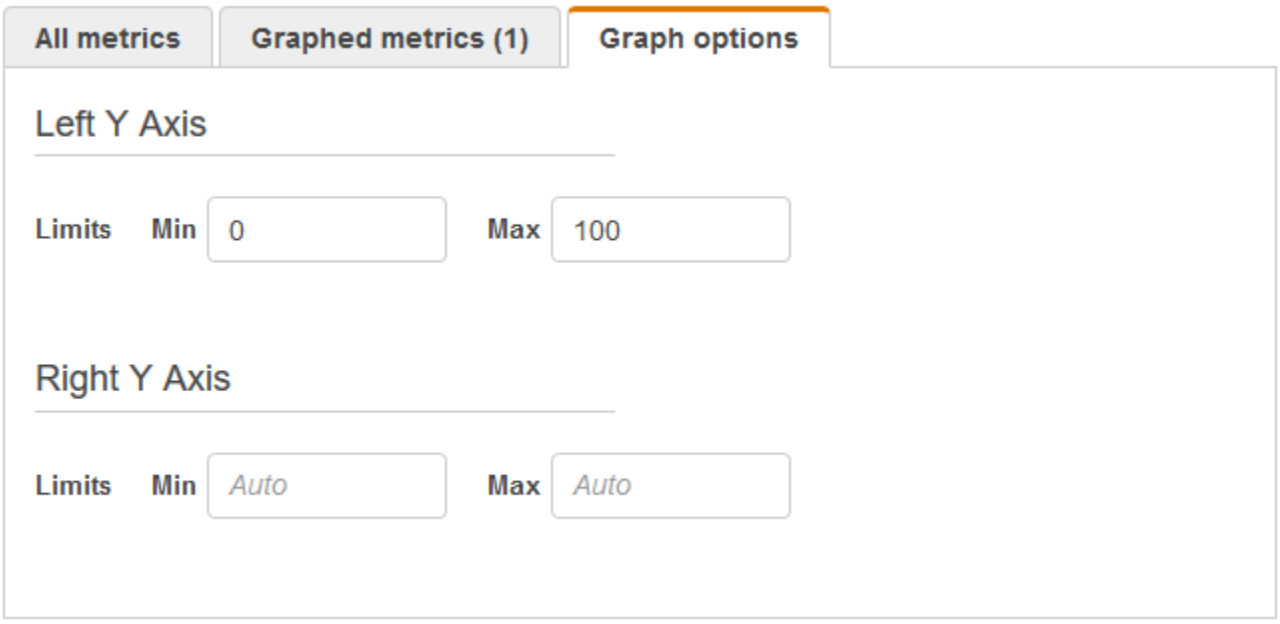
Você pode definir limites personalizados para o eixo Y em um gráfico para ajudar a visualizar melhor os dados. Por exemplo, é possível alterar os limites em um gráfico `CPUUtilization` para 100% para ficar fácil determinar se o uso da CPU está baixo (a linha é exibida na parte inferior do gráfico) ou alto (a linha é exibida na parte superior do gráfico).

Você pode alternar entre dois eixos Y diferentes para o gráfico. Isso é útil quando o gráfico contém métricas com unidades diferentes ou com uma grande diferença no intervalo de valores.

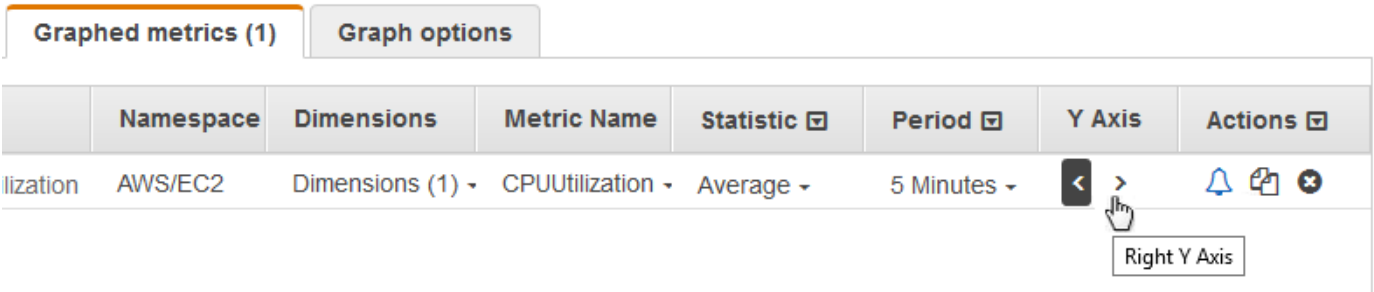
Para modificar o eixo Y de um gráfico

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Selecione um namespace de métrica (por exemplo, EC2) e uma dimensão de métrica (por exemplo, Per-Instance Metrics (Métricas por instância)).

4. A guia Todas as métricas exibe todas as métricas dessa dimensão naquele namespace. Para criar um gráfico de uma métrica, marque a caixa de seleção ao lado da métrica.
5. Na guia Opções de gráfico, especifique os valores Mín e Máx para o Eixo Y esquerdo. O valor de Min (Mínimo) não pode ser maior do que o valor de Max (Máximo).



6. Para criar um segundo eixo Y, especifique os valores Min (Mínimo) e Max (Máximo) para o Right Y Axis (Eixo Y direito).
7. Para alternar entre os dois eixos Y, selecione a guia Graphed metrics (Métricas em gráfico). Em Eixo Y, escolha Eixo Y esquerdo ou Eixo Y direito.

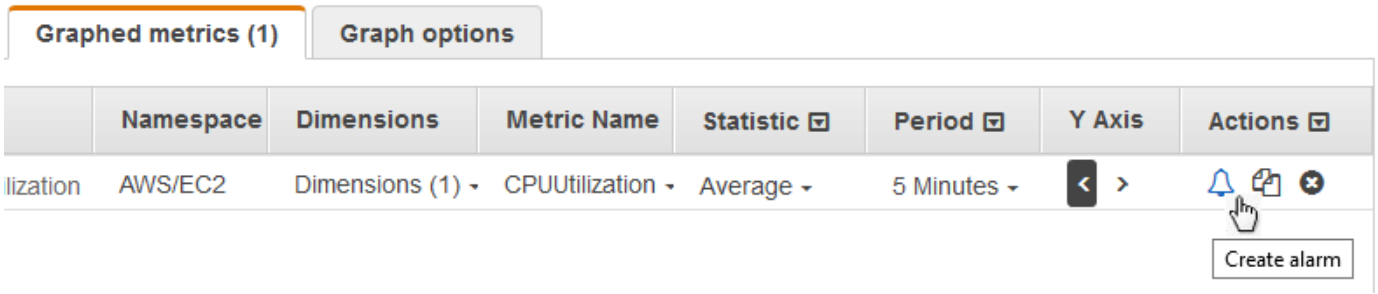


Criar um alarme a partir de uma métrica em um gráfico

Você pode criar um gráfico para uma métrica e, em seguida, um alarme da métrica no gráfico, que tem como vantagem preencher vários campos do alarme para você.

Para criar um alarme de uma métrica em um gráfico

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Selecione um namespace de métrica (por exemplo, EC2) e uma dimensão de métrica (por exemplo, Per-Instance Metrics (Métricas por instância)).
4. A guia Todas as métricas exibe todas as métricas dessa dimensão naquele namespace. Para criar um gráfico de uma métrica, marque a caixa de seleção ao lado da métrica.
5. Para criar um alarme para a métrica, selecione a guia Métricas em gráfico. Em Ações, escolha o ícone de alarme.



6. Em Conditions (Condições), escolha Static (Estática) ou Detecção de anomalias para especificar se um limite estático ou um modelo de detecção de anomalias para o alarme deve ser usado.

   Dependendo da sua escolha, insira o restante dos dados para as condições do alarme.
7. Escolha Additional configuration (Configuração adicional). Em Datapoints to alarm (Pontos de dados para alarme), especifique quantos períodos de avaliação (pontos de dados) devem estar no estado `ALARM` para disparar o alarme. Se os dois valores forem correspondentes, você criará um alarme que passa para o estado `ALARM` se esses períodos consecutivos estiverem violando.

   Para criar um alarme M de N, especifique um número menor para o primeiro valor que especificar para o segundo valor. Para ter mais informações, consulte Avaliar um alarme.
8. Para o Missing data treatment (Tratamento de dados ausentes), escolha como deseja que o alarme se comporte quando alguns pontos de dados estiverem ausentes. Para ter mais informações, consulte Configurar como os alarmes do CloudWatch tratam dados ausentes.
9. Escolha Próximo.
10. Em Notification (Notificação), selecione um tópico do SNS para notificar quando o alarme estiver no estado `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

Para que o alarme envie várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

Para que o alarme não envie notificações, escolha Remove (Remover).

11. Para que o alarme execute o Auto Scaling ou ações do EC2, escolha o botão apropriado e escolha o estado do alarme e a ação a ser executada.
12. Quando terminar, escolha Next (Próximo).
13. Digite um nome e uma descrição para o alarme. O nome deve conter somente caracteres ASCII. Em seguida, escolha Próximo.
14. Em Preview and create (Visualizar e criar), confirme se as informações e condições são o que você deseja e escolha Create alarm (Criar alarme).

Usar a detecção de anomalias do CloudWatch

Quando você habilita a detecção de anomalias para uma métrica, o CloudWatch aplica algoritmos estatísticos e de machine learning. Esses algoritmos analisam continuamente métricas de sistemas e aplicativos, determinam linhas de base normais e apontam anomalias com intervenção mínima do usuário.

Os algoritmos geram um modelo de detecção de anomalias. O modelo gera um intervalo de valores esperados que representam o comportamento normal da métrica.

Você pode habilitar a detecção de anomalias usando o AWS Management Console, a AWS CLI, o AWS CloudFormation ou o SDK da AWS. É possível habilitar a detecção de anomalias em métricas fornecidas pela AWS e também em métricas personalizadas. Em uma conta configurada como uma conta de monitoramento para a observabilidade entre contas do CloudWatch, você pode criar detectores de anomalias em métricas em contas de origem, além de métricas na conta de monitoramento.

Você pode usar o modelo de valores esperados de duas formas:

- Crie alarmes de detecção de anomalias com base no valor esperado de uma métrica. Esses tipos de alarmes não têm um limite estático para determinar o estado do alarme. Em vez disso, eles comparam o valor da métrica ao valor esperado com base no modelo de detecção de anomalias.

  É possível escolher se o alarme deverá ser acionado quando o valor da métrica estiver acima da faixa de valores esperados, abaixo da faixa ou acima ou ambos.

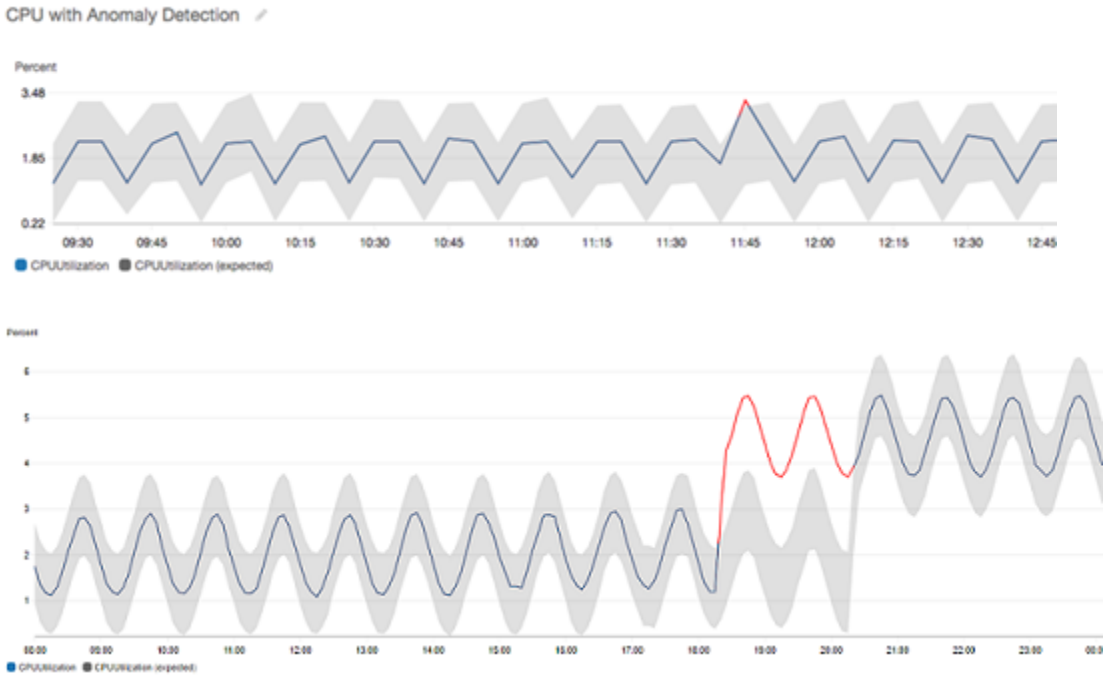
Para obter mais informações, consulte Criar um alarme do CloudWatch com base na detecção de anomalias.

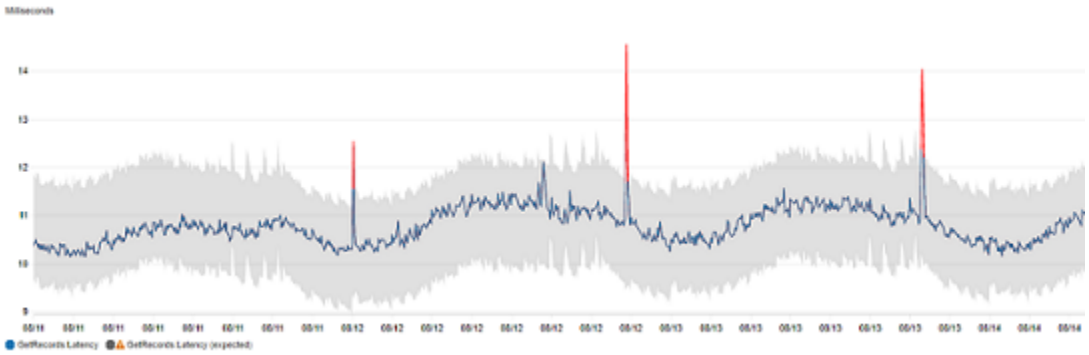
- Ao visualizar um gráfico de dados de métricas, sobreponha os valores esperados no gráfico como uma faixa. Isso faz com que fique visualmente claro quais valores no gráfico estão fora do intervalo normal. Para obter mais informações, consulte Criar um gráfico.

  Você também pode recuperar os valores superior e inferior do segmento do modelo usando a solicitação de API `GetMetricData` com a função matemática de métrica `ANOMALY_DETECTION_BAND`. Para obter mais informações, consulte GetMetricData.

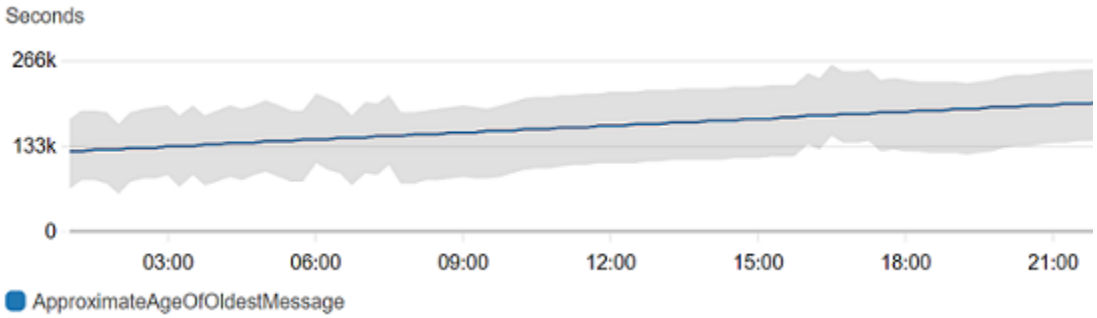
Em um gráfico com detecção de anomalias, o intervalo esperado de valores é mostrado como uma faixa cinza. Se o valor real da métrica for além dessa faixa, ela será mostrada como vermelha durante esse período.

Os algoritmos de detecção de anomalias consideram a sazonalidade e as mudanças de tendência das métricas. As mudanças de sazonalidade podem ser por hora, dia ou semana, conforme mostrado nos exemplos a seguir.

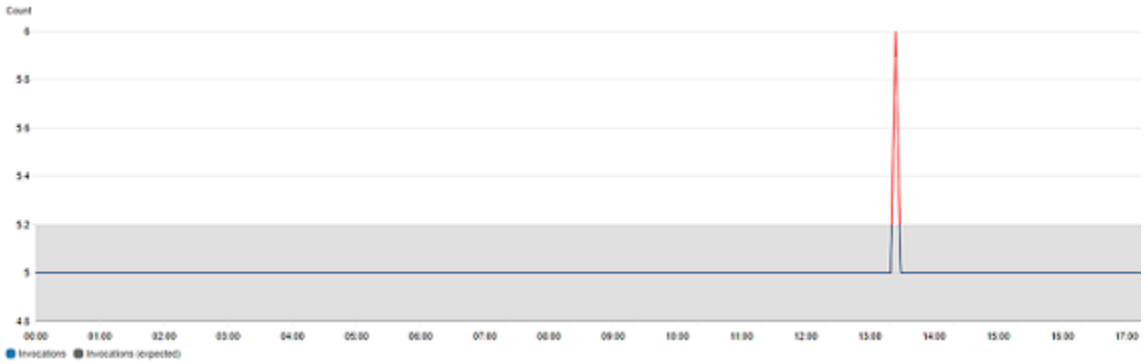




As tendências de maior alcance podem ser para baixo ou para cima.



As detecções de anomalias também funcionam bem com métricas com padrões planos.



Como funciona a detecção de anomalias do CloudWatch

Quando você habilita a detecção de anomalias para uma métrica, o CloudWatch aplica algoritmos de machine learning aos dados anteriores da métrica para criar um modelo dos valores esperados da métrica. O modelo avalia as tendências e padrões por hora, dia e semana da métrica. O algoritmo é treinado em até duas semanas de dados de métrica, mas é possível habilitar a detecção de anomalias em uma métrica mesmo que ela não tenha um total de duas semanas de dados.

Especifique um valor para o limite de detecção de anomalias que o CloudWatch usa junto com o modelo para determinar o intervalo “normal” de valores da métrica. Um valor mais alto para o limite de detecção de anomalias produz uma faixa mais larga de valores "normais".

O modelo de machine learning é específico para uma métrica e uma estatística. Por exemplo, se você habilitar a detecção de anomalias para uma métrica usando a estatística `AVG`, o modelo será específico à estatística `AVG`.

Ao criar um modelo para muitas métricas comuns de produtos da AWS, o CloudWatch garante que a banda não excederá os valores lógicos. Por exemplo, a faixa para `MemoryUtilization` de uma instância do EC2 permanecerá entre 0 e 100, e as faixas que acompanham as `Requests` do CloudFront, que não podem ser negativas, nunca ficarão abaixo de zero.

Depois de criar um modelo, a detecção de anomalias do CloudWatch avaliará continuamente o modelo e fará ajustes para garantir que ele seja o mais preciso possível. Isso inclui treinar novamente o modelo para fazer ajustes, caso os valores de métricas evoluam ao longo do tempo ou apresentem mudanças repentinas, além de incluir preditores para melhorar os modelos de métricas sazonais, variáveis ou esparsas.

Depois que habilitar a detecção de anomalias em uma métrica, você poderá excluir prazos específicos da métrica para que não sejam usados para treinar o modelo. Dessa forma, você pode excluir as implantações ou outros eventos incomuns para que não sejam usados para treinamento do modelo, garantindo a criação de um modelo mais preciso.

O uso de modelos de detecção de anomalias para alarmes gera cobranças na sua conta da AWS. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

Detecção de anomalias em matemática de métrica

A detecção de anomalias em matemática métrica é um recurso que pode ser usado para criar alarmes de detecção de anomalias na saída de expressões matemáticas métricas. É possível usar essas expressões para criar gráficos de visualização de bandas de detecção de anomalias. O recurso suporta funções aritméticas básicas, comparações e operadores lógicos e a maioria das outras funções. Para obter informações sobre funções que não são suportadas, consulteUsar matemática métricasnoGuia do usuário do Amazon CloudWatch.

Você pode criar modelos de detecção de anomalias com base em expressões matemáticas métricas semelhantes a forma que você já cria modelos de detecção de anomalias. No console do

CloudWatch, você pode aplicar a detecção de anomalias a expressões matemáticas métricas e selecionar a detecção de anomalias como um tipo de limite para essas expressões.

> **Note**
>
> A detecção de anomalias em matemática métrica só pode ser ativada e editada na versão mais recente da interface do usuário de métricas. Quando você cria detectores de anomalias com base em expressões matemáticas de métrica na nova versão da interface, pode visualizá-los na versão antiga, mas não os editar.

Para obter informações sobre como criar alarmes e modelos para detecção de anomalias e matemática métrica, consulte as seguintes seções:

- Criar um alarme do CloudWatch com base na detecção de anomalias
- Criar um alarme do CloudWatch com base em uma expressão matemática métrica

Você também pode criar, excluir e descobrir modelos de detecção de anomalias com base em expressões matemáticas de métrica usando a API do CloudWatch com `PutAnomalyDetector`, `DeleteAnomalyDetector` e`DescribeAnomalyDetectors`. Para obter informações sobre essas ações de API, consulte as seções a seguir emReferência de API do Amazon CloudWatch.

- PutAnomalyDetector
- DeleteAnomalyDetector
- DescribeAnomalyDetectors

Para obter informações sobre como os alarmes de detecção de anomalias são precificados, consulte Definição de preço do Amazon CloudWatch.

Usar matemática de métricas

A matemática métricas permite consultar várias métricas do CloudWatch e usar expressões matemáticas para criar novas séries temporais de acordo com essas métricas. Você pode visualizar as séries temporais resultantes no console do CloudWatch e adicioná-las aos painéis. Usando métricas do AWS Lambda como exemplo, você pode dividir a métrica `Errors` pela métrica `Invocations` para obter uma taxa de erro. Depois, adicione a série temporal resultante a um gráfico no painel do CloudWatch.

Você também pode executar a matemática métricas de forma programática usando a operação da API `GetMetricData`. Para obter mais informações, consulte GetMetricData.

Adicionar uma expressão matemática a um gráfico do CloudWatch

Você pode adicionar uma expressão matemática a um gráfico no painel do CloudWatch. Cada gráfico é limitado a usar um máximo de 500 métricas e expressões, para que você possa adicionar uma expressão matemática somente se o gráfico tiver 499 métricas ou menos. Isso se aplica mesmo que nem todas as métricas sejam exibidas no gráfico.

Para adicionar uma expressão matemática a um gráfico

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Crie ou edite um gráfico. Deve haver pelo menos uma métrica no gráfico.
3. Escolha Graphed metrics (Métricas em gráfico).
4. Selecione Math expression (Expressão matemática), Start with empty expression (Começar com expressão vazia). Uma nova linha é exibida para a expressão.
5. Na nova linha, na coluna Details (Detalhes), insira a expressão matemática. As tabelas na seção Sintaxe e funções de matemática métrica listam as funções que podem ser usadas na expressão.

   Para usar uma métrica ou o resultado de outra expressão como parte da fórmula para essa expressão, use o valor mostrado na coluna Id: por exemplo, m1+m2 ou e1-MIN(e1).

   Você pode alterar o valor de Id. Ela pode incluir números, letras e sublinhado e deve começar com uma letra minúscula. Alterar o valor de Id para um nome mais significativo também pode facilitar o entendimento de um gráfico: por exemplo, a alteração de m1 e m2 para errors (erros) e requests (solicitações).

   > **Tip**
   >
   > Escolha a seta para baixo ao lado de Math Expression (Expressão matemática) para ver uma lista de funções compatíveis, que podem ser usadas ao criar sua expressão.

6. Para a coluna Label (Rótulo) da expressão, insira um nome que descreva o que a expressão está calculando.

Se o resultado de uma expressão é uma matriz de séries temporais, cada uma dessas séries temporais é exibida no gráfico com uma linha separada, com cores diferentes. Imediatamente abaixo do gráfico está uma legenda para cada linha no gráfico. Para uma única expressão que produz várias séries temporais, as legendas para essas séries temporais estão no formato ***Expression-Label Metric-Label***. Por exemplo, se o gráfico inclui uma métrica com um rótulo de Erros e uma expressão FILL(METRICS(), 0) que tem um rótulo Filled With 0:, uma linha na legenda seria Filled With 0: Errors. Para que a legenda mostre apenas os rótulos de métrica originais, defina *Expression-Label* como vazio.

Quando uma expressão produz uma matriz de séries temporais no gráfico, você não pode alterar as cores usadas para cada uma dessas séries temporais.

7. Depois de adicionar as expressões desejadas, você pode simplificar o gráfico ocultando algumas das métricas originais. Para ocultar uma métrica ou expressão, desmarque a caixa de seleção à esquerda do campo Id.

Sintaxe de funções da matemática métricas

As seções a seguir explicam as funções disponíveis para a matemática métricas. Todas as funções devem ser escritas com letras maiúsculas (como AVG), e o campo Id de todas as métricas e expressões matemáticas devem começar com uma letra minúscula.

O resultado final de qualquer expressão matemática deve ser uma única série temporal ou um array de séries de tempo. Algumas funções escalares produzem um número. Você pode usar essas funções em uma função maior que, em última análise, produz uma série temporal. Por exemplo, o uso de AVG de uma única série temporal produz um número escalar, de forma que ele não pode ser o resultado final da expressão. Mas você pode usá-lo na função m1-AVG(m1) para exibir uma série temporal da diferença entre cada ponto de dados individual e o valor médio da série temporal.

Abreviações de tipos de dados

Algumas funções são válidas apenas para determinados tipos de dados. As abreviações na lista a seguir são usadas nas tabelas de funções para representar os tipos de dados compatíveis para cada função:

- S representa um número escalar, como 2, -5 ou 50,25.

- TS é uma série temporal (uma série de valores para uma única métrica do CloudWatch ao longo do tempo): por exemplo, a métrica `CPUUtilization` para a instância `i-1234567890abcdef0` para os últimos três dias.
- TS[] é uma matriz de séries temporais, como a série temporal para várias métricas.
- String[] é uma matriz de strings.

A função METRICS()

A função METRICS() retorna todas as métricas na solicitação. As expressões matemáticas não são incluídas.

Você pode usar METRICS() dentro de uma expressão maior que produz apenas uma série de tempo ou um array de séries de tempo. Por exemplo, a expressão SUM(METRICS()) retorna uma série temporal (TS) que é a soma dos valores de todas as métricas incluídas em gráfico. METRICS()/100 retorna uma matriz de séries temporais, cada um sendo é uma série de tempo que mostra cada ponto de dados de uma das métricas dividido por 100.

Você pode usar a função METRICS() com uma sequência para retornar apenas as métricas incluídas em gráfico que contêm essa sequência no campo Id. Por exemplo, a expressão SUM(METRICS("errors")) retorna uma série temporal que é a soma dos valores de todas as métricas incluídas em gráfico que contêm 'erros' no campo Id. Você também pode usar SUM([METRICS(“4xx”), METRICS(“5xx”)]) para correspondência a várias sequências.

Funções aritméticas básicas

A tabela a seguir lista as funções de aritmética básica compatíveis. Valores ausentes em uma séries temporal são tratados como 0. Se o valor de um ponto de dados fizer com que uma função tente dividir por zero, o ponto de dados será descartado.

Operation	Argumentos	Exemplos
Operadores aritméticos: + - * / ^	S, S	PERIOD(m1)/60
	S, TS	5 * m1
	TS, TS	m1 - m2
	S, TS[]	SUM(100/[m1, m2])

Operation	Argumentos	Exemplos
	TS, TS[]	AVG(METRICS()) METRICS()*100
Subtração unária -	S TS TS[]	-5*m1 -m1 SUM(-[m1, m2])

Comparação e operadores lógicos

Você pode usar operadores lógicos e de comparação com um par de séries temporais ou um par de valores escalares únicos. Quando você usa um operador de comparação com um par de séries temporais, os operadores retornam uma série temporal em que cada ponto de dados é 0 (falso) ou 1 (verdadeiro). Se você usar um operador de comparação em um par de valores escalares, um único valor escalar será retornado, 0 ou 1.

Quando operadores de comparação são usados entre duas séries temporais, e apenas uma das séries temporais tem um valor para um time stamp específico, a função trata o valor ausente na outra série cronológica como 0.

Você pode usar operadores lógicos em conjunto com operadores de comparação, para criar funções mais complexas.

A tabela a seguir lista os operadores com suporte.

Tipo de operador	Operadores compatíveis
Operadores de comparação	**==** **!=** **<=** **>=** **<**

Tipo de operador	Operadores compatíveis
	>
Operadores lógicos	E ou && OR ou ||

Para ver como esses operadores são usados, suponha que temos duas séries temporais: metric1 tem valores de `[30, 20, 0, 0]` e metric2 tem valores de `[20, -, 20, -]` nos quais `-` indica que não há valor para esse timestamp.

Expressão	Saída
(metric1 < metric2)	0, 0, 1, 0
(metric1 >= 30)	1, 0, 0, 0
(metric1 > 15 AND metric2 > 15)	1, 0, 0, 0

Funções compatíveis com a matemática métricas

A tabela a seguir descreve as funções que podem ser usadas em expressões matemáticas. Insira todas as funções em letras maiúsculas.

O resultado final de qualquer expressão matemática deve ser uma única série temporal ou um array de séries de tempo. Algumas funções em tabelas nas seções a seguir produzem um número escalar. Você pode usar essas funções em uma função maior que, em última análise, produz uma série temporal. Por exemplo, o uso de AVG de uma única série temporal produz um número escalar, de forma que ele não pode ser o resultado final da expressão. Mas você pode usá-lo na função m1-AVG(m1) para exibir uma série temporal da diferença entre cada ponto de dados individual e o valor médio desse ponto de dados.

Na tabela a seguir, cada exemplo na coluna Exemplos é uma expressão que resulta em uma única série temporal ou uma matriz de séries temporais. Isso mostra como funções que retornam números escalares podem ser usadas como parte de uma expressão válida que produz uma única série temporal.

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
ABS	TS TS[]	TS TS[]	Retorna o valor absoluto de cada ponto de dados.	ABS(m1-m2) MIN(ABS([m1, m2])) ABS(METRICS())	✓
ANOMALY_DETECTION_BAND	TS TS, S	TS[]	Retorna um segmento de detecção de anomalias para a métrica especificada. O segmento consiste em duas séries temporais, um que representa o limite superior do valor "normal" esperado da métrica e o outro que representa o limite inferior. A função pode levar dois argumentos. O primeiro é o ID da métrica para o qual o segmento vai ser criado. O segundo argumento é o número de desvios-padrão a ser usado para o segmento. Se você não especificar esse argumento, é usado o padrão de 2. Para ter mais informações, consulte Usar a detecção de anomalias do CloudWatch.	ANOMALY_DETECTION_BAND(m1) ANOMALY_DETECTION_BAND(m1,4)	

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
AVG	TS TS[]	S TS	A AVG de uma única série temporal retorna um escalar que representa a média de todos os pontos de dados na métrica. O AVG de uma matriz de séries temporais retorna uma única série temporal. Valores ausentes são tratados como 0. **Note** Recomendamos não usar essa função nos alarmes do CloudWatch se deseja que a função retorne um escalar. Por exemplo, `AVG(m2)`. Sempre que um alarme avalia se é necessário alterar o estado, o CloudWatch tenta recuperar um maior número de	SUM([m1,m2])/AVG(m2) AVG(METRICS())	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			pontos de dados do que o número especificado em Evaluation Periods (Períodos de avaliação). Essa função age de forma diferente quando dados extras são solicitados. Para usar essa função com alarmes, especialmente com ações do Auto Scaling, recomendamos que você configure o alarme para usar M dentre N pontos de dados, onde M < N.		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
CEIL	TS TS[]	TS TS[]	Retorna o teto de cada métrica. O teto é o menor valor inteiro maior ou igual a cada valor.	CEIL(m1) CEIL(METRICS()) SUM(CEIL(METRICS()))	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
DATAPOINT_COUNT	TS TS[]	S TS	Retorna uma contagem dos pontos de dados que informaram valores. Isso é útil para calcular médias de métricas esparsas. **Note** Recomendamos não usar essa função nos alarmes do CloudWatch. Sempre que um alarme avalia se é necessário alterar o estado, o CloudWatch tenta recuperar um maior número de pontos de dados do que o número especificado em Evaluation Periods (Períodos de avaliação). Essa função age de forma diferente quando dados	SUM(m1) / DATAPOINT_COUNT(m1) DATAPOINT_COUNT(METRICS())	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			extras são solicitados.		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
DB_PERF_INSIGHTS	String, string, string String, String, String[]	TS (caso seja fornecid uma única string) TS[] (caso seja fornecid uma matriz de strings)	Retorna métricas do Contador do Insights de Performance para bancos de dados como Amazon Relational Database Service e Amazon DocumentDB (compatível com MongoDB). Essa função retorna a mesma quantidade de dados que poderiam ser obtidos consultando diretamente as APIs do Insights de Performance. É possível usar essas métricas no CloudWatch para fazer representações gráficas e criar alarmes. **Important** Ao usar essa função, você deve especificar o ID exclusivo de recurso de banco de dados do banco de dados. Isso é diferente do identificador do	DB_PERF_INSIGHTS(‘RDS’, ‘db-ABCDEFGHIJKLMNOPQRSTUVWXY1’, ‘os.cpuUtilization.user.avg’) DB_PERF_INSIGHTS(‘DOCDB, ‘db-ABCDEFGHIJKLMNOPQRSTUVWXY1’, [‘os.cpuUtilization.idle.avg’, ‘os.cpuUtilization.user.max’])	

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			banco de dados. Para encontrar o ID de recurso do banco de dados no console do Amazon RDS, escolha a instância de banco de dados para visualizar os detalhes. Em seguida, escolha a guia Configuration (Configuração). O ID de recurso é exibido na seção Configuração. DB_PERF_INSIGHTS também traz a métrica `DBLoad` em intervalos inferiores a um minuto. As métricas do Performance Insights recuperadas com essa função não são armazenadas no CloudWatch. Portanto, alguns recursos do		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			CloudWatch, como observabilidade entre contas, detecção de anomalias, fluxos de métricas, explorador de métricas e Metric Insights, não funcionarão com as métricas do Insights de Performance que você recuperar com DB_PERF_INSIGHTS. Uma única solicitação usando a função DB_PERF_INSIGHTS pode recuperar os seguintes números de pontos de dados. • 1080 pontos de dados para períodos de alta resolução (1 s, 10 s, 30 s) • 1440 pontos de dados para períodos de resolução padrão (1 m, 5 m, 1 h, 1 d) A função DB_PERF_INSIGHTS oferece suporte somente para os períodos a seguir:		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			• 1 segundo • 10 segundos • 30 segundos • 1 minuto • 5 minutos • 1 hora • 1 dia Para obter mais informações sobre as métricas de contador do Insights de Performance do Amazon RDS, consulte Métricas de contador do Insights de Performance. Para obter mais informações sobre as métricas de contador do Insights de Performance do Amazon DocumentDB, consulte Métricas de contador do Insights de Performance. **Note** Métricas de alta resolução com granularidade de menos		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			de um minuto recuperadas pelo DB_PERF_INSIGHTS são aplicáveis somente à métrica DBLoad ou às métricas do sistema operacional caso você tenha ativado o monitoramento aprimorado em uma resolução maior. Para obter mais informações sobre o monitoramento avançado do Amazon RDS, consulte Monitoramento de métricas do SO com monitoramento avançado. É possível criar um alarme de alta resolução usando a função		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			DB_PERF_INSIGHTS por um período máximo de três horas. É possível usar o console do CloudWatch para representar graficamente as métricas recuperadas com a função DB_PERF_INSIGHTS para qualquer intervalo de tempo.		
DIFF	TS TS[]	TS TS[]	Retorna a diferença entre cada valor na série temporal e o valor anterior dessa série temporal.	DIFF(m1)	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
DIFF_TIME	TS TS[]	TS TS[]	Retorna a diferença, em segundos, entre o carimbo de data/hora de cada valor da série temporal e o carimbo de data/hora do valor anterior dessa série temporal.	DIFF_TIME(METRICS())	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
FILL	TS, [S | REPEAT | LINEAR] TS[], [TS | S | REPEAT | LINEAR]	TS TS[]	Preenche os valores ausentes de uma série temporal. Há várias opções para os valores a serem usados para preencher os valores ausentes: • É possível especificar um valor a ser usado como o valor de preenchimento. • É possível especificar uma métrica a ser usada como o valor de preenchimento. • Use a palavra-chave REPEAT para preencher valores ausentes com o valor real mais recente da métrica antes do valor ausente. • Use a palavra-chave LINEAR para preencher os valores ausentes com valores que criam uma interpolação linear entre os valores do início e do fim da lacuna.	FILL(m1,10) FILL(METRICS(), 0) FILL(METRICS(), m1) FILL(m1, MIN(m1)) FILL(m1, REPEAT) FILL(METRICS(), LINEAR)	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			Note Ao usar essa função em um alarme, você poderá encontrar um problema, se suas métricas estiverem sendo publicadas com um pequeno atraso e se o minuto mais recente nunca teve dados. Neste caso, FILL substitui esse ponto de dados ausente pelo valor solicitado. Isso faz com que o ponto de dados mais recente da métrica seja sempre o valor de preenchimento, o que pode resultar no bloqueio do alarme no estado		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			OK ou no estado ALARM. É possível contornar isso usando um alarme M de N. Para ter mais informações, consulte Avaliar um alarme.		
FIRST LAST	TS[]	TS	Retorna a primeira ou a última série temporal de uma matriz de séries temporais. Isso é útil quando usado com a função SORT. Ele também pode ser usado para obter os limites superior e inferior da função ANOMALY_DETECTION_BAND.	IF(FIRST(SORT(METRICS(), AVG, DESC))>100, 1, 0) examina a métrica superior de uma matriz, que é classificada por AVG. Depois, ele retorna 1 ou 0 para cada ponto de dados, dependendo se esse valor de ponto de dados é superior a 100. LAST(ANOMALY_DETECTION_BAND(m1)) retorna o limite superior da faixa de previsão de anomalia.	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
FLOOR	TS TS[]	TS TS[]	Retorna o piso de cada métrica. O piso é o maior valor inteiro menor ou igual a cada valor.	FLOOR(m1) FLOOR(METRICS())	✓
IF	Expressão IF	TS	Use IF junto com um operador de comparação para filtrar pontos de dados de uma série temporal ou criar uma série temporal mista composta por várias séries temporais agrupadas. Para ter mais informações, consulte Usar expressões IF.	Para ver exemplos, consulte Usar expressões IF.	✓
INSIGHT_RULE_METRIC	INSIGHT_RULE_METRIC(ruleName, metricName)	TS	Use INSIGHT_RULE_METRIC para extrair estatísticas de uma regra no Contributor Insights. Para ter mais informações, consulte Criar gráfico de métricas geradas por regras.		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
LAMBDA	LAMBDAmbdaFunctionName [, optionalarg]*)	TS TS{}	Chama uma função do Lambda para consultar métricas de uma fonte de dados que não seja o CloudWatch. Para ter mais informações, consulte Como passar argumentos para sua função do Lambda.		
LOG	TS TS[]	TS TS[]	O LOG de uma série temporal retorna o valor logaritmo natural de cada valor na série temporal.	LOG(METRICS())	✓
LOG10	TS TS[]	TS TS[]	O LOG10 de uma série temporal retorna o valor logaritmo de base 10 de cada valor na série temporal.	LOG10(m1)	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
MAX	TS TS[]	S TS	O MAX de uma única série temporal retorna um escalar que representa o valor máximo de todos os pontos de dados na métrica. Se você inserir uma matriz de séries temporais, a função MAX criará e retornará uma série temporal que consiste no valor mais alto para cada ponto de dados em comparação com as séries temporais que foram usadas como entrada. **Note** Recomendamos não usar essa função nos alarmes do CloudWatch se deseja que a função retorne um escalar. Por exemplo, `MAX(m2)` sempre que um	MAX(m1)/m1 MAX(METRICS())	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			alarme avalia se é necessário alterar o estado, o CloudWatch tenta recuperar um maior número de pontos de dados do que o número especificado como Períodos de avaliação. Essa função age de forma diferente quando dados extras são solicitados.		
METRIC_COUNT	TS[]	S	Retorna o número de métricas na matriz de séries temporais.	m1/METRIC_COUNT(METRICS())	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
METRICS	nulo string	TS[]	A função METRICS() retorna todas as métricas do CloudWatch na solicitação. As expressões matemáticas não são incluídas. Você pode usar METRICS() dentro de uma expressão maior que produz apenas uma série de tempo ou um array de séries de tempo. Você pode usar a função METRICS() com uma sequência para retornar apenas as métricas incluídas em gráfico que contêm essa sequência no campo Id. Por exemplo, a expressão SUM(METRICS("errors")) retorna uma série temporal que é a soma dos valores de todas as métricas incluídas em gráfico que contêm 'erros' no campo Id. Você também pode usar SUM([METRICS(“4xx”	AVG(METRICS()) SUM(METRICS("errors"))	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			), METRICS(“5xx”)]) para correspondência a várias sequências.		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
MIN	TS TS[]	S TS	O MIN de uma única série temporal retorna um escalar que representa o valor mínimo de todos os pontos de dados na métrica. Se você inserir uma matriz de séries temporais, a função MIN criará e retornará uma série temporal que consiste no valor mais baixo para cada ponto de dados em comparação com as séries temporais que foram usadas como entrada. Se você inserir uma matriz de séries temporais, a função MIN criará e retornará uma série temporal que consiste no valor mais alto para cada ponto de dados em comparação com as séries temporais que foram usadas como entrada.	m1-MIN(m1) MIN(METRICS())	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			Note Recomendamos não usar essa função nos alarmes do CloudWatch se deseja que a função retorne um escalar. Por exemplo, `MIN(m2)` sempre que um alarme avalia se é necessário alterar o estado, o CloudWatch tenta recuperar um maior número de pontos de dados do que o número especificado como Períodos de avaliação. Essa função age de forma diferente quando dados extras são solicitados.		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?

MINUTE
HOUR
DAY
DATE
MONTH
YEAR
EPOCH
	TS	TS	Essas funções obtêm o período e o intervalo da série temporal e retornam uma nova série temporal não esparsa onde cada valor é baseado em seu carimbo de data/hora.
			• MINUTE retorna uma série de tempo não esparsa de inteiros entre 0 e 59 que representam o minuto UTC de cada carimbo de data/hora da série temporal original.
			• HOUR retorna uma série de tempo não esparsa de inteiros entre 0 e 23 que representam a hora UTC de cada carimbo de data/hora da série temporal original.
			• DAY retorna uma série de tempo não esparsa de inteiros entre 1 e 7 que representam o dia da semana UTC de cada carimbo de data/hora
				MINUTE(m1)
				IF(DAY(m1)<6,m1) retorna métricas somente de dias úteis, de segunda a sexta-feira.
				IF(MONTH(m1) == 4,m1) retorna somente métricas publicadas em abril.
					✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			da série temporal original, em que 1 representa segunda-feira e 7 representa o domingo. • DATE retorna uma série de tempo não esparsa de inteiros entre 1 e 31 que representam o dia do mês UTC de cada carimbo de data/hora da série temporal original. • MONTH retorna uma série de tempo não esparsa de inteiros entre 1 e 12 que representam o mês UTC de cada carimbo de data/hora da série temporal original, em que 1 representa janeiro e 12 representa dezembro. • YEAR retorna uma série de tempo não esparsa de inteiros que representam o ano UTC de cada carimbo de data/hora		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			da série temporal original. • EPOCH retorna uma série de tempo não esparsa de inteiros que representam o tempo UTC, em segundos, desde o início da época, de cada carimbo de data/hora da série temporal original. A época é 1.º de janeiro de 1970.		
PERIOD	TS	S	Retorna o período da métrica em segundos. Entrada válida são métricas, não os resultados de outras expressões.	m1/PERIOD(m1)	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
RATE	TS TS[]	TS TS[]	Retorna a taxa de alteração da métrica por segundo. Isso é calculado como a diferença entre o último ponto de dados e o valor anterior, dividido pelo valor do ponto de dados a diferença de tempo em segundos entre os dois valores. **Important** Definir alarmes em expressões que usam a função RATE em métricas com dados esparsos pode resultar em um comportamento imprevisível, porque o intervalo de pontos de dados obtidos ao avaliar o alarme pode variar com base na última publicaçã	RATE(m1) RATE(METRICS())	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			o dos pontos de dados.		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
REMOVE_EMPTY	TS[]	TS[]	Remove todas as séries temporais que não tenham pontos de dados de uma matriz de séries temporais. O resultado é uma matriz de séries temporais em que cada série temporal contém pelo menos um ponto de dados. **Note** Recomendamos não usar essa função nos alarmes do CloudWatch. Sempre que um alarme avalia se é necessário alterar o estado, o CloudWatch tenta recuperar um maior número de pontos de dados do que o número especificado em Evaluation Periods (Períodos de	REMOVE_EMPTY(METRICS())	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			avaliação). Essa função age de forma diferente quando dados extras são solicitados.		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
RUNNING_SUM	TS TS[]	TS TS[]	Retorna uma série temporal com a soma dos valores na série temporal original. **Note** Recomendamos não usar essa função nos alarmes do CloudWatch. Sempre que um alarme avalia se é necessário alterar o estado, o CloudWatch tenta recuperar um maior número de pontos de dados do que o número especificado em Evaluation Periods (Períodos de avaliação). Essa função age de forma diferente quando dados	RUNNING_SUM([m1,m2])	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			extras são solicitados.		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
SEARCH	Expressão de pesquisa	Uma ou mais TS	Retorna uma ou mais séries temporais que correspondem aos critérios de pesquisa que você especificar. A função SEARCH permite que você adicione várias séries temporais relacionadas a um gráfico com uma expressão. O gráfico é atualizado dinamicamente para incluir novas métricas que serão adicionadas posteriormente e correspondem aos critérios de pesquisa. Para ter mais informações, consulte Usar expressões de pesquisa em gráficos. Não é possível criar um alarme com base em uma expressão SEARCH. Isso ocorre porque as expressões de pesquisa retornam várias séries temporais, e um alarme baseado em uma expressão matemática pode		✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			observar apenas uma série temporal. Se você fez login em uma conta de monitoramento na observabilidade entre contas do CloudWatch, a função SEARCH encontrará métricas nas contas de origem e na conta de monitoramento.		
SERVICE_QUOTA	TS que é uma métrica de uso	TS	Retorna a cota de serviço da métrica de uso determinada. É possível usar isso para visualizar como seu uso atual se compara à cota e para definir alarmes que avisam quando você se aproxima da cota. Para ter mais informações, consulte Métricas de uso do AWS.		✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
SLICE	(TS[], S, S) ou (TS[], S)	TS[] TS	Recupera parte de uma matriz de séries temporais. Isso é especialmente útil quando combinado com SORT. Por exemplo, você pode excluir o resultado superior de uma matriz de séries temporais. Você pode usar dois argumentos escalares para definir o conjunto de séries temporais a serem retornados. Os dois escalares definem o início (inclusivo) e o fim (exclusivo) da matriz a serem retornados. A matriz é indexada por zero, portanto, a primeira série temporal na matriz é a série temporal 0. Como alternativa, você pode especificar apenas um valor, e o CloudWatch retornará todas as séries temporais começando com esse valor.	SLICE(SORT(METRICS(), SUM, DESC), 0, 10) retorna as 10 métricas da matriz na solicitação que têm o valor SUM mais alto. SLICE(SORT(METRICS(), AVG, ASC), 5) classifica a matriz de métricas pela estatística AVG e retorna todas as séries temporais, exceto as 5 com o menor AVG.	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
SORT	(TS[], FUNCT SORT_(R) (TS[], FUNCT SORT_(R, S)	TS[]	Classifica uma matriz de séries temporais de acordo com a função especificada. A função que você usa pode ser AVG, MIN, MAX ou SUM. A ordem de classificação pode ser ASC para ordem crescente (valores mais baixos primeiro) ou DESC para classificar os valores mais altos primeiro. Você também pode especificar um número após a ordem de classificação que atua como um limite. Por exemplo, especific ar um limite de 5 retorna apenas as 5 principai s séries temporais da classificação. Quando essa função matemática for exibida em um gráfico, os rótulos para cada métrica no gráfico também serão classific ados e numerados.	SORT(METRICS(), AVG, DESC, 10) calcula o valor médio de cada série temporal, classifica as séries temporais com os valores mais altos no início da classificação e retorna apenas as 10 séries temporais com as médias mais altas. SORT(METRICS(), MAX, ASC) classifica a matriz de métricas pela estatística MAX e retorna todas elas em ordem crescente.	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
STDDEV	TS TS[]	S TS	O STDDEV de uma única série temporal retorna um escalar que representa o desvio padrão de todos os pontos de dados na métrica. O STDDEV de uma matriz de séries temporais retorna uma única série temporal. **Note** Recomendamos não usar essa função nos alarmes do CloudWatch se deseja que a função retorne um escalar. Por exemplo, `STDDEV(m2)` sempre que um alarme avalia se é necessário alterar o estado, o CloudWatch tenta recuperar um maior número de pontos de dados do que	m1/STDDEV(m1) STDDEV(METRICS())	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			o número especificado como Períodos de avaliação . Essa função age de forma diferente quando dados extras são solicitados.		

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
SUM	TS TS[]	S TS	A SUM de uma única série temporal retorna um escalar que representa a soma dos valores de todos os pontos de dados na métrica. O SUM de uma matriz de séries temporais retorna uma única série temporal. **Note** Recomendamos não usar essa função nos alarmes do CloudWatch se deseja que a função retorne um escalar. Por exemplo, `SUM(m1)`. Sempre que um alarme avalia se é necessário alterar o estado, o CloudWatch tenta recuperar um maior número de pontos de dados do que	SUM(METRICS())/SUM(m1) SUM([m1,m2]) SUM(METRICS("errors"))/SUM(METRICS("requests"))*100	✓

Função	Argumentos	Tipo de retorno*	Descrição	Exemplos	Aceito entre contas?
			o número especificado em Evaluation Periods (Períodos de avaliação). Essa função age de forma diferente quando dados extras são solicitados.		
TIME_SERIES	S	TS	Retorna uma série temporal não esparsa na qual cada valor é definido como um argumento escalar.	TIME_SERIES(MAX(m1)) TIME_SERIES(5*AVG(m1)) TIME_SERIES(10)	✓

*Não é permitido usar apenas uma função que retorne um número escalar, pois todos os resultados finais de expressões devem ser uma única série temporal ou uma matriz de séries temporais. Em vez disso, use essas funções como parte de uma expressão maior que retorne uma série temporal.

Usar expressões IF

Use IF junto com um operador de comparação para filtrar pontos de dados de uma série temporal ou criar uma série temporal mista composta por várias séries temporais agrupadas.

IF usa os seguintes argumentos:

```
IF(condition, trueValue, falseValue)
```

A condição será avaliada como FALSE se o valor do ponto de dados da condição for 0 e TRUE se o valor da condição for qualquer outro valor, se esse valor for positivo ou negativo. Se a condição for uma série temporal, ela será avaliada separadamente para cada time stamp.

O seguinte lista as sintaxes válidas. Para cada uma destas sintaxes, a saída é uma única série temporal.

- IF(TS ***Operador de comparação*** S, S | TS, S | TS)

  > **Note**
  >
  > Se `TS comparison operator S` for TRUE, mas `metric2` não tiver um ponto de dados correspondente, o resultado será 0.

- IF(TS, TS, TS)
- IF(TS, S, TS)
- IF(TS, TS, S)
- IF(TS, S, S)
- IF(S, TS, TS)

As seções a seguir fornecem mais detalhes e exemplos para essas sintaxes.

IF(TS ***operador de comparação*** S, scalar2 | metric2, scalar3 | metric3)

O valor da série de tempo de saída correspondente:

- tem o valor de scalar2 ou metric2, se TS ***Operador de comparação*** S for TRUE
- tem o valor de scalar3 ou metric3, se TS ***Operador de comparação*** S for FALSE
- tem o valor de 0 se o ***Operador de comparação*** TS for TRUE e o ponto de dados correspondente em metric2 não existir.
- tem o valor de 0 se o ***Operador de comparação*** TS for FALSE e o ponto de dados correspondente em metric3 não existir.
- é uma série de tempo vazia, se o ponto de dados correspondente não existir em metric3, ou se scalar3/metric3 for omitido da expressão

IF(metric1, metric2, metric3)

Para cada ponto de dados da metric1, o valor da série temporal de saída correspondente:

- tem o valor de metric2, se o ponto de dados correspondente de metric1 for TRUE.
- tem o valor de metric3, se o ponto de dados correspondente de metric1 for FALSE.
- tem o valor de 0, se o ponto de dados correspondente de metric1 for TRUE e o ponto de dados correspondente não existir em metric2.
- é descartado, se o ponto de dados correspondente de metric1 for FALSE e o ponto de dados correspondente não existir em metric3
- é descartado, se o ponto de dados correspondente de metric1 for FALSE e metric3 for omitido da expressão.
- é descartado se o ponto de dados correspondente em metric1 não existir.

A tabela a seguir mostra um exemplo para essa sintaxe.

Métrica ou função	Valores
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
(metric3)	[0, 0, 20, -, 20]
IF(metric1, metric2, metric3)	[30, 0, 20, 0, -]

IF(metric1, scalar2, metric3)

Para cada ponto de dados da metric1, o valor da série temporal de saída correspondente:

- tem o valor de scalar2, se o ponto de dados correspondente de metric1 for TRUE.
- tem o valor de metric3, se o ponto de dados correspondente de metric1 for FALSE.
- é descartado, se o ponto de dados correspondente de metric1 for FALSE e o ponto de dados correspondente não existir em metric3, ou se metric3 for omitido da expressão.

Métrica ou função	Valores
(metric1)	[1, 1, 0, 0, -]
scalar2	5
(metric3)	[0, 0, 20, -, 20]
IF(metric1, scalar2, metric3)	[5, 5, 20, -, -]

IF(metric1, metric2, scalar3)

Para cada ponto de dados da metric1, o valor da série temporal de saída correspondente:

- tem o valor de metric2, se o ponto de dados correspondente de metric1 for TRUE.
- tem o valor de scalar3, se o ponto de dados correspondente de metric1 for FALSE.
- tem o valor de 0, se o ponto de dados correspondente de metric1 for TRUE e o ponto de dados correspondente não existir em metric2.
- é descartado se o ponto de dados correspondente em metric1 não existir.

Métrica ou função	Valores
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
scalar3	5
IF(metric1, metric2, scalar3)	[30, 0, 5, 5, -]

IF(scalar1, metric2, metric3)

O valor da série de tempo de saída correspondente:

- tem o valor de metric2, se scalar1 for TRUE.
- tem o valor de metric3, se scalar1 for FALSE.

- é uma série temporal vazia, se metric3 for omitido da expressão.

Exemplos de caso de uso para expressões IF

Os exemplos a seguir ilustram os possíveis usos da função IF.

- Para exibir somente os valores baixos de uma métrica:

  IF(metric1<400, metric1)
- Para alterar cada ponto de dados de uma métrica para um de dois valores, para mostrar valores altos e mínimos relativos da métrica original:

  IF(metric1 < 400, 10, 2)
- Para exibir um 1 para cada time stamp em que a latência está acima do limite e exibir um 0 para todos os outros pontos de dados:

  IF(latency>threshold, 1, 0)

Usar a matemática métrica com a operação da API GetMetricData

Você pode usar `GetMetricData` para executar cálculos usando expressões matemáticas, bem como para recuperar grandes lotes de dados de métricas em uma chamada de API. Para obter mais informações, consulte GetMetricData.

Detecção de anomalias em matemática métrica

A detecção de anomalias em matemática métrica é um recurso que você pode usar para criar alarmes de detecção de anomalias em métricas únicas e nas saídas de expressões matemáticas métricas. É possível usar essas expressões para criar gráficos de visualização de bandas de detecção de anomalias. O recurso suporta funções aritméticas básicas, comparações e operadores lógicos e a maioria das outras funções.

A detecção de anomalias em matemática métrica não suporta as seguintes funções:

- Expressões que contenham mais de um ANOMALY_DETECTION_BAND na mesma linha.
- Expressões que contenham mais de 10 métricas ou expressões matemáticas.
- Expressões que contenham a expressão METRICS.
- Expressões que contenham a função SEARCH.

- Expressões que usam a função DP_PERF_INSIGHTS.
- Expressões que usem métricas com períodos diferentes.
- Detectores de anomalias da matemática de métricas que usam métricas de alta resolução como entrada.

Para obter mais informações sobre esse recurso, consulte Usar a detecção de anomalias do CloudWatch no Guia do usuário do Amazon CloudWatch.

Usar expressões de pesquisa em gráficos

As expressões de pesquisa são um tipo de expressão matemática que você pode adicionar aos gráficos do CloudWatch. As expressões de pesquisa permitem que você adicione rapidamente várias métricas relacionadas a um gráfico. Elas também permitem que você crie gráficos dinâmicos que adicionam automaticamente as métricas apropriadas à exibição, mesmo se essas métricas não existirem ao criar o gráfico.

Por exemplo, você pode criar uma expressão de pesquisa que exibe a métrica da `AWS/EC2` `CPUUtilization` para todas as instâncias na região. Se, posteriormente, você executar uma nova instância, o `CPUUtilization` da nova instância é adicionado automaticamente ao gráfico.

Quando você usa uma expressão de pesquisa em um gráfico, a pesquisa localiza a expressão de pesquisa em nomes de métricas, namespaces, nomes de dimensão e valores de dimensão. Você pode usar operadores boolianos para pesquisas mais complexas e eficazes. Uma expressão de pesquisa pode encontrar somente métricas que relataram dados nas últimas duas semanas.

Não é possível criar um alarme com base na expressão SEARCH. Isso ocorre porque as expressões de pesquisa retornam várias séries temporais, e um alarme baseado em uma expressão matemática pode observar apenas uma série temporal.

Se você estiver usando uma conta de monitoramento na observabilidade entre contas do CloudWatch, as expressões de pesquisa poderão encontrar métricas nas contas de origem vinculadas a essa conta de monitoramento.

Tópicos

- Sintaxe de expressão de pesquisa do CloudWatch
- Exemplos de expressão de pesquisa do CloudWatch
- Criar um gráfico do CloudWatch com uma expressão de pesquisa

Sintaxe de expressão de pesquisa do CloudWatch

Uma expressão de pesquisa válida tem o formato a seguir.

```
SEARCH(' {Namespace, DimensionName1, DimensionName2, ...} SearchTerm', 'Statistic')
```

Por exemplo:

```
SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization"', 'Average')
```

- A primeira parte da consulta depois da palavra `SEARCH`, entre chaves, é o metric schema (esquema de métrica) a ser pesquisado. O esquema de métrica contém um namespace de métrica e um ou mais nomes de dimensão. A inclusão de um esquema de métrica em uma consulta de pesquisa é opcional. Se especificado, o esquema de métrica deve conter um namespace e pode conter um ou mais nomes de dimensão que são válidos nesse namespace.

  Você não precisa usar aspas dentro do esquema de métrica, a menos que um nome de dimensão ou namespace inclua espaços ou caracteres não alfanuméricos. Nesse caso, você deve colocar o nome que contém esses caracteres entre aspas duplas.
- O `SearchTerm` também é opcional, mas uma pesquisa válida deve conter o esquema de métrica, o `SearchTerm` ou ambos. O `SearchTerm` geralmente contém um ou mais IDs de métrica, nomes de métrica ou valores de dimensão. O `SearchTerm` pode incluir vários termos para serem pesquisados, tanto por correspondência parcial como por correspondência exata. Ele também pode conter operadores boolianos.

  Usar um ID de conta em um `SearchTerm` só funciona em contas configuradas como contas de monitoramento para a observabilidade entre contas do CloudWatch. A sintaxe de um ID de conta no `SearchTerm` é `:aws.AccountId = "`*`444455556666`*`"`. Você também pode usar `'LOCAL'` para especificar a própria conta de monitoramento: `:aws.AccountId = 'LOCAL'`

  Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

  O `SearchTerm` pode incluir um ou mais designadores, como `MetricName=`, conforme este exemplo, mas o uso de designadores não é obrigatório.

  O esquema de métrica e `SearchTerm` devem estar juntos entre aspas simples.
- O `Statistic` é o nome de qualquer estatística válida do CloudWatch. Ele deve ser colocado entre aspas simples. Para ter mais informações, consulte Estatísticas.

O exemplo anterior pesquisa o namespace `AWS/EC2` para as métricas que têm `InstanceId` como nome de uma dimensão. Ele retorna todas as métricas `CPUUtilization` que encontra, com o gráfico exibindo a estatística `Average`.

Uma expressão de pesquisa pode encontrar somente métricas que relataram dados nas últimas duas semanas.

Limites da expressão de pesquisa

O tamanho máximo da consulta da expressão de pesquisa é de 1024 caracteres. Você pode ter até 100 expressões de pesquisa em um gráfico. Um gráfico pode exibir até 500 séries temporais.

Expressões de pesquisa do CloudWatch: tokenização

Quando você especifica um `SearchTerm`, a função de pesquisa busca tokens, que são substrings geradas automaticamente pelo CloudWatch de todos os nomes de métrica, nomes de dimensão, valores de dimensão e namespaces. O CloudWatch gera tokens diferenciados pela capitalização camel-case na string original. Os caracteres numéricos também são usados como o início de novos tokens, e os caracteres não alfanuméricos funcionam como delimitadores, criando tokens antes e depois dos caracteres não alfanuméricos.

Uma string contínua do mesmo tipo do caractere delimitador de token resulta em um token.

Todos os tokens são gerados em letras minúsculas. A tabela a seguir mostra alguns exemplos de tokens gerados.

String original	Tokens gerados
CustomCount1	`customcount1` , `custom`, `count`, `1`
SDBFailure	`sdbfailure` , `sdb`, `failure`
Project2-trial333	`project2trial333` , `project`, `2`, `trial`, `333`

Expressões de pesquisa do CloudWatch: correspondências parciais

Quando você especifica um `SearchTerm`, o termo de pesquisa também é tokenizado. O CloudWatch localiza métricas com base em correspondências parciais, que são correspondências

de um único token gerado a partir do termo de pesquisa para um único token gerado a partir de um namespace, nome ou valor da dimensão ou nome da métrica.

As pesquisas de correspondência parcial de um único token não distinguem letras maiúsculas de minúsculas. Por exemplo, o uso de qualquer um dos seguintes termos de pesquisa pode retornar a métrica `CustomCount1`:

- `count`
- `Count`
- `COUNT`

No entanto, o uso de `couNT` como um termo de pesquisa não localiza `CustomCount1`, pois a capitalização no termo de pesquisa `couNT` é tokenizada em `cou` e `NT`.

As pesquisas também podem corresponder a tokens compostos, que são vários tokens que aparecem consecutivamente no nome original. Para corresponder a um token composto, a pesquisa diferencia maiúsculas de minúsculas. Por exemplo, se o termo original for `CustomCount1`, pesquisas por `CustomCount` ou `Count1` serão bem-sucedidas, mas pesquisas por `customcount` ou `count1` não.

Expressões de pesquisa do CloudWatch: correspondências exatas

Você pode definir uma pesquisa para localizar apenas correspondências exatas do termo de pesquisa usando aspas duplas ao redor da parte do termo de pesquisa que requer uma correspondência exata. Essas aspas duplas são inseridas entre as aspas simples usadas em torno de todo o termo de pesquisa. Por exemplo, **`SEARCH(' {MyNamespace}, "CustomCount1" ', 'Maximum')`** localizará a string exata `CustomCount1` se ela existir como um nome da métrica, nome ou valor da dimensão no namespace chamado `MyNamespace`. No entanto, as pesquisas **`SEARCH(' {MyNamespace}, "customcount1" ', 'Maximum')`** ou **`SEARCH(' {MyNamespace}, "Custom" ', 'Maximum')`** não localizam essa string.

Você pode combinar termos de correspondência parcial e termos de correspondência exata em uma única expressão de pesquisa. Por exemplo, **`SEARCH(' {AWS/NetworkELB, LoadBalancer} "ConsumedLCUs" OR flow ', 'Maximum')`** retorna a métrica do Elastic Load Balancing chamada `ConsumedLCUs` e todas as métricas ou dimensões do Elastic Load Balancing que contenham o token `flow`.

O uso da correspondência exata também é uma boa maneira de localizar nomes com caracteres especiais, como caracteres não alfanuméricos ou espaços, conforme o exemplo a seguir.

```
SEARCH(' {"My Namespace", "Dimension@Name"}, "Custom:Name[Special_Characters" ',
 'Maximum')
```

Expressões de pesquisa do CloudWatch: excluir um esquema de métrica

Todos os exemplos mostrados até agora incluem um esquema de métrica entre chaves. As pesquisas que omitem um esquema de métrica também são válidas.

Por exemplo, **`SEARCH(' "CPUUtilization" ', 'Average')`** retorna todos os nomes de métrica, nomes de dimensão, valores de dimensão e namespaces que são uma correspondência exata para a string `CPUUtilization`. Nos namespaces de métrica da AWS, isso pode incluir métricas de vários produtos, incluindo Amazon EC2, Amazon ECS, SageMaker, entre outros.

Para restringir essa pesquisa a apenas um serviço da AWS, a prática recomendada é especificar o namespace e todas as dimensões necessárias no esquema de métrica, como no exemplo a seguir. Embora isso restrinja a pesquisa ao namespace `AWS/EC2`, ela ainda retornaria resultados de outras métricas se você tivesse definido `CPUUtilization` como um valor da dimensão para essas métricas.

```
SEARCH(' {AWS/EC2, InstanceType} "CPUUtilization" ', 'Average')
```

Como alternativa, você pode adicionar o namespace no `SearchTerm` como no exemplo a seguir. No entanto, neste exemplo, a pesquisa corresponderia a qualquer string `AWS/EC2`, mesmo se ela fosse um valor ou nome de dimensão personalizados.

```
SEARCH(' "AWS/EC2" MetricName="CPUUtilization" ', 'Average')
```

Expressões de pesquisa do CloudWatch: especificar nomes de propriedade na pesquisa

A pesquisa de correspondência exata por `"CustomCount1"` a seguir retorna todas as métricas exatamente com esse nome.

```
SEARCH(' "CustomCount1" ', 'Maximum')
```

No entanto, ela também retorna métricas com nomes de dimensão, valores de dimensão ou namespaces de `CustomCount1`. Para estruturar ainda mais sua pesquisa, você pode especificar

o nome de propriedade do tipo de objeto que deseja localizar nas pesquisas. O exemplo a seguir pesquisa todos os namespaces e retorna métricas chamadas `CustomCount1`.

```
SEARCH(' MetricName="CustomCount1" ', 'Maximum')
```

Você também pode usar pares nome/valor de dimensão e namespaces como nomes de propriedade, conforme os exemplos a seguir. O primeiro desses exemplos também ilustra que você pode usar nomes de propriedade com pesquisas de correspondência parcial.

```
SEARCH(' InstanceType=micro ', 'Average')
```

```
SEARCH(' InstanceType="t2.micro" Namespace="AWS/EC2" ', 'Average')
```

Expressões de pesquisa do CloudWatch: caracteres não alfanuméricos

Os caracteres não alfanuméricos servem como delimitadores e marcam onde os nomes de métricas, as dimensões, os namespaces e os termos de pesquisa devem ser separados em tokens. Quando os termos são tokenizados, os caracteres não alfanuméricos são removidos e não aparecem nos tokens. Por exemplo, `Network-Errors_2` gera os tokens `network`, `errors`, e `2`.

O termo de pesquisa pode incluir qualquer caractere não alfanumérico. Se esses caracteres aparecerem no termo de pesquisa, eles poderão especificar tokens compostos em uma correspondência parcial. Por exemplo, todas as pesquisas a seguir localizariam métricas chamadas `Network-Errors-2` ou `NetworkErrors2`.

```
network/errors
network+errors
network-errors
Network_Errors
```

Quando você estiver fazendo uma pesquisa de valor exato, qualquer caractere não alfanumérico usado na pesquisa exata deverá ser o caractere correto que aparece na string que está sendo pesquisada. Por exemplo, se você quiser encontrar `Network-Errors-2`, a pesquisa por `"Network-Errors-2"` será bem-sucedida, mas uma pesquisa por `"Network_Errors_2"` não.

Quando você realizar uma pesquisa de correspondência exata, os caracteres a seguir deverão ser recuados com uma barra invertida.

```
" \ ( )
```

Por exemplo, para encontrar o nome da métrica `Europe\France Traffic(Network)` por correspondência exata, use o termo de pesquisa **`"Europe\\France Traffic\(Network\)"`**

Expressões de pesquisa do CloudWatch: operadores boolianos

A pesquisa oferece suporte ao uso de operadores boolianos `AND`, `OR` e `NOT` no `SearchTerm`. Os operadores boolianos são inseridos entre aspas simples usadas ao redor de todo o termo de pesquisa. Os operadores boolianos fazem distinção de maiúsculas e minúsculas, portanto `and`, `or` e `not` não são válidos como operadores boolianos.

Você pode usar o `AND` explicitamente na pesquisa, como o **`SEARCH('{AWS/EC2,InstanceId} network AND packets', 'Average')`**. Não usar nenhum operador booliano entre os termos de pesquisa vai fazer com que eles sejam implicitamente pesquisados como se houvesse um operador do `AND`, portanto, o **`SEARCH(' {AWS/EC2,InstanceId} network packets ', 'Average')`** gera os mesmos resultados de pesquisa.

Use `NOT` para excluir subconjuntos de dados dos resultados. Por exemplo, o **`SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT i-1234567890123456 ', 'Average')`** retorna o `CPUUtilization` para todas as suas instâncias, exceto para a instância do `i-1234567890123456`. Você também pode usar uma cláusula `NOT` como o único termo de pesquisa. Por exemplo, o **`SEARCH( 'NOT Namespace=AWS ', 'Maximum')`** gera todas as suas métricas personalizadas (métricas com namespaces que não incluem o `AWS`).

Você pode usar várias expressões `NOT` em uma consulta. Por exemplo, o **`SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT "ProjectA" NOT "ProjectB" ', 'Average')`** retorna o `CPUUtilization` de todas as instâncias na região, exceto para aquelas com valores de dimensão do `ProjectA` ou do `ProjectB`.

Você pode combinar operadores boolianos para realizar pesquisas mais eficientes e detalhadas, conforme os exemplos a seguir. Use parênteses para agrupar os operadores.

Os dois exemplos a seguir retornam todos os nomes de métricas que contêm `ReadOps` de ambos os namespaces do EC2 e do EBS.

```
SEARCH(' (EC2 OR EBS) AND MetricName=ReadOps ', 'Maximum')
```

```
SEARCH(' (EC2 OR EBS) MetricName=ReadOps ', 'Maximum')
```

O exemplo a seguir restringe a pesquisa anterior a somente os resultados que incluem `ProjectA`, o que pode ser o valor de uma dimensão.

```
SEARCH(' (EC2 OR EBS) AND ReadOps AND ProjectA ', 'Maximum')
```

O exemplo a seguir usa agrupamentos aninhados. Ele retorna métricas do Lambda para `Errors` de todas as funções, e `Invocations` de funções com nomes que incluem as strings `ProjectA` ou `ProjectB`.

```
SEARCH(' {AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average')
```

Expressões de pesquisa do CloudWatch: usar expressões matemáticas

Você pode usar uma expressão de pesquisa dentro de expressões matemáticas em um gráfico.

Por exemplo, **`SUM(SEARCH(' {AWS/Lambda, FunctionName} MetricName="Errors" ', 'Sum'))`** retorna a soma da métrica `Errors` de todas as funções do Lambda.

O uso de linhas separadas para a expressão de pesquisa e a expressão matemática pode gerar resultados mais úteis. Por exemplo, suponha que você use as duas expressões a seguir em um gráfico. A primeira linha exibe linhas `Errors` separadas para cada uma das funções do Lambda. O ID dessa expressão é `e1`. A segunda linha adiciona outra linha que mostra a soma dos erros de todas as funções.

```
SEARCH(' {AWS/Lambda, FunctionName}, MetricName="Errors" ', 'Sum')
SUM(e1)
```

Exemplos de expressão de pesquisa do CloudWatch

Os exemplos a seguir ilustram mais sintaxe e usos de expressões de pesquisa. Vamos começar com uma pesquisa por `CPUUtilization` em todas as instâncias na região e examinar variações.

Este exemplo exibe uma linha para cada instância na região, mostrando a métrica `CPUUtilization` do namespace `AWS/EC2`.

```
SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" ', 'Average')
```

A alteração de `InstanceId` para `InstanceType` muda o gráfico para mostrar uma linha para cada tipo de instância usado na região. Os dados de todas as instâncias de cada tipo são agregados em uma linha para esse tipo de instância.

```
SEARCH(' {AWS/EC2,InstanceType} MetricName="CPUUtilization" ', 'Average')
```

O exemplo a seguir agrega o `CPUUtilization` por tipo de instância e exibe uma linha para cada tipo de instância que inclui a string `micro`.

```
SEARCH('{AWS/EC2,InstanceType} InstanceType=micro MetricName="CPUUtilization" ',
 'Average')
```

Este exemplo restringe o exemplo anterior, alterando o `InstanceType` para uma pesquisa exata por instâncias t2.micro.

```
SEARCH('{AWS/EC2,InstanceType} InstanceType="t2.micro" MetricName="CPUUtilization" ',
 'Average')
```

A pesquisa a seguir remove a parte `{metric schema}` da consulta, portanto, a métrica `CPUUtilization` de todos os namespaces aparece no gráfico. Isso pode retornar diversos resultados, pois o gráfico inclui várias linhas para a métrica `CPUUtilization` de cada serviço da AWS agregados em diferentes dimensões.

```
SEARCH('MetricName="CPUUtilization" ', 'Average')
```

Para restringir um pouco esses resultados, você pode especificar dois namespaces de determinadas métricas.

```
SEARCH('MetricName="CPUUtilization" AND ("AWS/ECS" OR "AWS/ES") ', 'Average')
```

O exemplo anterior é a única maneira de fazer uma pesquisa de vários namespaces específicos com uma consulta de pesquisa, já que você pode especificar apenas um esquema de métrica em cada consulta. No entanto, para adicionar mais estrutura, você pode usar duas consultas no gráfico, conforme o exemplo a seguir. Este exemplo também adiciona mais estrutura especificando uma dimensão a ser usada para agregar os dados para o Amazon ECS.

```
SEARCH('{AWS/ECS ClusterName}, MetricName="CPUUtilization" ', 'Average')
```

```
SEARCH(' {AWS/EBS} MetricName="CPUUtilization" ', 'Average')
```

O exemplo a seguir retorna a métrica do Elastic Load Balancing chamada `ConsumedLCUs` e todas as métricas ou dimensões do Elastic Load Balancing que contenham o token `flow`.

```
SEARCH('{AWS/NetworkELB, LoadBalancer} "ConsumedLCUs" OR flow ', 'Maximum')
```

O exemplo a seguir usa agrupamentos aninhados. Ele retorna métricas do Lambda para `Errors` de todas as funções, e `Invocations` de funções com nomes que incluem as strings `ProjectA` ou `ProjectB`.

```
SEARCH('{AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average')
```

O exemplo a seguir exibe todas as suas métricas personalizadas, excluindo métricas geradas por serviços da AWS.

```
SEARCH('NOT Namespace=AWS ', 'Average')
```

O exemplo a seguir exibe métricas com nomes de métricas, namespaces, nomes de dimensão e valores de dimensão que contêm a string `Errors` como parte do nome.

```
SEARCH('Errors', 'Average')
```

O exemplo a seguir restringe essa pesquisa a correspondências exatas. Por exemplo, essa pesquisa localiza o nome da métrica `Errors`, mas não métricas chamadas `ConnectionErrors` ou `errors`.

```
SEARCH(' "Errors" ', 'Average')
```

O exemplo a seguir mostra como especificar nomes que contêm espaços ou caracteres especiais na parte do esquema de métrica do termo de pesquisa.

```
SEARCH('{"Custom-Namespace", "Dimension Name With Spaces"}, ErrorCount ', 'Maximum')
```

Exemplos de expressão de pesquisa da observabilidade entre contas do CloudWatch

Exemplos da observabilidade entre contas do CloudWatch

Se você fez login em uma conta configurada como uma conta de monitoramento na observabilidade entre contas do CloudWatch, poderá usar a função SEARCH para retornar as métricas das contas de origem especificadas. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

O exemplo a seguir recupera todas as métricas do Lambda da conta com o ID de conta 111122223333.

```
SEARCH(' AWS/Lambda :aws.AccountId = "111122223333" ', 'Average')
```

O exemplo a seguir recupera todas as métricas da AWS/EC2 de duas contas: 111122223333 e 777788889999.

```
SEARCH(' AWS/EC2 :aws.AccountId = ("111122223333" OR "777788889999") ', 'Average')
```

O exemplo a seguir recupera todas as métricas da AWS/EC2 da conta de origem 111122223333 e da própria conta de monitoramento.

```
SEARCH(' AWS/EC2 :aws.AccountId = ("111122223333" OR 'LOCAL') ', 'Average')
```

O exemplo a seguir recupera a SUM da métrica MetaDataToken da conta 444455556666 com a dimensão InstanceId.

```
SEARCH('{AWS/EC2,InstanceId} :aws.AccountId=444455556666 MetricName=\"MetadataNoToken
\"','Sum')
```

Criar um gráfico do CloudWatch com uma expressão de pesquisa

No console do CloudWatch, você poderá acessar o recurso de pesquisa quando adicionar um gráfico a um painel, ou usando a visualização Metrics (Métricas).

Não é possível criar um alarme com base em uma expressão SEARCH. Isso ocorre porque as expressões de pesquisa retornam várias séries temporais, e um alarme baseado em uma expressão matemática pode observar apenas uma série temporal.

Para adicionar um gráfico com uma expressão de pesquisa a um painel existente

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Dashboards (Painéis) e selecione um painel.

3. Escolha Add widget (Adicionar widget).
4. Selecione Line (Linha) ou Stacked area (Área empilhada) e depois selecione Configure (Configurar).
5. Na guia Graphed metrics (Métricas em gráfico), selecione Add a math expression (Adicionar uma expressão matemática).
6. Para Details (Detalhes), insira a expressão de pesquisa que você deseja. Por exemplo, **`SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization"', 'Average')`**
7. (Opcional) Para adicionar outra expressão de pesquisa ou expressão matemática ao gráfico, selecione Add a math expression (Adicionar uma expressão matemática)
8. (Opcional) Depois de adicionar uma expressão de pesquisa, você pode especificar um rótulo dinâmico para aparecer na legenda do gráfico para cada métrica. Os rótulos dinâmicos exibem uma estatística sobre a métrica e são atualizados automaticamente quando o painel ou o gráfico é atualizado. Para adicionar um rótulo dinâmico, escolha Graphed metrics (Métricas em gráfico) e Dynamic labels (Rótulos dinâmicos).

   Por padrão, os valores dinâmicos que você adiciona ao rótulo aparecem no início do rótulo. Em seguida, você pode clicar no valor do Label (Rótulo) para a métrica para editar o rótulo. Para ter mais informações, consulte Usar rótulos dinâmicos.
9. (Opcional) Para adicionar uma única métrica ao gráfico, selecione a guia All metrics (Todas as métricas) e analise a métrica desejada.
10. (Opcional) Para alterar o período mostrado no gráfico, selecione custom (personalizar) na parte superior do gráfico, ou um dos períodos à esquerda de custom (personalizar).
11. (Opcional) As anotações horizontais ajudam os usuários do painel a ver rapidamente quando uma métrica aumentou bastante até determinado nível ou se a métrica está dentro de um intervalo predefinido. Para adicionar uma anotação horizontal, selecione Graph options (Opções do gráfico) e Add horizontal annotation (Adicionar anotação horizontal):

    a. Em Label (Rótulo), insira um rótulo para a anotação.
    b. Em Value (Valor), insira o valor da métrica em que a anotação horizontal aparece.
    c. Em Fill, especifique se o preenchimento do sombreamento deve ser usado com essa anotação. Por exemplo, selecione `Above` ou `Below` para a área correspondente ser preenchida. Se você especificar `Between`, outro campo `Value` será exibido e a área do gráfico entre os dois valores será preenchida.
    d. Em Axis (Eixos), especifique se os números em `Value` se referem à métrica associada ao eixo Y esquerdo ou direito, caso o gráfico inclua várias métricas.

Você pode alterar a cor de preenchimento de uma anotação, escolhendo a cor no quadrado de cores na coluna à esquerda da anotação.

Repita essas etapas para adicionar várias anotações horizontais ao mesmo gráfico.

Para ocultar uma anotação, desmarque a caixa de seleção na coluna da esquerda para essa anotação.

Para excluir uma anotação, selecione x na coluna Actions (Ações).

12. (Opcional) Anotações verticais ajudam você a marcar as etapas em um gráfico, como eventos operacionais ou o início e o fim de uma implantação. Para adicionar uma anotação vertical, selecione Graph options (Opções de gráfico) e Add vertical annotation (Adicionar anotação vertical):

    a. Em Label (Rótulo), insira um rótulo para a anotação. Para exibir apenas a data e hora na anotação, mantenha o campo Label (Rótulo) em branco.
    b. Para Date (Data), especifique a data e hora onde a anotação vertical aparece.
    c. Para Fill (Preencher), especifique se deseja usar preenchimento do sombreamento antes ou depois de uma anotação vertical, ou entre duas anotações verticais. Por exemplo, selecione `Before` ou `After` para a área correspondente ser preenchida. Se você especificar `Between`, outro campo `Date` será exibido e a área do gráfico entre os dois valores será preenchida.

    Repita essas etapas para adicionar várias anotações verticais ao mesmo gráfico.

    Para ocultar uma anotação, desmarque a caixa de seleção na coluna da esquerda para essa anotação.

    Para excluir uma anotação, selecione x na coluna Actions (Ações).

13. Selecione Create widget (Criar widget).
14. Escolha Save dashboard (Salvar painel).

Para usar a visualização Metrics (Métricas) para criar gráficos de métricas pesquisadas

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.

3. No campo de pesquisa, insira os tokens a serem pesquisados: por exemplo, **`cpuutilization t2.small`**.

   Os resultados que correspondem à pesquisa aparecem.

4. Para criar um gráfico de todas as métricas que correspondem à pesquisa, selecione Graph search (Criar gráfico da pesquisa).

   ou

   Para refinar a pesquisa, escolha um dos namespaces que apareceram nos resultados de pesquisa.

5. Se tiver selecionado um namespace para refinar os resultados, você poderá fazer o seguinte:

   a. Para criar um gráfico de uma ou mais métricas, marque a caixa de seleção ao lado de cada métrica. Para selecionar todas as métricas, marque a caixa de seleção na linha de cabeçalho da tabela.
   b. Para refinar a pesquisa, passe o mouse sobre o nome de uma métrica e selecione Add to search (Adicionar à pesquisa) ou Search for this only (Pesquisar apenas por isto).
   c. Para visualizar a ajuda para uma métrica, selecione o nome da métrica e escolha What is this? (O que é isso?).

   As métricas selecionadas aparecem no gráfico.

6. (Opcional) Selecione um dos botões na barra de pesquisa para editar a parte do termo de pesquisa.
7. (Opcional) Para adicionar o gráfico a um painel, selecione Actions (Ações) e Add to dashboard (Adicionar ao painel).

Obter estatísticas de uma métrica

Definições de estatísticas do CloudWatch

Estatísticas são agregações de dados de métrica ao longo de períodos especificados. Ao representar em gráficos ou recuperar as estatísticas de uma métrica, especifique a propriedade Period (Período), como cinco minutos, usando-a para calcular cada valor estatístico. Por exemplo, se Period for cinco minutos, Sum será a soma de todos os valores de amostra coletados durante o período de cinco minutos, enquanto o Minimum será o menor valor coletado no período de cinco minutos.

O CloudWatch oferece suporte às estatísticas de métricas a seguir.

- SampleCount é o número de pontos de dados no período.
- Sum é a soma dos valores de todos os pontos de dados coletados no período.
- Average (Média) é o valor de `Sum/SampleCount` durante o período especificado.
- Minimum (Mínimo) é valor mais baixo observado durante o período especificado.
- Maximum (Máximo) é valor mais alto observado durante o período especificado.
- O Percentil (p) indica a posição relativa de um valor no conjunto de dados. Por exemplo, p95 é o 95.º percentil e significa que 95% dos dados desse período são inferiores a esse valor e 5% são superiores a esse valor. Percentis ajudam você a ter uma melhor compreensão da distribuição de seus dados de métrica.
- Média aparada (TM) é a média de todos os valores que estão entre dois limites especificados. Os valores que ultrapassam os limites são ignorados quando a média é calculada. Defina os limites como um ou dois números entre 0 e 100, de até 10 casas decimais. Os números podem ser valores absolutos ou porcentagens. Por exemplo, tm90 calcula a média após a remoção dos 10% dos pontos de dados com os valores mais altos. TM (2%:98%) calcula a média depois de remover os 2% dos pontos de dados mais baixos e 2% dos pontos de dados mais altos. TM (150:1000) calcula a média depois de remover todos os pontos de dados que são menores ou iguais a 150 ou maiores que 1000.
- Média interquartil (IQM) é a média aparada do intervalo interquartil, ou 50% do meio dos valores. É equivalente a TM (25%:75%).
- A média winsorizada (WM) é semelhante à média aparada. Porém, com a média winsorizada, os valores que estão fora do limite não são ignorados, sendo considerados iguais ao valor na borda do limite apropriado. Após essa normalização, calcula-se a média. Defina os limites como um ou dois números entre 0 e 100, de até 10 casas decimais. Por exemplo, wm98 calcula a média enquanto trata os 2% dos valores mais altos para ser igual ao valor do 98.º percentil. WM (10%:90%) calcula a média enquanto trata os 10% dos pontos de dados mais altos como o valor do limite de 90%, e tratando os 10% dos pontos de dados mais baixoscomo o valor do limite de 10%.
- A classificação de percentis (PR) é a porcentagem de valores que atendem a um limite fixo. Por exemplo, PR(:300) retorna a porcentagem de pontos de dados que têm um valor de 300 ou menos. PR (100:2000) retorna a porcentagem de pontos de dados que têm um valor entre 100 e 2000.

  A classificação do percentil é exclusiva no limite inferior e inclusiva no limite superior.

- Contagem aparada (TC) é o número de pontos de dados no intervalo escolhido para uma estatística de média aparada. Por exemplo, tc90 retorna o número de pontos de dados que não incluem quaisquer pontos de dados que se enquadram nos 10% dos valores mais altos. A TC (0.005:0.030) retorna o número de pontos de dados com valores entre 0,005 (exclusive) e 0,030 (inclusive).
- Soma aparada (TS) é a soma de valores de pontos de dados no intervalo escolhidos para uma estatística de média aparada. Equivale a (Média aparada) * (Contagem aparada). Por exemplo, tc90 retorna a soma de pontos de dados que não incluem quaisquer pontos de dados que se enquadram nos 10% dos valores mais altos. TS (80%:) retorna a soma dos valores dos pontos de dados, não incluindo quaisquer pontos de dados com valores entre os 80% mais baixos do intervalo de valores.

> **Note**
>
> Para média aparada, contagem aparada, soma aparada e média winsorizada, se você definir dois limites como valores fixos em vez de porcentagens, o cálculo incluirá valores iguais ao limite superior, mas não incluirá valores iguais ao limite inferior.

Sintaxe

Para média aparada, contagem aparada, soma aparada e média Winsorizada, aplicam-se as seguintes regras de sintaxe:

- Usar parênteses com um ou dois números com sinais de porcentagem define os limites a serem usados como os valores no conjunto de dados que se enquadram entre os dois percentis especificados. Por exemplo, TM (10%:90%) usa apenas os valores entre os percentis 10 e 90. TM (:95%) usa os valores da extremidade mais baixa dos dados configurados até o 95.º percentil, ignorando os 5% dos pontos de dados com os valores mais altos.
- Usar parênteses com um ou dois números sem sinais de porcentagem define os limites a serem usados como os valores no conjunto de dados que se enquadram entre os valores explícitos que você especificar. Por exemplo, TC (80:500) usa apenas os valores que estão entre 80 (exclusive) e 500 (inclusive). TC (:0,5) usa apenas os valores iguais a 0,5 ou menores.
- O uso de um número sem parênteses calcula usando porcentagens, ignorando pontos de dados que são maiores do que o percentil especificado. Por exemplo, tm99 calcula a média enquanto ignora o 1% dos pontos de dados com o valor mais alto. É o mesmo que TM(:99%).

- Média aparada, contagem aparada, soma aparada e média winsorizada podem ser abreviadas usando letras maiúsculas ao especificar um intervalo, como TM (5%:95%), TM (100:200) ou TM(:95%). Elas só podem ser abreviadas usando letras minúsculas quando você especifica apenas um número, como tm99.

Casos de uso das estatísticas

- Trimmed mean (Média aparada) é mais útil para métricas com um tamanho de amostra grande, como latência de página da Web. Por exemplo, tm99 ignora valores aberrantes extremamente altos que poderiam resultar de problemas de rede ou erros humanos, para fornecer um número mais preciso para a latência média de solicitações típicas. Da mesma forma, TM(10%:) não considera os 10% de valores de latência mais baixos, como os resultantes de acertos de cache. E TM(10%:99%) exclui os dois tipos de valores aberrantes. Recomendamos que você use média aparada para monitorar a latência.
- É recomendável manter o controle da contagem aparada sempre que você estiver usando média aparada, para garantir que o número de valores que estão sendo usados em seus cálculos médios aparados são suficientes para ser estatisticamente significativo.
- A classificação de percentil permite que você coloque valores em “compartimentos” de intervalos, e é possível usar isso para criar manualmente um histograma. Para fazer isso, divida seus valores em vários compartimentos, como PR(:1), PR(1:5), PR(5:10) e PR(10:). Coloque cada um desses compartimentos em uma visualização como gráficos de barras, e você terá um histograma.

  A classificação do percentil é exclusiva no limite inferior e inclusiva no limite superior.

Percentis versus média aparada

Um percentil, como p99, e uma média aparada, como tm99, medem valores semelhantes, mas não idênticos. Tanto p99 como tm99 ignoram o 1% dos pontos de dados com os valores mais altos, que são considerados valores aberrantes. Depois disso, p99 é o valor máximo dos 99% restantes, enquanto tm99 é a média dos 99% restantes. Se você estiver observando a latência de solicitações da Web, p99 diz-lhe a pior experiência do cliente, ignorando valores aberrantes, enquanto tm99 informa a experiência média do cliente, ignorando valores aberrantes.

A média aparada é uma boa estatística de latência a observar para quem está procurando otimizar a experiência do cliente.

Requisitos para usar percentis, média aparada e algumas outras estatísticas

O CloudWatch precisa dos pontos de dados brutos para calcular as seguintes estatísticas:

- Percentis
- Média aparada
- Média interquartil
- Média winsorizada
- Soma aparada
- Contagem aparada
- Classificação por percentil

Se publicar dados para estatísticas personalizadas usando um conjunto de estatísticas em vez de dados brutos, você só poderá recuperar esses tipos de estatísticas para esses dados se uma das seguintes condições for verdadeira:

- O valor SampleCount do conjunto de estatísticas é 1 e Min, Max e Sum são todos iguais.
- Min e Max são iguais, e Sum é igual a Min multiplicado por SampleCount.

Os seguintes exemplos de produtos da AWS incluem métricas compatíveis com esses tipos de estatísticas.

- API Gateway
- Application Load Balancer
- Amazon EC2
- Elastic Load Balancing
- Kinesis
- Amazon RDS

Além disso, esse tipo de estatística não está disponível para métricas quando qualquer um dos valores de métrica são números negativos.

Os exemplos a seguir mostram como obter estatísticas de métricas do CloudWatch para seus recursos, como suas instâncias do EC2.

Exemplos

- Obter estatísticas para um recurso específico
- Agregar estatísticas entre recursos
- Agregar estatísticas por grupo de Auto Scaling
- Agregar estatísticas por imagem de máquina da Amazon (AMI)

Obter estatísticas para um recurso específico

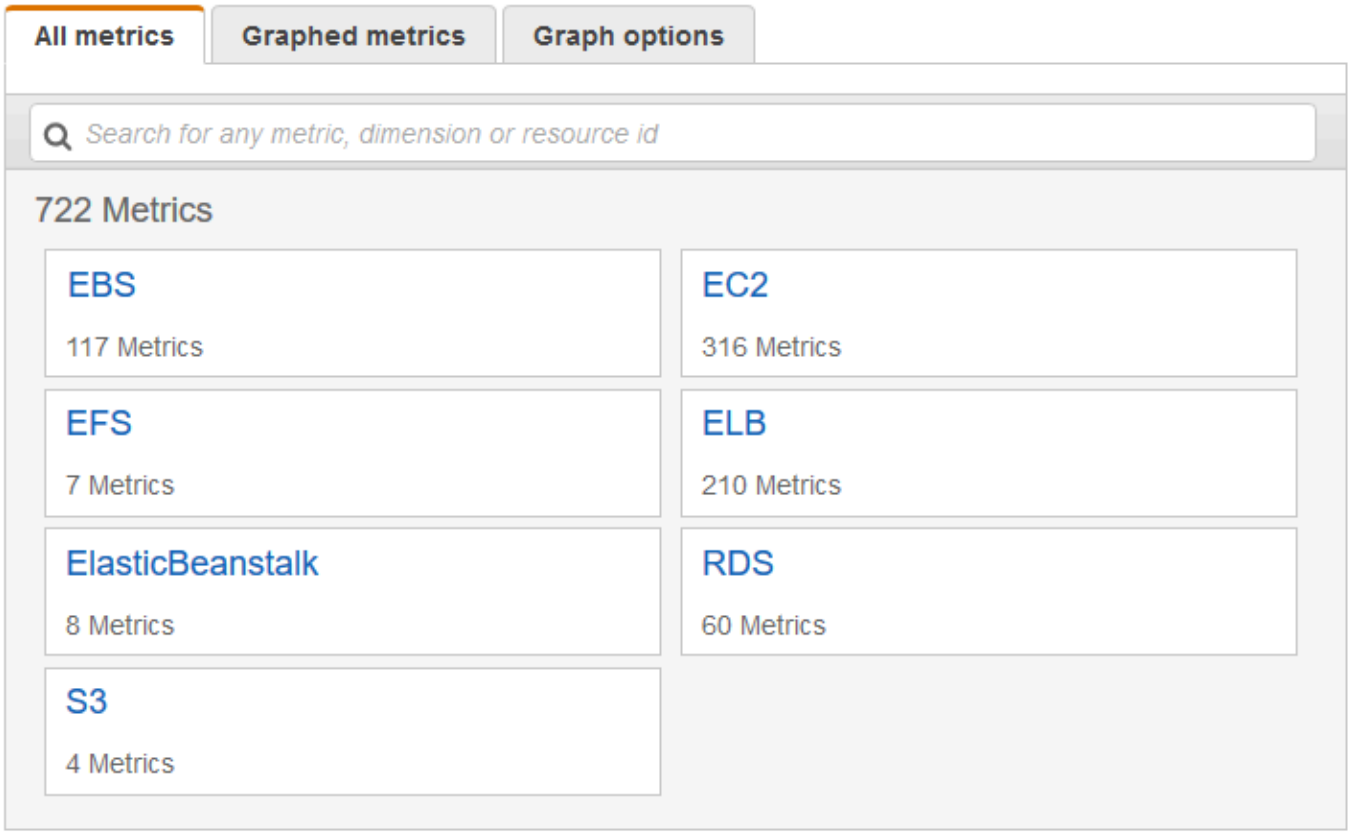
O exemplo a seguir mostra como determinar a utilização máxima de CPU de uma determinada instância do EC2.

Requisitos

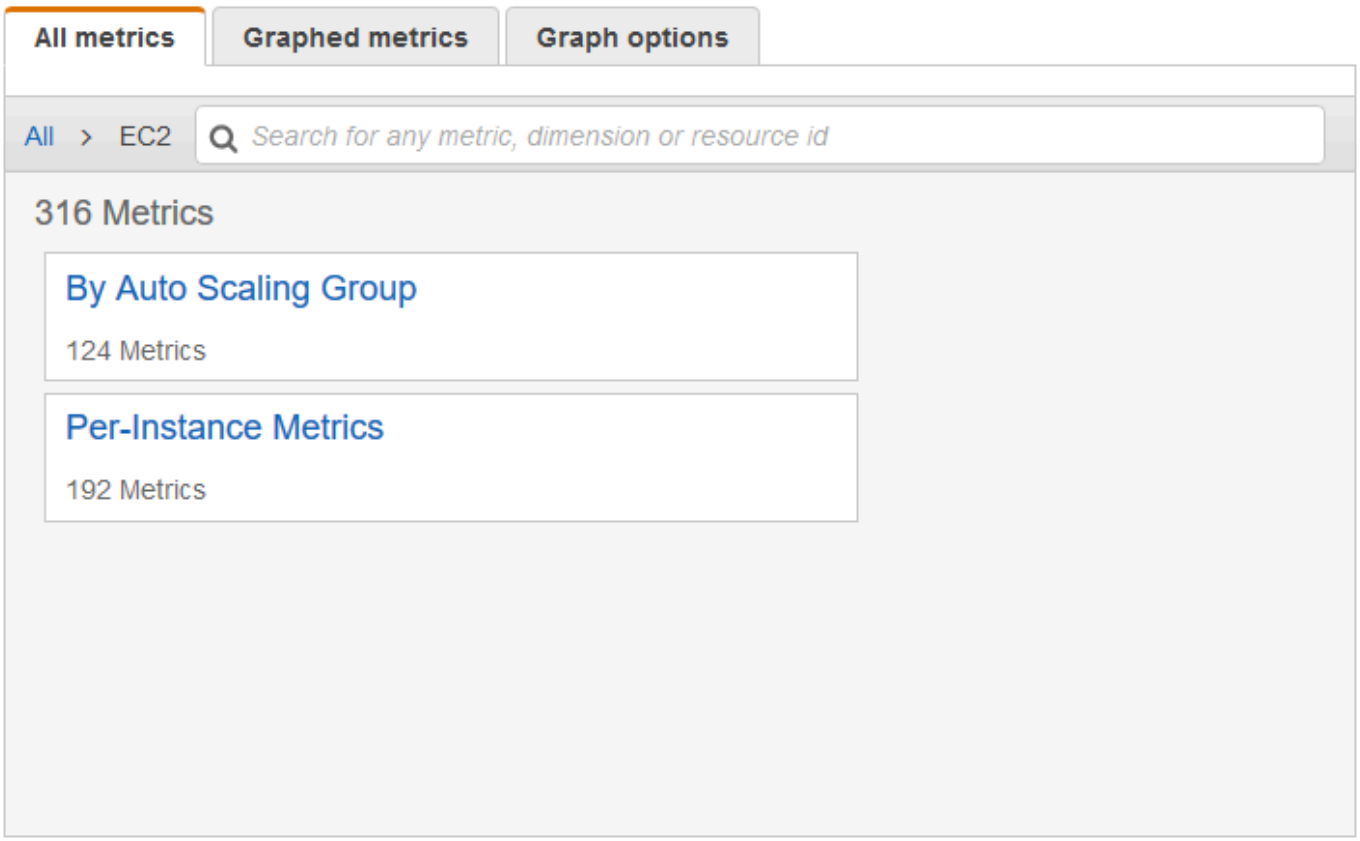
- É necessário ter o ID da instância. É possível obter o ID da instância usando o console do Amazon EC2 ou o comando describe-instances.
- Por padrão, o monitoramento básico é ativado, mas é possível habilitar o monitoramento detalhado. Para obter mais informações, consulte Habilitar ou desabilitar o monitoramento detalhado de instâncias no Manual do usuário do Amazon EC2 para instâncias do Linux.

Para exibir a média de utilização da CPU para uma instância específica usando o console

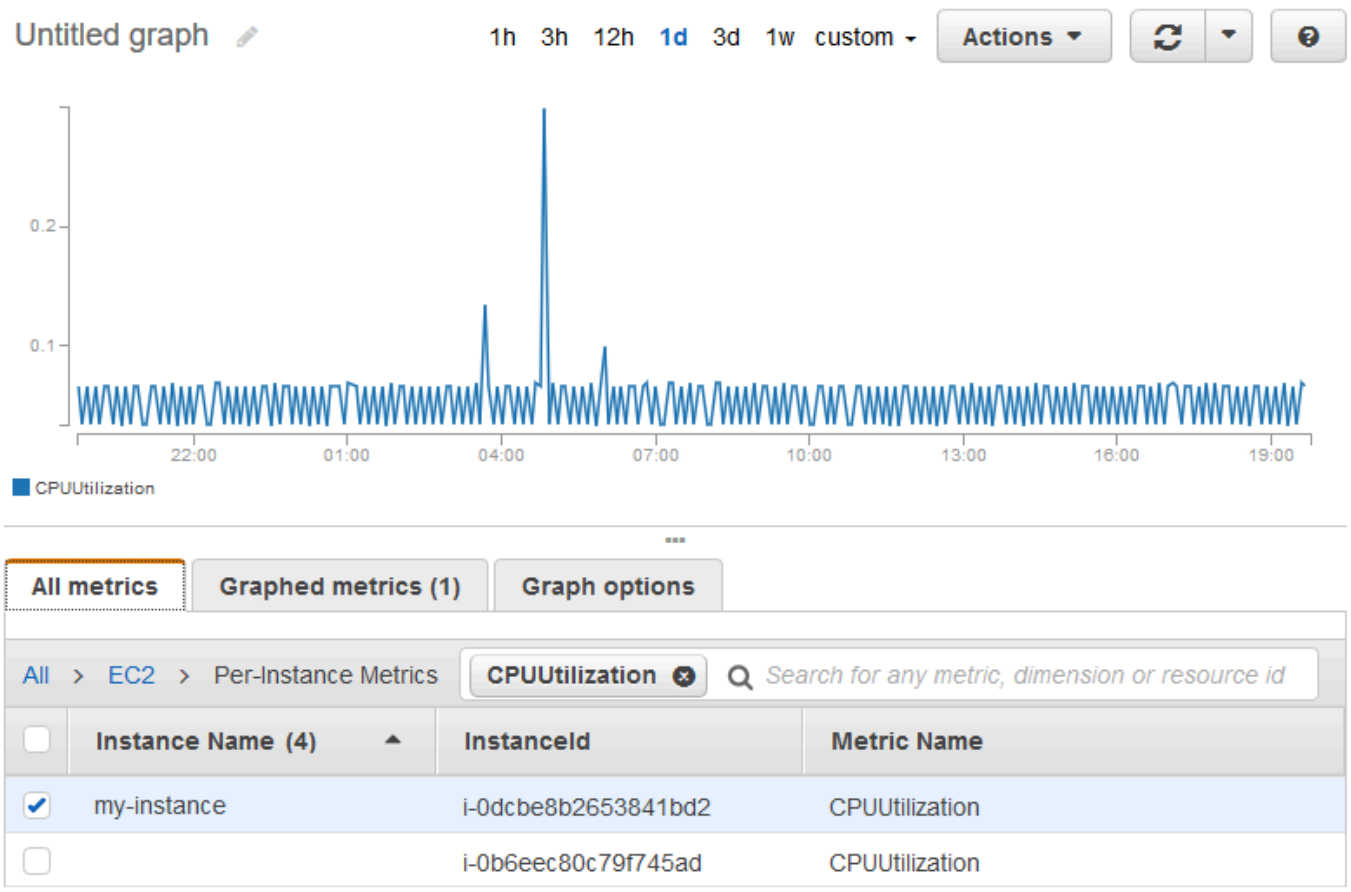
1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Selecione o namespace de métrica do EC2.



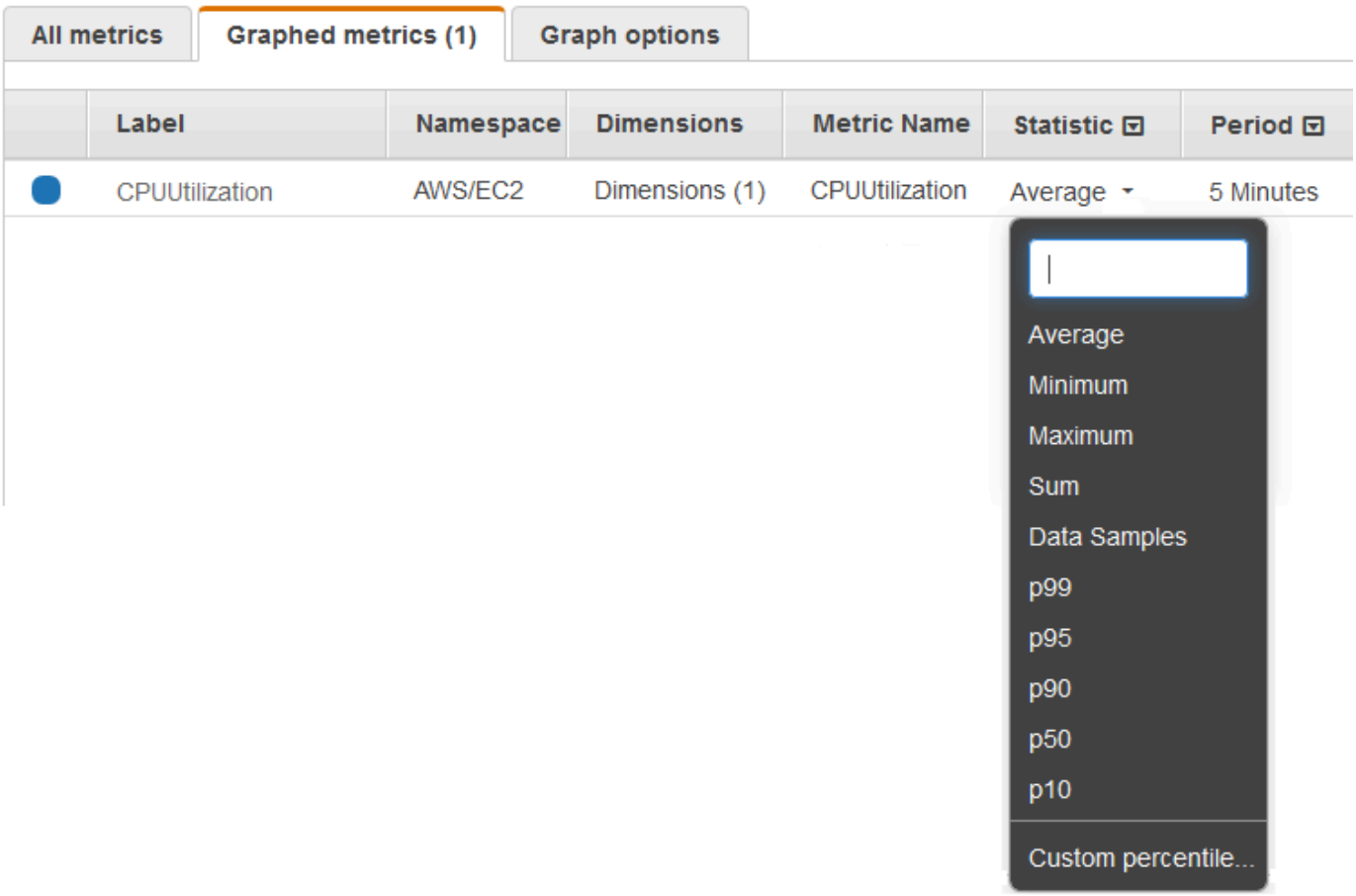
4. Selecione a dimensão Per-Instance Metrics (Métricas por instância).



5. No campo de pesquisa, digite **`CPUUtilization`** e pressione Enter. Selecione a linha para a instância específica, que exibe um gráfico da métrica `CPUUtilization` para a instância. Para alterar o nome do gráfico, escolha o ícone de lápis. Para alterar o período, selecione um dos valores predefinidos ou escolha custom (personalizado).



6. Para alterar a estatística, escolha a guia Métricas em gráfico. Escolha o cabeçalho da coluna ou um valor individual e escolha uma das estatísticas ou percentis predefinidos ou especifique um percentil personalizado (por exemplo, **`p99.999`**).



7. Para alterar o período, escolha a guia Métricas em gráfico. Escolha o cabeçalho de coluna ou um valor individual e, em seguida, escolha um valor diferente.

Para obter a utilização da CPU por instância do EC2 usando a AWS CLI

Use o comando get-metric-statistics conforme indicado a seguir para obter a métrica `CPUUtilization` para a instância especificada.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=InstanceId,Value=i-1234567890abcdef0 --statistics Maximum \
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00 --period 360
```

As estatísticas retornadas são valores de seis minutos para o intervalo de 24 horas solicitado. Cada valor representa a porcentagem de utilização máxima da CPU para a instância especificada por determinado período de seis minutos. Os pontos de dados não são retornados em ordem cronológica. A tabela a seguir mostra o início da saída de exemplo (a saída completa inclui pontos de dados para cada seis minutos do período de 24 horas).

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Agregar estatísticas entre recursos

Você pode agregar as métricas de recursos da AWS entre vários recursos. As métricas são completamente separadas entre regiões, mas é possível usar a matemática métricas para agregar métricas semelhantes entre regiões. Para ter mais informações, consulte Usar matemática de métricas.

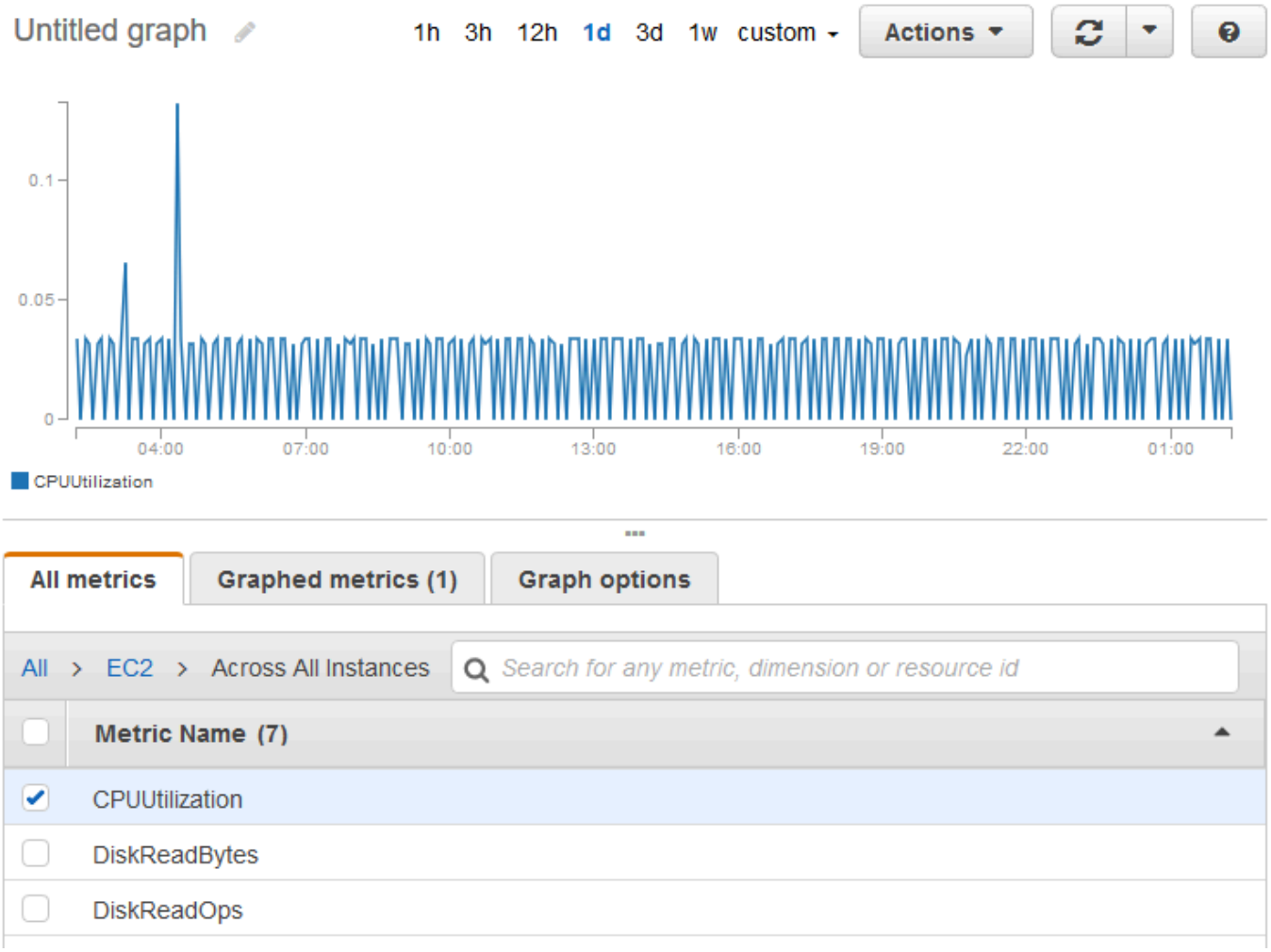
Por exemplo, você pode agregar estatísticas para suas instâncias EC2 com monitoramento detalhado habilitado. As instâncias que usam o monitoramento básico não são incluídas. Portanto, você deve habilitar o monitoramento detalhado (a um custo adicional), que fornece os dados em períodos de 1 minuto. Para obter mais informações, consulte Habilitar ou desabilitar o monitoramento detalhado de instâncias no Manual do usuário do Amazon EC2 para instâncias do Linux.

Este exemplo mostra como obter o uso médio da CPU para suas instâncias EC2. Como nenhuma dimensão é especificada, o CloudWatch retorna estatísticas para todas as dimensões no namespace `AWS/EC2`. Para obter estatísticas de outras métricas, consulte Produtos da AWS que publicam métricas do CloudWatch.

> **Important**
>
> Essa técnica para recuperar todas as dimensões em um namespace da AWS não funciona para namespaces personalizados que você publicar no CloudWatch. Com namespaces personalizados, especifique o conjunto completo de dimensões associadas a um determinado ponto de dados para recuperar estatísticas que incluam o ponto de dados.

Para exibir a utilização média da CPU para suas instâncias do EC2

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Escolha o namespace EC2 e selecione Em todas as instâncias.
4. Selecione a linha que contém `CPUUtilization`, que exibe um gráfico da métrica para todas as suas instâncias do EC2. Para alterar o nome do gráfico, escolha o ícone de lápis. Para alterar o período, selecione um dos valores predefinidos ou escolha custom (personalizado).



5. Para alterar a estatística, escolha a guia Métricas em gráfico. Escolha o cabeçalho da coluna ou um valor individual e escolha uma das estatísticas ou percentis predefinidos ou especifique um percentil personalizado (por exemplo, **`p95.45`**).
6. Para alterar o período, escolha a guia Métricas em gráfico. Escolha o cabeçalho da coluna ou um valor individual e, então, escolha um valor diferente.

Para obter a utilização média da CPU entre suas instâncias do EC2 usando a AWS CLI

Use o comando get-metric-statistics da seguinte forma:

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --statistics "Average" "SampleCount" \
--start-time 2016-10-11T23:18:00 --end-time 2016-10-12T23:18:00 --period 3600
```

A seguir está um exemplo de saída:

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2016-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```


Agregar estatísticas por grupo de Auto Scaling

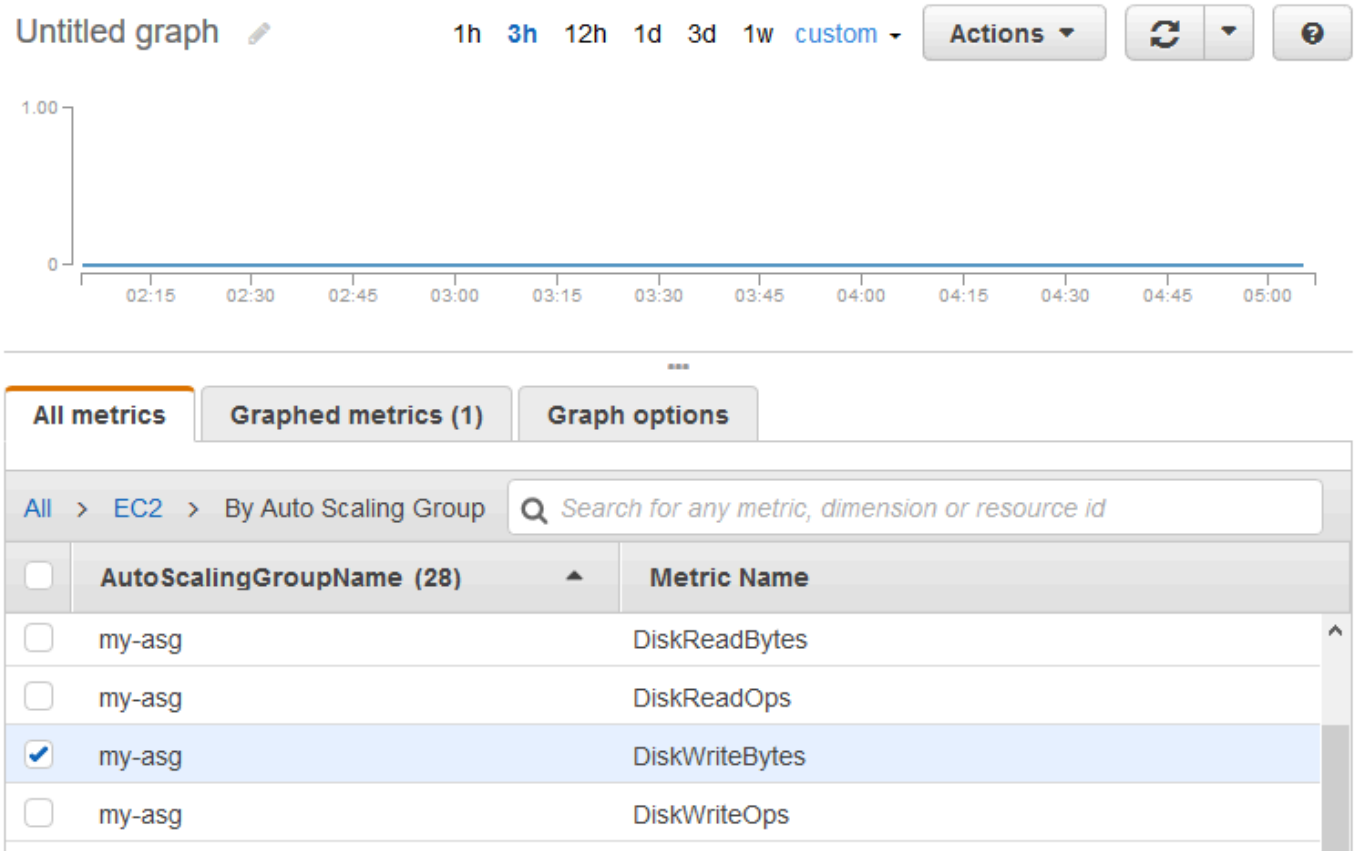
É possível agregar estatísticas para as instâncias do EC2 em um grupo do Auto Scaling. As métricas são completamente separadas entre regiões, mas é possível usar a matemática métricas do CloudWatch para agregar e transformar métricas de várias regiões. Também é possível usar o painel de contas cruzadas para executar matemática métricas em métricas de contas diferentes.

Este exemplo mostra como obter o total de bytes gravados em disco para um grupo do Auto Scaling. O total é calculado para períodos de 1 minuto para um intervalo de 24 horas em todas as instâncias do EC2 no grupo do Auto Scaling especificado.

Para exibir DiskWriteBytes para as instâncias em um grupo do Auto Scaling usando o console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Metrics (Métricas).
3. Escolha o namespace EC2 e escolha By Auto Scaling Group (Por grupo de Auto Scaling).

4. Selecione a linha da métrica DiskWriteBytes e o grupo do Auto Scaling específico, que exibe um gráfico da métrica para as instâncias no grupo do Auto Scaling. Para alterar o nome do gráfico, escolha o ícone de lápis. Para alterar o período, selecione um dos valores predefinidos ou escolha custom (personalizado).



5. Para alterar a estatística, escolha a guia Métricas em gráfico. Escolha o cabeçalho da coluna ou um valor individual e escolha uma das estatísticas ou percentis predefinidos ou especifique um percentil personalizado (por exemplo, **`p95.45`**).
6. Para alterar o período, escolha a guia Métricas em gráfico. Escolha o cabeçalho da coluna ou um valor individual e, então, escolha um valor diferente.

Para obter DiskWriteBytes para as instâncias em um grupo do Auto Scaling usando a AWS CLI

Use o comando get-metric-statistics da seguinte forma.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes
--dimensions Name=AutoScalingGroupName,Value=my-asg --statistics "Sum" "SampleCount" \
--start-time 2016-10-16T23:18:00 --end-time 2016-10-18T23:18:00 --period 360
```

A seguir, um exemplo de saída.

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2016-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2016-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

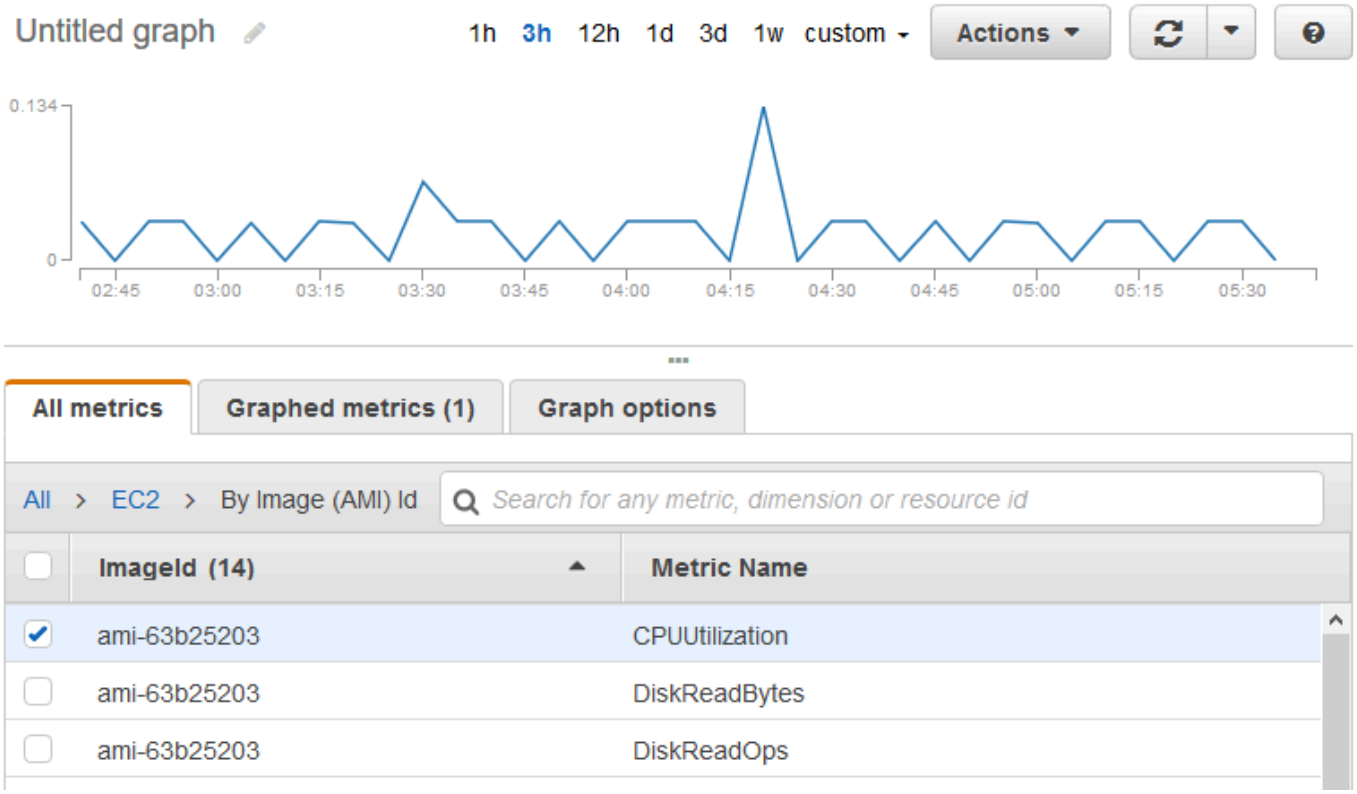

Agregar estatísticas por imagem de máquina da Amazon (AMI)

Você pode agregar estatísticas para as instâncias do EC2 com monitoramento detalhado habilitado. As instâncias que usam o monitoramento básico não são incluídas. Para obter mais informações, consulte Habilitar ou desabilitar o monitoramento detalhado de instâncias no Manual do usuário do Amazon EC2 para instâncias do Linux.

Este exemplo mostra como determinar a utilização média da CPU para todas as instâncias que usam a AMI especificada. A média é intervalos de mais de 60 segundos para um período de um dia.

Para exibir a utilização média da CPU pelo AMI usando o console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Metrics (Métricas).
3. Escolha o namespace EC2 e escolha By Image (AMI) Id (Por ID de imagem (AMI)).
4. Selecione a linha da métrica `CPUUtilization` e a AMI específica, que exibe um gráfico da métrica da AMI especificada. Para alterar o nome do gráfico, escolha o ícone de lápis. Para alterar o período, selecione um dos valores predefinidos ou escolha custom (personalizado).



5. Para alterar a estatística, escolha a guia Métricas em gráfico. Escolha o cabeçalho da coluna ou um valor individual e escolha uma das estatísticas ou percentis predefinidos ou especifique um percentil personalizado (por exemplo, **`p95.45`**).
6. Para alterar o período, escolha a guia Métricas em gráfico. Escolha o cabeçalho da coluna ou um valor individual e, então, escolha um valor diferente.

Para obter a utilização média da CPU por AMI usando a AWS CLI

Use o comando get-metric-statistics da seguinte forma.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=ImageId,Value=ami-3c47a355 --statistics Average \
--start-time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00 --period 3600
```

A operação retorna estatísticas que são valores de uma hora para o intervalo de um dia. Cada valor representa uma porcentagem de utilização média da CPU para instâncias do EC2 que executam a AMI especificada. A seguir, um exemplo de saída.

```
{
    "Datapoints": [
```

```
        {
            "Timestamp": "2016-10-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Publicar métricas personalizadas do

Você também pode publicar suas próprias métricas no CloudWatch usando a AWS CLI ou uma API. Você pode visualizar gráficas de estatísticas de suas métricas publicadas com o AWS Management Console.

O CloudWatch armazena dados sobre uma métrica como uma série de pontos de dados. Cada ponto de dados tem um time stamp associado. Você pode até mesmo publicar um conjunto agregado de pontos de dados chamado conjunto de estatísticas.

Tópicos

Métricas de alta resolução

Cada métrica é um dos seguintes:

- Resolução padrão, com dados de granularidade de um minuto
- Resolução alta, com dados de granularidade de um segundo

Por padrão, as métricas produzidas por serviços da AWS têm resolução padrão. Quando você publica uma métrica personalizada, pode defini-la com resolução padrão ou alta. Quando você publica uma métrica de alta resolução, o CloudWatch a armazena com uma resolução de 1 segundo. Você pode ler e recuperar essa métrica no período de 1 segundo, 5 segundos, 10 segundos, 30 segundos ou em qualquer múltiplo de 60 segundos.

As métricas de alta resolução podem também dar a você insight mais imediato da atividade de subminuto da seu aplicativo. Lembre-se de que cada chamada `PutMetricData` de uma métrica personalizada é cobrada. Portanto, chamar `PutMetricData` com mais frequência em uma métrica de alta resolução pode resultar em tarifas mais altas. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

Se você definir um alarme em uma métrica de alta resolução, pode especificar um alarme de alta resolução com um período de 10 ou 30 segundos ou pode definir um alarme regular com um período de qualquer múltiplo de 60 segundos. Há uma tarifa maior para alarmes de alta resolução com um período de 10 ou 30 segundos.

Usar dimensões

Em métricas personalizadas, o parâmetro `--dimensions` é comum. Uma dimensão esclarece com mais detalhes qual é a métrica e quais dados ela armazena. Você pode ter até 30 dimensões atribuídas a uma métrica. Cada dimensão é definida por um par de nome e valor.

A forma como você especifica uma dimensão é diferente quando você usa comandos diferentes. Com put-metric-data, você especifica cada dimensão como *`MyName`*=*`MyValue`*, e com get-metric-statistics ou put-metric-alarm você usa o formato `Name`=*`MyName`*, `Value`=*`MyValue`*. Por exemplo, o seguinte comando publica uma métrica `Buffers` com duas dimensões denominadas `InstanceId` e `InstanceType`.

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace MyNameSpace --unit
 Bytes --value 231434333 --dimensions InstanceId=1-23456789,InstanceType=m1.small
```

Este comando recupera as estatísticas para esta mesma métrica. Separe as partes de nome e valor de uma única dimensão com vírgulas, mas, se tiver várias dimensões, use um espaço entre uma dimensão e a próxima.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --namespace MyNameSpace --
dimensions Name=InstanceId,Value=1-23456789 Name=InstanceType,Value=m1.small --start-
time 2016-10-15T04:00:00Z --end-time 2016-10-19T07:00:00Z --statistics Average --period
 60
```

Se uma única métrica incluir várias dimensões, você deverá especificar um valor para cada dimensão definida ao usar get-metric-statistics. Por exemplo, a métrica `BucketSizeBytes` do Amazon S3 inclui as dimensões `BucketName` e `StorageType`. Portanto, você deve especificar as duas dimensões com get-metric-statistics.

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --start-time
 2017-01-23T14:23:00Z --end-time 2017-01-26T19:30:00Z --period 3600 --namespace
 AWS/S3 --statistics Maximum --dimensions Name=BucketName,Value=MyBucketName
 Name=StorageType,Value=StandardStorage --output table
```

Para ver quais dimensões estão definidas para uma métrica, use o comando list-metrics.

Publicar pontos de dados únicos

Para publicar um único ponto de dados para uma métrica nova ou existente, use o comando put-metric-data com um valor e um time stamp. Por exemplo, cada uma das seguintes ações publica um ponto de dados.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 2 --timestamp 2016-10-20T12:00:00.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 4 --timestamp 2016-10-20T12:00:01.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 5 --timestamp 2016-10-20T12:00:02.000Z
```

Se você chamar esse comando com um novo nome da métrica, o CloudWatch criará uma métrica para você. Caso contrário, o CloudWatch associará os dados com a métrica existente que você especificou.

> **Note**
>
> Ao criar uma métrica, pode levar até dois minutos antes que seja possível recuperar as estatísticas para a nova métrica usando o comando get-metric-statistics. No entanto, pode levar até 15 minutos até que a nova métrica apareça na lista de métricas recuperadas com o comando list-metrics.

Embora você possa publicar pontos de dados com carimbos de data/hora com precisão de milésimo de segundo, o CloudWatch agrega os dados com a precisão mínima de um segundo. O CloudWatch registra a média (soma de todos os itens dividida pelo número de itens) dos valores recebidos para cada período, bem como o número de amostras, o valor máximo e o valor mínimo para o mesmo período. Por exemplo, a métrica `PageViewCount` dos exemplos anteriores contém três pontos de dados com time stamps com apenas alguns segundos de intervalo. Se você tiver o período definido como 1 minuto, o CloudWatch agregará os três pontos de dados, pois eles têm carimbos de data/hora em um período de 1 minuto.

Você pode usar o comando get-metric-statistics para recuperar estatísticas com base nos pontos de dados publicados.

```
aws cloudwatch get-metric-statistics --namespace MyService --metric-name PageViewCount
 \
--statistics "Sum" "Maximum" "Minimum" "Average" "SampleCount" \
--start-time 2016-10-20T12:00:00.000Z --end-time 2016-10-20T12:05:00.000Z --period 60
```

A seguir, um exemplo de saída.

```
{
    "Datapoints": [
        {
            "SampleCount": 3.0,
            "Timestamp": "2016-10-20T12:00:00Z",
            "Average": 3.6666666666666665,
            "Maximum": 5.0,
            "Minimum": 2.0,
            "Sum": 11.0,
            "Unit": "None"
        }
    ],
    "Label": "PageViewCount"
```

```
}
```

Publicar conjuntos de estatísticas

Você pode agregar seus dados antes de publicá-los no CloudWatch. Quando você tem vários pontos de dados por minuto, a agregação de dados reduz o número de chamadas para put-metric-data. Por exemplo, em vez de chamar put-metric-data várias vezes para três pontos de dados com três segundos de diferença um do outro, é possível agregar os dados em um conjunto de estatísticas publicado com uma chamada, usando o parâmetro `--statistic-values`.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService
 --statistic-values Sum=11,Minimum=2,Maximum=5,SampleCount=3 --
timestamp 2016-10-14T12:00:00.000Z
```

O CloudWatch precisa dos pontos e dados brutos para calcular percentis. Se você publicar dados usando um conjunto de estatísticas, não poderá recuperar estatísticas de percentis para esses dados, a menos que uma das seguintes condições seja verdadeira:

- O `SampleCount` do conjunto de estatísticas é 1.
- O `Minimum` e o `Maximum` do conjunto de estatísticas são iguais

Publicar o valor zero

Quando os dados são mais esporádicos e você tem períodos sem dados associados, você pode optar por publicar o valor zero (`0`) para esse período ou nenhum valor. Se você usar chamadas periódicas para `PutMetricData` a fim de monitorar a integridade do seu aplicativo, talvez queira publicar zero em vez de nenhum valor. Por exemplo, é possível definir um alarme do CloudWatch para notificar você, se a aplicação não publicar métricas a cada cinco minutos. É recomendável que esse aplicativo publique zeros para períodos sem dados associados.

Você também pode publicar zeros se desejar rastrear o número total de pontos de dados ou se quiser que estatísticas como mínima e média incluam pontos de dados com o valor 0.

Parar de publicar métricas

Para parar de publicar métricas personalizadas no CloudWatch, altere o código da aplicação ou do serviço de forma a parar de usar PutMetricData. O CloudWatch não extrai métricas de aplicações.

Ele apenas recebe o que é enviado para ele. Portanto, para parar de publicar suas métricas, é necessário interrompê-las na fonte.

Usar alarmes do Amazon CloudWatch

É possível criar alarmes de métrica e compostos no Amazon CloudWatch.

- Um alarme de métrica observa uma única métrica do CloudWatch ou o resultado de uma expressão matemática baseada em métricas do CloudWatch. O alarme executa uma ou mais ações com base no valor da métrica ou na expressão em relação a um limite em alguns períodos. A ação pode ser enviar uma notificação a um tópico do Amazon SNS, executar uma ação do Amazon EC2 ou uma ação do Amazon EC2 Auto Scaling ou criar um OpsItem ou incidente no Systems Manager.
- Um alarme composto inclui uma expressão de regra que leva em conta os estados de outros alarmes que você criou. O alarme composto entrará no estado ALARM somente se todas as condições da regra forem atendidas. Os alarmes especificados na expressão de regra de um alarme composto podem incluir alarmes de métrica e outros alarmes compostos.

  O uso de alarmes compostos pode reduzir o ruído do alarme. Você pode criar vários alarmes de métrica e também criar um alarme composto e configurar alertas apenas para o alarme composto. Por exemplo, um alarme composto poderá entrar no estado ALARM somente quando todos os alarmes de métrica subjacentes estiverem no estado ALARM.

  Os alarmes compostos podem enviar notificações do Amazon SNS quando mudam de estado e podem criar OpsItems ou incidentes do Systems Manager quando entram no estado ALARM, mas não podem executar ações do EC2 ou ações do Auto Scaling.

> **Note**
>
> Você pode criar quantos alarmes quiser em sua conta da AWS.

É possível adicionar alarmes aos painéis, para monitorar e receber alertas sobre seus recursos da AWS e aplicações em várias regiões. Após ser adicionado a um painel, o alarme ficará cinza quando estiver no estado `INSUFFICIENT_DATA` e vermelho quando estiver no estado `ALARM`. O alarme é mostrado sem cor quando está no estado `OK`.

Também é possível adicionar como favoritos alarmes recém-visitados via opção Favorites and recents (Favoritos e recentes) no painel de navegação do console do CloudWatch. A opção Favorites

and recents (Favoritos e recentes) contém colunas para seus alarmes favoritos e alarmes visitados recentemente.

Um alarme invoca ações somente quando muda de estado. A exceção se aplica a alarmes com ações do Auto Scaling. Para ações do Auto Scaling, o alarme continuará invocando a ação para cada período que ele permanecer no novo estado.

Um alarme pode observar uma métrica da mesma conta. Se você habilitou a funcionalidade entre contas no console do CloudWatch, também poderá criar alarmes que observem métricas em outras contas da AWS. Não há suporte para a criação de alarmes compostos entre contas. A criação de alarmes entre contas que usam expressões matemáticas é compatível, exceto as funções `ANOMALY_DETECTION_BAND`, `INSIGHT_RULE` e `SERVICE_QUOTA` que não são compatíveis com alarmes entre contas.

> **Note**
>
> O CloudWatch não testa nem valida as ações especificadas nem detecta erros do Amazon EC2 Auto Scaling ou do Amazon SNS resultantes de uma tentativa de invocar ações não existentes. Verifique se as ações de alarme existem.

Estados de alarme de métrica

Um alarme de métrica tem estes estados possíveis:

- `OK`: a métrica ou a expressão está dentro do limite definido.
- `ALARM`: a métrica ou a expressão está fora do limite definido.
- `INSUFFICIENT_DATA`: o alarme acabou de ser acionado, a métrica não está disponível ou não há dados suficientes para a métrica determinar o estado do alarme.

Avaliar um alarme

Ao criar um alarme, você especifica três configurações para habilitar o CloudWatch e avaliar quando alterar o estado do alarme:

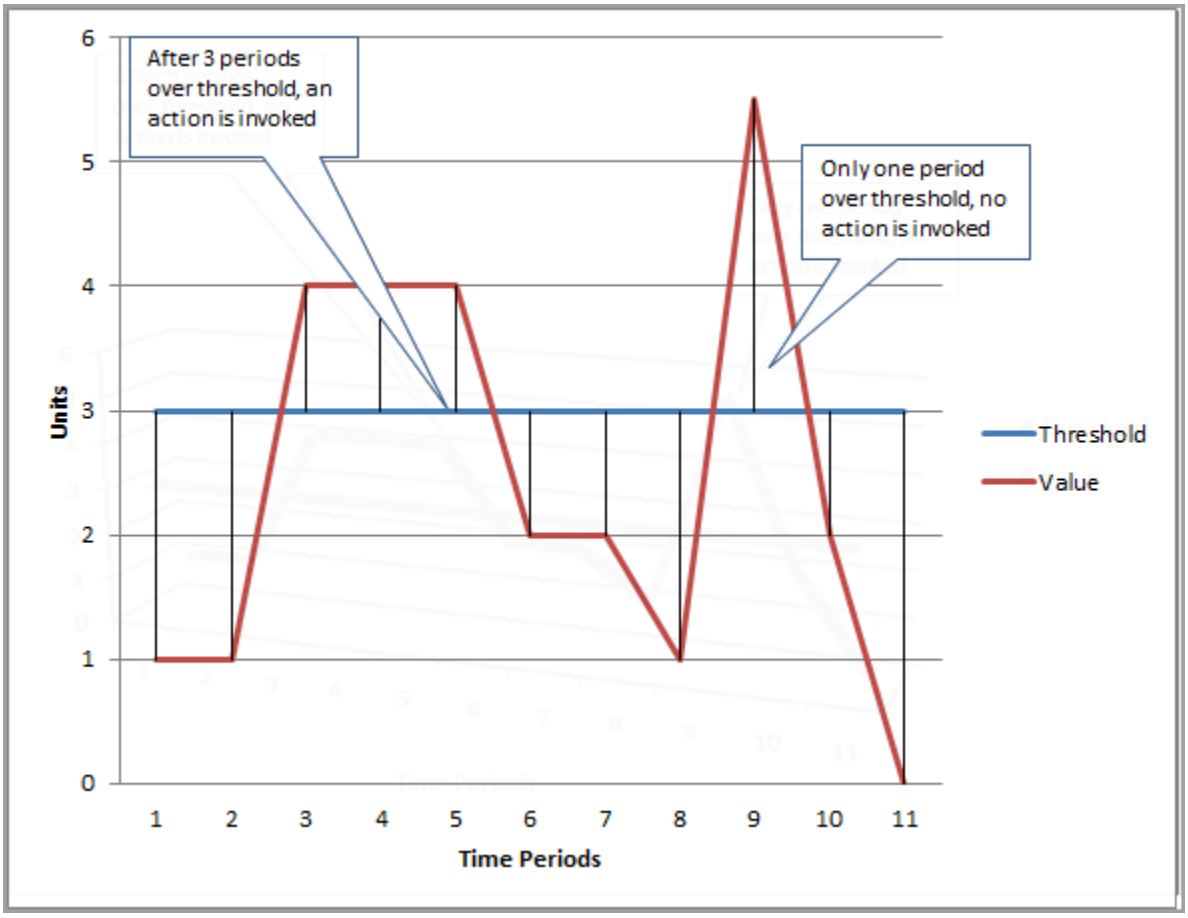
- Período é o intervalo de tempo para avaliar a métrica ou a expressão e criar cada ponto de dados de um alarme. Ele é expresso em segundos.

- Evaluation Periods (Períodos de avaliação) é o número de períodos mais recentes, ou pontos de dados, para avaliar quando determinar o estado do alarme.
- Datapoints to Alarm (Pontos de dados para alarme) é o número de pontos de dados dentro dos períodos de avaliação que devem estar violando para fazer com que o alarme passe para o estado `ALARM`. Os pontos de dados de violação não precisam ser consecutivos, mas eles devem estar dentro do último número de pontos de dados igual ao Evaluation Period (Período de avaliação).

Para qualquer período de um minuto ou mais, um alarme é avaliado a cada minuto, e a avaliação é baseada na janela de tempo definida pelo Período epelos Períodos de Avaliação. Por exemplo, se o Período for de 5 minutos (300 segundos) e os Períodos de Avaliação forem de 1, no final do minuto 5 o alarme será avaliado com base nos dados dos minutos 1 a 5. Então, no final do minuto 6, o alarme será avaliado com base nos dados dos minutos 2 a 6.

Se o período do alarme for de 10 segundos ou de 30 segundos, o alarme será avaliado a cada 10 segundos.

Na figura a seguir, o limite para um alarme de métrica é definido como três unidades. Tanto o Evaluation Period (Período de avaliação) como os Datapoints to Alarm (Pontos de Dados para Alarme) são 3. Ou seja, quando todos os pontos de dados nos três períodos consecutivos mais recentes estiverem acima do limite, o alarme passará para o estado `ALARM`. Na figura, isso acontece do terceiro ao quinto períodos de tempo. No período seis, o valor fica abaixo do limite. Portanto, um dos períodos que estão sendo avaliados não é violado, e o estado do alarme volta para `OK`. Durante o nono período, o limite é violado novamente, mas somente em um período. Consequentemente, o estado do alarme permanece `OK`.



Ao configurar Evaluation Periods (Períodos de avaliação) e Datapoints to Alarm (Pontos de dados para alarme) como valores diferentes, você está configurando um alarme “M de N”. Datapoints to Alarm (Pontos de dados para alarme) é (“M”) e Evaluation Periods (Períodos de avaliação) é (“N”). O intervalo de avaliação é o número de períodos de avaliação multiplicado pela duração do período. Por exemplo, se você configurar 4 de 5 pontos de dados com um período de 1 minuto, o intervalo de avaliação será de 5 minutos. Se você configurar 3 de 3 pontos de dados com um período de 10 minutos, o intervalo de avaliação será de 30 minutos.

> **Note**
>
> Se os pontos de dados estiverem ausentes logo depois que você criar um alarme, e se a métrica estava sendo relatada para o CloudWatch antes da criação do alarme, ao avaliá-lo, o CloudWatch recuperará os pontos de dados mais recentes, de antes de o alarme ter sido criado.

Ações de alarme

É possível especificar quais ações um alarme realizará ao mudar de estado entre os estados OK, ALARM e INSUFFICIENT_DATA.

A maioria das ações pode ser definida para a transição para cada um dos três estados. Com exceção das ações do Auto Scaling, as ações acontecem somente em transições de estado e não serão executadas novamente se a condição persistir por horas ou dias. É possível usar o fato de que várias ações são permitidas para que um alarme envie um email quando um limite for violado e, em seguida, outro quando a condição de violação terminar. Isso o ajudará a verificar se suas ações de escalonamento ou recuperação são acionadas quando esperado e estão funcionando conforme desejado.

As ações apresentadas a seguir são compatíveis como ações de alarme.

- Notificar um ou mais assinantes ao usar um tópico do Amazon Simple Notification Service. Os assinantes podem ser aplicações e também pessoas. Para obter mais informações sobre o Amazon SNS, consulte O que é o Amazon SNS?.
- Invocar uma função do Lambda. Essa é a maneira mais fácil de automatizar ações personalizadas em alterações de estado de alarme.
- Os alarmes baseados em métricas do EC2 também podem executar ações do EC2, como interromper, encerrar, reinicializar ou recuperar uma instância do EC2. Para ter mais informações, consulte Criar alarmes para interromper, terminar, reinicializar ou recuperar uma instância do EC2.
- Os alarmes podem executar ações para escalar um grupo do Auto Scaling. Para obter mais informações, consultePolíticas de escalabilidade simples e em etapas do Amazon EC2 Auto Scaling.
- Os alarmes podem criar OpsItems no OpsCenter do Systems Manager ou criar incidentes no AWS Systems Manager Incident Manager. Essas ações são executadas apenas quando o alarme entra no estado ALARM (ALARME). Para obter mais informações, consulte Configurar o CloudWatch para criar OpsItems de alarmes e Criação de incidentes.

Ações de alarme para o Lambda

Os alarmes do CloudWatch garantem uma invocação assíncrona da função do Lambda para uma determinada alteração de estado, exceto nos seguintes casos:

- Quando a função não existe.

- Quando o CloudWatch não está autorizado a invocar a função do Lambda.

Se o CloudWatch não conseguir acessar o serviço do Lambda ou se a mensagem for rejeitada por outro motivo, o CloudWatch tentará novamente até que a invocação seja bem-sucedida. O Lambda enfileira a mensagem e processa novas tentativas de execução. Para obter mais informações sobre esse modelo de execução, incluindo informações sobre como o Lambda lida com erros, consulte Invocação assíncrona no Guia do desenvolvedor do AWS Lambda.

Você pode invocar uma função do Lambda na mesma conta ou em outras contas da AWS.

Ao especificar um alarme para invocar uma função do Lambda como uma ação de alarme, é possível optar por especificar o nome da função, o alias da função ou uma versão específica de uma função.

Ao especificar uma função do Lambda como uma ação de alarme, você deve criar uma política de recursos para a função com a finalidade de permitir que a entidade principal de serviço do CloudWatch invoque a função.

Uma maneira de fazer isso é ao usar a AWS CLI, como no seguinte exemplo:

```
aws lambda add-permission \
--function-name my-function-name \
--statement-id AlarmAction \
--action 'lambda:InvokeFunction' \
--principal lambda.alarms.cloudwatch.amazonaws.com \
--source-account 111122223333 \
--source-arn arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name
```

Como alternativa, é possível criar uma política semelhante a um dos exemplos apresentados a seguir e, em seguida, atribuí-la à função.

O exemplo apresentado a seguir especifica a conta na qual o alarme está localizado, portanto, somente os alarmes dessa conta (111122223333) podem invocar a função.

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [{
        "Sid": "AlarmAction",
        "Effect": "Allow",
        "Principal": {
            "Service": "lambda.alarms.cloudwatch.amazonaws.com"
```

```
        },
        "Action": "lambda:InvokeFunction",
        "Resource": "arn:aws:lambda:us-east-1:444455556666:function:function-name",
        "Condition": {
            "StringEquals": {
                "AWS:SourceAccount": "111122223333"
            }
        }
    }]
}
```

O exemplo apresentado a seguir tem um escopo mais restrito, permitindo que somente o alarme especificado na conta indicada invoque a função.

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "AlarmAction",
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.alarms.cloudwatch.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:444455556666:function:function-name",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "111122223333",
          "AWS:SourceArn": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
        }
      }
    }]
}
```

Não recomendamos a criação de uma política que não especifique uma conta de origem, porque essas políticas são vulneráveis a problemas de “confused deputy”.

Objeto de evento enviado do CloudWatch para o Lambda

Ao configurar uma função do Lambda como uma ação de alarme, o CloudWatch entrega uma carga JSON à função do Lambda quando invoca a função. Essa carga JSON serve como o objeto de

evento para a função. É possível extrair dados desse objeto em JSON e usá-los em sua função. Veja a seguir um exemplo de um objeto de evento de um alarme de métrica.

```
{
  'source': 'aws.cloudwatch',
  'alarmArn': 'arn:aws:cloudwatch:us-east-1:444455556666:alarm:lambda-demo-metric-
alarm',
  'accountId': '444455556666',
  'time': '2023-08-04T12:36:15.490+0000',
  'region': 'us-east-1',
  'alarmData': {
    'alarmName': 'lambda-demo-metric-alarm',
    'state': {
      'value': 'ALARM',
      'reason': 'test',
      'timestamp': '2023-08-04T12:36:15.490+0000'
    },
    'previousState': {
      'value': 'INSUFFICIENT_DATA',
      'reason': 'Insufficient Data: 5 datapoints were unknown.',
      'reasonData':
 '{"version":"1.0","queryDate":"2023-08-04T12:31:29.591+0000","statistic":"Average","period":60
[],"threshold":5.0,"evaluatedDatapoints":[{"timestamp":"2023-08-04T12:30:00.000+0000"},
{"timestamp":"2023-08-04T12:29:00.000+0000"},
{"timestamp":"2023-08-04T12:28:00.000+0000"},
{"timestamp":"2023-08-04T12:27:00.000+0000"},
{"timestamp":"2023-08-04T12:26:00.000+0000"}]}',
      'timestamp': '2023-08-04T12:31:29.595+0000'
    },
    'configuration': {
      'description': 'Metric Alarm to test Lambda actions',
      'metrics': [
        {
          'id': '1234e046-06f0-a3da-9534-EXAMPLEe4c',
          'metricStat': {
            'metric': {
              'namespace': 'AWS/Logs',
              'name': 'CallCount',
              'dimensions': {
                'InstanceId': 'i-12345678'
              }
            },
            'period': 60,
```

```
            'stat': 'Average',
            'unit': 'Percent'
          },
          'returnData': True
        }
      ]
    }
  }
}
```

Veja a seguir um exemplo de um objeto de evento de um alarme composto.

```
{
  'source': 'aws.cloudwatch',
  'alarmArn': 'arn:aws:cloudwatch:us-east-1:111122223333:alarm:SuppressionDemo.Main',
  'accountId': '111122223333',
  'time': '2023-08-04T12:56:46.138+0000',
  'region': 'us-east-1',
  'alarmData': {
    'alarmName': 'CompositeDemo.Main',
    'state': {
      'value': 'ALARM',
      'reason': 'arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild transitioned to ALARM at Friday 04
 August, 2023 12:54:46 UTC',
      'reasonData': '{"triggeringAlarms":[{"arn":"arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild","state":
{"value":"ALARM","timestamp":"2023-08-04T12:54:46.138+0000"}}]}',
      'timestamp': '2023-08-04T12:56:46.138+0000'
    },
    'previousState': {
      'value': 'ALARM',
      'reason': 'arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild transitioned to ALARM at Friday 04
 August, 2023 12:54:46 UTC',
      'reasonData': '{"triggeringAlarms":[{"arn":"arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild","state":
{"value":"ALARM","timestamp":"2023-08-04T12:54:46.138+0000"}}]}',
      'timestamp': '2023-08-04T12:54:46.138+0000',
      'actionsSuppressedBy': 'WaitPeriod',
      'actionsSuppressedReason': 'Actions suppressed by WaitPeriod'
    },
    'configuration': {
```

```
      'alarmRule': 'ALARM(CompositeDemo.FirstChild) OR
 ALARM(CompositeDemo.SecondChild)',
      'actionsSuppressor': 'CompositeDemo.ActionsSuppressor',
      'actionsSuppressorWaitPeriod': 120,
      'actionsSuppressorExtensionPeriod': 180
    }
  }
}
```

Configurar como os alarmes do CloudWatch tratam dados ausentes

Às vezes, nem todos os pontos de dados esperados para uma métrica são relatados ao CloudWatch. Por exemplo, isso pode ocorrer quando uma conexão é perdida, um servidor é desativado, ou quando uma métrica informa apenas dados de forma intermitente por padrão.

O CloudWatch permite que você especifique como tratar pontos de dados ausentes ao avaliar um alarme. Isso pode ajudar a configurar seu alarme para passar ao estado `ALARM` quando for apropriado para o tipo de dados que está sendo monitorado. Você pode evitar falsos positivos quando dados ausentes não indicam um problema.

Assim como cada alarme está sempre em um dos três estados, cada ponto de dados específico relatado do CloudWatch se insere em uma destas três categorias:

- Não violar (dentro do limite)
- Violar (violando o limite)
- Missing (Ausente)

Para cada alarme, é possível especificar o CloudWatch para tratar pontos de dados ausentes como qualquer uma destas opções:

- `notBreaching`: os pontos de dados ausentes são tratados como "bons" e dentro do limite
- `breaching`: os pontos de dados ausentes são tratados como “ruins” e violando o limite
- `ignore`: o estado do alarme atual é mantido
- `missing`: se todos os pontos de dados no intervalo de avaliação do alarme estiverem ausentes, o alarme passará para INSUFFICIENT_DATA.

A melhor opção depende do tipo de métrica e da finalidade do alarme. Por exemplo, se você estiver criando um alarme de reversão de aplicações usando uma métrica que relata dados continuamente, pode desejar tratar os pontos de dados ausentes como uma violação, pois isso pode indicar que há algo de errado. No entanto, para uma métrica que gera pontos de dados somente quando ocorre um erro, como `ThrottledRequests` no Amazon DynamoDB, é possível tratar dados ausentes como `notBreaching`. O comportamento padrão é `missing`.

> **Important**
>
> Os alarmes configurados nas métricas do Amazon EC2 podem entrar temporariamente no estado INSUFFICIENT_DATA se houver pontos de dados de métricas ausentes. Isso é raro, mas pode acontecer quando o relatório de métricas é interrompido, mesmo quando a instância do Amazon EC2 está íntegra. Para os alarmes nas métricas do Amazon EC2 que estão configurados para executar ações de interrupção, encerramento, reinicialização ou recuperação, recomendamos configurar esses alarmes para tratar os dados ausentes como `missing` e para que esses alarmes sejam acionados somente quando estiverem no estado ALARM.

Escolher a melhor opção para seu alarme evita alterações desnecessárias e enganosas e também indica com mais precisão a integridade do seu sistema.

> **Important**
>
> Alarmes que avaliam métricas no namespace `AWS/DynamoDB` sempre ignoram dados ausentes, mesmo que você escolha uma opção diferente para como o alarme deve tratar dados ausentes. Quando uma métrica `AWS/DynamoDB` tem dados ausentes, os alarmes que avaliam essa métrica permanecem no estado em que estiverem na ocasião.

Como o estado do alarme é avaliado quando há dados ausentes

Sempre que um alarme avalia se é necessário alterar o estado, o CloudWatch tenta recuperar um maior número de pontos de dados do que o número especificado em Evaluation Periods (Períodos de avaliação). O número exato de pontos de dados que ele tenta recuperar depende do tamanho do período de alarme e se ele é baseado em uma métrica com resolução padrão ou alta. O período de pontos de dados que ele tenta recuperar é o intervalo de avaliação.

Assim que o CloudWatch recupera esses pontos de dados, ocorre o seguinte:

- Se não houver pontos de dados ausentes no intervalo de avaliação, o CloudWatch avaliará o alarme com base nos pontos de dados mais recentes coletados. O número de pontos de dados avaliados é igual ao valor de Evaluation Periods (Períodos de avaliação) do alarme. Os pontos de dados excedentes mais antigos no intervalo de avaliação não são necessários e são ignorados.
- Se alguns pontos de dados no intervalo de avaliação estiverem ausentes, mas o total de pontos de dados recuperados corretamente for igual ou maior que os Evaluation Periods (Períodos de avaliação) do alarme, o CloudWatch avaliará o estado do alarme com base nos pontos de dados reais mais recentes que foram recuperados corretamente, inclusive os pontos de dados excedentes necessários mais antigos no período de avaliação. Nesse caso, o valor que você define para como tratar dados ausentes não é necessário e é ignorado.
- Se alguns pontos de dados no intervalo de avaliação estiverem ausentes e o número de pontos de dados reais que foram recuperados for menor do que o número de Evaluation Periods (Períodos de avaliação) do alarme, o CloudWatch preencherá os pontos de dados ausentes com o resultado especificado sobre como tratar dados ausentes e avaliará o alarme. Contudo, todos os pontos de dados reais no intervalo de avaliação serão incluídos na avaliação. O CloudWatch usa pontos de dados ausentes o mínimo possível.

> **Note**
>
> Um caso específico desse comportamento é que os alarmes do CloudWatch podem reavaliar repetidamente o último conjunto de pontos de dados por um período depois que o fluxo da métrica é interrompido. Essa reavaliação pode fazer com que o estado do alarme mude e as ações sejam executadas novamente, se ele tiver sido alterado imediatamente antes do fluxo de métrica ser interrompido. Para atenuar esse comportamento, use períodos mais curtos.

As tabelas a seguir ilustram exemplos do comportamento de avaliação de alarme. Na primeira tabela, Datapoints to Alarm (Pontos de dados para alarme) e Evaluation Periods (Períodos de avaliação) são 3. O CloudWatch recupera os cinco pontos de dados mais recentes ao avaliar o alarme, caso algum dos três pontos de dados mais recentes esteja ausente. O intervalo de avaliação do alarme é 5.

A coluna 1 exibe os cinco pontos de dados mais recentes, pois o intervalo de avaliação é 5. Esses pontos de dados são exibidos com o ponto de dados mais recente à direita, em que 0 é um ponto de dados de não violação, X é um ponto de dados violação e - é um ponto de dados ausente.

A coluna 2 mostra quantos dos 3 pontos de dados necessários estão ausentes. Embora os 5 pontos de dados mais recentes sejam avaliados, apenas 3 (a configuração para Evaluation Periods (Períodos de avaliação)) são necessárias para avaliar o estado do alarme. O número de pontos de dados na coluna 2 é o número de pontos de dados que devem ser "preenchidos", usando a configuração de como dados ausentes estão sendo tratados.

Nas colunas de 3 a 6, os cabeçalhos de coluna são os valores possíveis para tratar dados ausentes. As linhas dessas colunas mostram o estado do alarme definido para cada uma dessas possíveis formas de tratar dados ausentes.

Pontos de dados	Número de pontos de dados que deverão ser preenchidos	MISSING (AUSENTE)	IGNORE	VIOLAÇÃO	NÃO VIOLAÇÃO
0 - X - X	0	OK	OK	OK	OK
- - - - 0	2	OK	OK	OK	OK
- - - - -	3	INSUFFICIENT_DATA	Reter estado atual	ALARM	OK
0 X X - X	0	ALARM	ALARM	ALARM	ALARM
- - X - -	2	ALARM	Reter estado atual	ALARM	OK

Na segunda linha da tabela anterior, o alarme permanece em OK mesmo que os dados ausentes sejam tratados como violação, porque um ponto de dados existente não está violando o limite, e isso é avaliado junto com dois pontos de dados ausentes que são tratados como violação. Na próxima vez em que esse alarme for avaliado, se os dados ainda estiverem ausentes, ele passará para ALARM, pois esse ponto de dados de não violação não estará mais no intervalo de avaliação.

A terceira linha, onde todos os cinco pontos de dados mais recentes estão ausentes, ilustra como as várias configurações para tratar dados ausentes afetam o estado do alarme. Se os pontos de dados ausentes forem considerados de violação, o alarme entrará no estado ALARM; caso forem considerados de não violação, o alarme entrará no estado OK. Se os pontos de dados ausentes

forem ignorados, o alarme reterá o estado atual que tinha antes dos pontos de dados ausentes. E se os pontos de dados ausentes são apenas considerados ausentes, então o alarme não tem dados reais recentes suficientes para fazer uma avaliação e passa para INSUFFICIENT_DATA.

Na quarta linha, o alarme passa para o estado ALARM em todos os casos porque os três pontos de dados mais recentes estão em violação, e tanto os Evaluation Periods (Períodos de avaliação) como os Datapoints to Alarm (Pontos de Dados para Alarme) do alarme são ambos definidos como 3. Nesse caso, o ponto de dados que falta é ignorado, e a configuração para como avaliar dados que estão faltando não é necessária, pois há 3 pontos de dados reais para avaliar.

A linha 5 representa um caso especial de avaliação de alarme chamado estado de alarme prematuro. Para ter mais informações, consulte Evitar transições prematuras para o estado do alarme.

Na próxima tabela, o Period (Período) é novamente definido como 5 minutos, e Datapoints to Alarm (Pontos de dados para alarme) é somente 2, enquanto Evaluation Periods (Períodos de avaliação) é 3. Esse é um alarme 2 de 3, M de N.

O intervalo de avaliação é 5. Esse é o número máximo de pontos de dados recentes que são recuperados e podem ser usados caso alguns pontos de dados estejam ausentes.

Pontos de dados	Número de pontos de dados ausentes	AUSENTE	IGNORE	VIOLAÇÃO	NÃO VIOLAÇÃO
0 - X - X	0	ALARM	ALARM	ALARM	ALARM
0 0 X 0 X	0	ALARM	ALARM	ALARM	ALARM
0 - X - -	1	OK	OK	ALARM	OK
- - - - 0	2	OK	OK	ALARM	OK
- - - - X	2	ALARM	Reter estado atual	ALARM	OK

Nas linhas 1 e 2, o alarme sempre passa para o estado ALARM porque dois dos três pontos de dados mais recentes estão em violação. Na linha 2, os dois pontos de dados mais antigos no

intervalo de avaliação não são necessários porque nenhum dos três pontos de dados mais recentes está ausente. Portanto, esses dois pontos de dados mais antigos são ignorados.

Nas linhas 3 e 4, o alarme só passará para o estado ALARM se os dados ausentes forem tratados como violação, e nesse caso os dois pontos de dados ausentes mais recentes serão tratados como violação. Na linha 4, esses dois pontos de dados ausentes que são tratados como violação fornecem os dois pontos de dados de violação necessários para acionar o estado ALARM.

A linha 5 representa um caso especial de avaliação de alarme chamado estado de alarme prematuro. Para obter mais informações, consulte a seção a seguir.

Evitar transições prematuras para o estado do alarme

A avaliação de alarmes do CloudWatch inclui lógica para tentar evitar alarmes falsos, nos quais o alarme entra no estado ALARM prematuramente quando os dados são intermitentes. O exemplo da linha 5 nas tabelas da seção anterior ilustra essa lógica. Nessas linhas e nos exemplos a seguir, os Evaluation Periods (Períodos de avaliação) são 3, e o intervalo de avaliação é de 5 pontos de dados. Os Datapoints to Alarm (Pontos de dados para alarme) são 3, exceto para o exemplo M de N, em que os Datapoints to Alarm são 2.

Suponha que os dados mais recentes de um alarme sejam - - - - X, com quatro pontos de dados ausentes e um ponto de dados de violação como ponto de dados mais recente. Como o próximo ponto de dados pode não ser violado, o alarme não entra imediatamente no estado ALARM quando os dados são - - - - X ou - - - X - e Datapoints to Alarm (Pontos de dados para alarme) são 3. Desta forma, evitam-se os falsos positivos quando o próximo ponto de dados for de não violação e faz com que os dados sejam - - - X 0 ou - - X - 0.

Porém, se os últimos pontos de dados forem - - X - -, o alarme entra no estado ALARM mesmo se os pontos de dados ausentes forem tratados como ausentes. Isso ocorre porque os alarmes são projetados para sempre entrar no estado ALARM quando o ponto de dados de violação mais antigo disponível durante o número de pontos de dados dos Evaluation Periods (Períodos de avaliação) é pelo menos tão antigo quanto o valor dos Datapoints to Alarm (Pontos de dados para alarme) e todos os outros pontos de dados mais recentes estão em violação ou ausentes. Neste caso, o alarme entra no estado ALARM mesmo que o número total de pontos de dados disponíveis seja inferior a M Datapoints to Alarm (Pontos de dados para alarme).

Essa lógica de alarme também se aplica a alarmes M de N. Se o ponto de dados em violação mais antigo durante o intervalo de avaliação for pelo menos tão antigo quanto o valor de Datapoints to Alarm (Pontos de dados para alarme), e todos os pontos de dados mais recentes estiverem em

violação ou ausentes, o alarme entrará no estado ALARM para qualquer valor de M Datapoints to Alarm (Pontos de dados para alarme).

Alarmes de alta resolução

Se você definir um alarme em uma métrica de alta resolução, pode especificar um alarme de alta resolução com um período de 10 ou 30 segundos ou pode definir um alarme regular com um período de qualquer múltiplo de 60 segundos. Há um custo maior para alarmes de alta resolução. Para obter mais informações sobre as métricas de alta resolução, consulte Publicar métricas personalizadas do
.

Alarmes em expressões matemáticas

Defina um alarme com base no resultado de uma expressão matemática baseada em uma ou mais métricas do CloudWatch. Uma expressão matemática usada para um alarme pode incluir até 10 métricas. Toda métrica deve estar usando o mesmo período.

Para um alarme baseado em uma expressão matemática, é possível especificar como você deseja que o CloudWatch trate pontos de dados ausentes. Nesse caso, o ponto de dados é considerado ausente se a expressão matemática não retornar um valor para esse ponto de dados.

Os alarmes baseados em expressões matemáticas não poderão realizar ações do Amazon EC2.

Para obter mais informações sobre expressões matemáticas e sintaxe de métrica, consulte Usar matemática de métricas.

Alarmes do CloudWatch baseados em percentual e exemplos de poucos dados

Quando você define um percentil como a estatística para um alarme, você pode especificar o que fazer quando não há dados suficientes para uma boa avaliação estatística. Você pode escolher que o alarme avalie a estatística de qualquer forma e possivelmente altere o estado do alarme. Ou você pode determinar que o alarme ignore a métrica enquanto o tamanho da amostra for baixo e esperar para avaliá-la até que haja dados suficientes para serem estatisticamente significativos.

Para percentis entre 0,5 (inclusive) e 1,00 (exclusive), essa configuração é usada quando há menos de 10/(1 percentil) pontos de dados durante o período de avaliação. Por exemplo, essa configuração seria usada se houvesse menos do que 1.000 amostras para um alarme em um p99 percentil. Para

percentis entre 0 e 0,5 (exclusive), a configuração é usada quando há menos de 10/percentil pontos de dados.

Recursos comuns dos alarmes do CloudWatch

Estes recursos se aplicam a todos os alarmes do CloudWatch:

- Não há limite para o número de alarmes que você pode criar. Para criar ou atualizar um alarme, use o console do CloudWatch, a ação PutMetricAlarm da API ou o comando put-metric-alarm na AWS CLI.
- Os nomes dos alarmes devem conter somente caracteres UTF-8 da e não podem conter caracteres de controle ASCII
- É possível listar um ou todos os alarmes configurados no momento e listar todos os alarmes em um determinado estado usando o console do CloudWatch, a ação DescribeAlarms da API ou o comando describe-alarms na AWS CLI.
- É possível desabilitar e habilitar alarmes usando as ações console do DisableAlarmActions e EnableAlarmActions da API ou os comandos disable-alarm-actions e enable-alarm-actions na AWS CLI.
- É possível testar um alarme configurando-o para qualquer estado usando a ação SetAlarmState da API ou o comando set-alarm-state na AWS CLI. Essa alteração de estado temporária dura somente até ocorrer a próxima comparação de alarmes.
- É possível criar um alarme para uma métrica personalizada antes de criar essa métrica personalizada. Para o alarme ser válido, é necessário incluir todas as dimensões para a métrica personalizada, além do namespace e do nome da métrica na definição do alarme. Para fazer isso, você pode usar a ação PutMetricAlarm da API ou o comando put-metric-alarm na AWS CLI.
- É possível exibir o histórico de um alarme usando o console do CloudWatch, a ação DescribeAlarmHistory da API ou o comando describe-alarm-history na AWS CLI. O CloudWatch preserva o histórico de alarmes por 30 dias. Cada transição de estado é marcada com um time stamp exclusivo. Em casos raros, o histórico pode mostrar mais de uma notificação para uma alteração de estado. O time stamp permite confirmar alterações de estado exclusivas.
- Você pode adicionar alarmes como favoritos na opção Favorites and recents (Favoritos e recentes) no painel de navegação do console do CloudWatch movendo o ponteiro do mouse sobre o alarme que deseja adicionar e escolhendo o símbolo de estrela próximo a ele.
- O número de períodos de avaliação para um alarme multiplicado pela duração de cada período de avaliação não pode exceder um dia.

> **Note**
>
> Alguns recursos da AWS não enviam dados de métrica para o CloudWatch em determinadas condições.
> Por exemplo, o Amazon EBS não pode enviar dados de métrica para um volume disponível que não esteja anexado a uma instância do Amazon EC2, porque não há atividade de métrica a ser monitorada para esse volume. Se você tiver um alarme definido para essa métrica, poderá visualizar a alteração do estado para `INSUFFICIENT_DATA`. Isso pode indicar que o recurso está inativo e não necessariamente indicar que há um problema.
> É possível especificar como cada alarme lida com os dados ausentes. Para ter mais informações, consulte Configurar como os alarmes do CloudWatch tratam dados ausentes.

Recomendações de alarmes de práticas recomendadas para serviços da AWS

O CloudWatch fornece recomendações de alarme prontas para uso. Esses são alarmes do CloudWatch que recomendamos que você crie para métricas publicadas por outros serviços da AWS. Essas recomendações podem ajudar a identificar as métricas para as quais você deve definir alarmes a fim de seguir as práticas recomendadas de monitoramento. As recomendações também sugerem os limites de alarme a serem definidos. Seguir essas recomendações pode contribuir para que você não perca um monitoramento importante da sua infraestrutura da AWS.

Para encontrar as recomendações de alarme, use a seção de métricas do console do CloudWatch e selecione a opção de filtro de recomendações de alarmes. Se você navegar até os alarmes recomendados no console e, em seguida, criar um alarme recomendado, o CloudWatch poderá preencher previamente algumas das configurações de alarme. Para alguns alarmes recomendados, o valor limite do alarme também é pré-preenchido. Também é possível usar o console para baixar definições de alarme de infraestrutura como código para alarmes recomendados e, em seguida, usar esse código para criar o alarme no AWS CloudFormation, na AWS CLI ou no Terraform.

Além disso, é possível visualizar a lista de alarmes recomendados em Alarmes recomendados.

Será feita uma cobrança pelos alarmes que você criar, na mesma taxa de quaisquer outros alarmes que você criar no CloudWatch. O uso das recomendações não acarreta custos adicionais. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

Localizar e criar alarmes recomendados

Siga estas etapas para encontrar as métricas para as quais o CloudWatch recomenda que você defina alarmes e, opcionalmente, para criar um desses alarmes. O primeiro procedimento explica como encontrar as métricas que têm alarmes recomendados e como criar um desses alarmes.

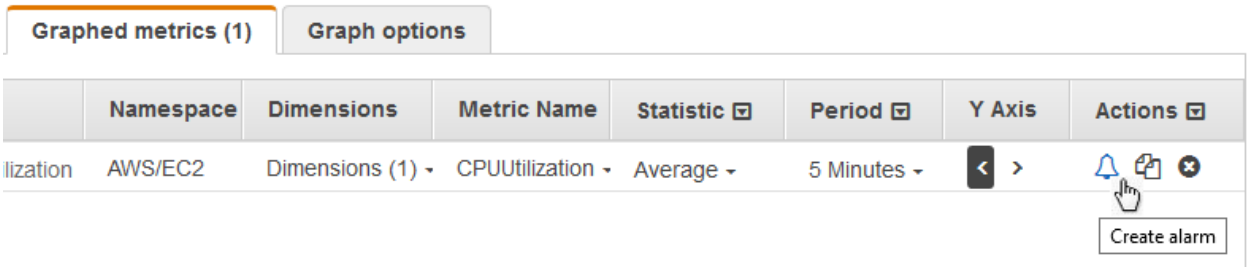
Você também pode fazer o download em massa das definições de alarme de infraestrutura como código para todos os alarmes recomendados em um namespace da AWS, como `AWS/Lambda` ou `AWS/S3`. Essas instruções são apresentadas mais adiante neste tópico.

Localizar as métricas com alarmes recomendados e criar um único alarme recomendado

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), All metrics (Todas as métricas).
3. Acima da tabela Métricas, selecione Recomendações de alarme.

   A lista de namespaces de métricas é filtrada para incluir apenas as métricas que têm recomendações de alarme e que os serviços em sua conta estão publicando.
4. Escolha o namespace para um serviço.

   A lista de métricas nesse namespace é filtrada para incluir apenas aquelas que têm recomendações de alarme.
5. Para visualizar a intenção do alarme e o limite recomendado para uma métrica, selecione Exibir detalhes.
6. Para criar um alarme para uma das métricas, siga um destes procedimentos:
   - Para usar o console para criar o alarme, faça o seguinte:
     a. Marque a caixa de seleção da métrica e selecione a guia Métricas representadas graficamente.
     b. Escolha o ícone de alarme.



O assistente de criação de alarme é exibido, com o nome da métrica, a estatística e o período preenchidos com base na recomendação de alarme. Se a recomendação incluir um valor limite específico, esse valor também será pré-preenchido.

c. Escolha Próximo.

d. Em Notificação, selecione um tópico do SNS para receber notificações quando o alarme transitar para o estado `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

Para que o alarme envie várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

Para que o alarme não envie notificações, escolha Remove (Remover).

e. Para que o alarme execute o Auto Scaling ou ações do EC2, escolha o botão apropriado e escolha o estado do alarme e a ação a ser executada.

f. Quando terminar, escolha Next (Próximo).

g. Digite um nome e uma descrição para o alarme. O nome deve conter somente caracteres ASCII. Em seguida, escolha Próximo.

h. Em Preview and create (Visualizar e criar), confirme se as informações e condições são o que você deseja e escolha Create alarm (Criar alarme).

- Para fazer o download de uma definição de alarme de infraestrutura como código para usar no AWS CloudFormation, na AWS CLI ou no Terraform, escolha Baixar código de alarme e selecione o formato desejado. O código baixado terá as configurações recomendadas para o nome da métrica, a estatística e o limite.

Fazer download das definições de alarme de infraestrutura como código para todos os alarmes recomendados para um serviço da AWS

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), All metrics (Todas as métricas).
3. Acima da tabela Métricas, selecione Recomendações de alarme.

   A lista de namespaces de métricas é filtrada para incluir apenas as métricas que têm recomendações de alarme e que os serviços em sua conta estão publicando.
4. Escolha o namespace para um serviço.

A lista de métricas nesse namespace é filtrada para incluir apenas aquelas que têm recomendações de alarme.

5. A opção Baixe o código de alarme exibe o número de alarmes recomendado para as métricas nesse namespace. Para baixar definições de alarme de infraestrutura como código para todos os alarmes recomendados, selecione Download do código de alarme e, em seguida, escolha o formato de código desejado.

Alarmes recomendados

As seções a seguir listam as métricas para as quais recomendamos que você defina alarmes de práticas recomendadas. Para cada métrica, também são exibidas as dimensões, a intenção do alarme, o limite recomendado, a justificativa do limite, a duração do período e o número de pontos de dados.

Algumas métricas podem aparecer duas vezes na lista. Isso acontece quando alarmes diferentes são recomendados para combinações diferentes de dimensões dessa métrica.

Pontos de dados para alarme é o número de pontos de dados que devem ser violados para enviar o alarme para o estado ALARME. Períodos de avaliação é o número de períodos que são levados em conta quando o alarme é avaliado. Se esses números forem iguais, o alarme entrará em estado ALARME somente quando esse número de períodos consecutivos tiver valores que ultrapassem o limite. Se Pontos de dados para alarme for menor que os Períodos de avaliação, será um alarme "M de N" e o alarme entrará em estado ALARME se pelo menos os pontos de dados de Pontos de dados para alarme estiverem violando qualquer conjunto de pontos de dados de Períodos de avaliação. Para ter mais informações, consulte Avaliar um alarme.

Tópicos

- Amazon API Gateway
- Amazon EC2 Auto Scaling
- Amazon CloudFront
- Amazon Cognito
- Amazon DynamoDB
- Amazon EBS
- Amazon EC2
- Amazon ElastiCache

- Amazon EC2 (AWS/ElasticGPUs)
- Amazon ECS
- Amazon ECS com o Container Insights
- Amazon EFS
- Amazon EKS com o Container Insights
- Amazon Kinesis Data Streams
- Lambda
- Lambda Insights
- Amazon VPC (AWS/NATGateway)
- Link privado da AWS (AWS/PrivateLinkEndpoints)
- Link privado da AWS (AWS/PrivateLinkServices)
- Amazon RDS
- Amazon Route 53 Public Data Plane
- Amazon S3
- S3ObjectLambda
- Amazon SNS
- Amazon SQS
- AWS VPN

Amazon API Gateway

4XXError

Dimensões: ApiName, estágio

Descrição do alarme: esse alarme detecta uma alta taxa de erros do lado do cliente. Isso pode indicar um problema na autorização ou nos parâmetros da solicitação do cliente. Isso também pode significar que um recurso foi removido ou que um cliente está solicitando um recurso que não existe. Considere a possibilidade de ativar o CloudWatch Logs e verificar se há algum erro que possa estar causando os erros 4XX. Ademais, considere a possibilidade de ativar as métricas detalhadas do CloudWatch para visualizar essa métrica por recurso e método e restringir a origem dos erros. Os erros também podem ser causados por exceder o limite de controle de utilização configurado. Se as respostas e os logs estiverem relatando taxas altas e inesperadas de erros 429, siga este guia para solucionar esse problema.

Intenção: esse alarme pode detectar altas taxas de erros no lado do cliente para as solicitações do API Gateway.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: o limite sugerido detecta quando mais de 5% do total de solicitações estão recebendo erros 4XX. No entanto, você pode ajustar o limite para se adequar ao tráfego das solicitações, bem como às taxas de erro aceitáveis. Você também pode analisar dados históricos para determinar a taxa de erro aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente. Os erros 4XX que ocorrem com frequência precisam receber um alarme. No entanto, definir um valor muito baixo para o limite pode fazer com que o alarme seja muito sensível.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

5XXError

Dimensões: ApiName, estágio

Descrição do alarme: esse alarme detecta uma alta taxa de erros do lado do cliente. Isso pode indicar que há algo errado no back-end da API, na rede ou na integração entre o gateway da API e a API de back-end. Essa documentação pode ajudar a solucionar a causa dos erros 5xx.

Intenção: esse alarme pode detectar altas taxas de erros no lado do servidor para as solicitações do API Gateway.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: o limite sugerido detecta quando mais de 5% do total de solicitações estão recebendo erros 5XX. No entanto, é possível ajustar o limite de acordo com o tráfego das solicitações e com as taxas de erro aceitáveis. Você também pode analisar dados históricos para determinar a taxa de erro aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente. Os erros 5XX que ocorrem com frequência precisam receber um alarme.

No entanto, definir um valor muito baixo para o limite pode fazer com que o alarme seja muito sensível.

Período: 60

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: GREATER_THAN_THRESHOLD

Contagem

Dimensões: ApiName, estágio

Descrição do alarme: esse alarme ajuda a detectar um baixo volume de tráfego para o estágio da API REST. Isso pode ser um indicador de um problema com a aplicação que chama a API, como o uso de endpoints incorretos. Também pode ser um indicador de um problema com a configuração ou as permissões da API, tornando-a inacessível para os clientes.

Intenção: esse alarme pode detectar um volume de tráfego inesperadamente baixo para o estágio da API REST. Recomendamos que você crie esse alarme se sua API receber um número previsível e consistente de solicitações em condições normais. Caso as métricas detalhadas do CloudWatch estejam ativadas e você possa prever o volume normal de tráfego por método e recurso, recomendamos que você crie alarmes alternativos para ter um monitoramento mais detalhado das quedas de volume de tráfego para cada recurso e método. Esse alarme não é recomendado para APIs que não esperam tráfego constante e consistente.

Estatística: SampleCount

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite com base na análise de dados históricos para determinar qual é a contagem de solicitações de linha de base esperada para sua API. Definir o limite em um valor muito alto pode fazer com que o alarme seja muito sensível em períodos de baixo tráfego normal e esperado. Por outro lado, configurá-lo em um valor muito baixo pode fazer com que o alarme não perceba quedas menores e anômalas no volume de tráfego.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: LESS_THAN_THRESHOLD

Contagem

Dimensões: ApiName, estágio, recurso, método

Descrição do alarme: esse alarme ajuda a detectar um baixo volume de tráfego para o recurso e o método da API REST no estágio. Isso pode indicar um problema com a aplicação que está chamando a API, como o uso de endpoints incorretos. Também pode ser um indicador de um problema com a configuração ou as permissões da API, tornando-a inacessível para os clientes.

Intenção: esse alarme pode detectar um volume de tráfego inesperadamente baixo para o recurso e o método da API REST no estágio. Recomendamos que você crie esse alarme se sua API receber um número previsível e consistente de solicitações em condições normais. Esse alarme não é recomendado para APIs que não esperam tráfego constante e consistente.

Estatística: SampleCount

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite com base na análise de dados históricos para determinar qual é a contagem de solicitações de linha de base esperada para sua API. Definir o limite em um valor muito alto pode fazer com que o alarme seja muito sensível em períodos de baixo tráfego normal e esperado. Por outro lado, configurá-lo em um valor muito baixo pode fazer com que o alarme não perceba quedas menores e anômalas no volume de tráfego.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: LESS_THAN_THRESHOLD

Contagem

Dimensões: ApiId, estágio

Descrição do alarme: esse alarme ajuda a detectar um baixo volume de tráfego para o estágio da API HTTP. Isso pode indicar um problema com a aplicação que está chamando a API, como o uso de endpoints incorretos. Também pode ser um indicador de um problema com a configuração ou as permissões da API, tornando-a inacessível para os clientes.

Intenção: esse alarme pode detectar um volume de tráfego inesperadamente baixo para o estágio da API HTTP. Recomendamos que você crie esse alarme se sua API receber um número previsível e consistente de solicitações em condições normais. Caso as métricas detalhadas do CloudWatch estejam ativadas e você possa prever o volume normal de tráfego por rota, recomendamos que você crie alarmes alternativos a esse para que haja um monitoramento mais detalhado das quedas no volume de tráfego de cada rota. Esse alarme não é recomendado para APIs que não esperam tráfego constante e consistente.

Estatística: SampleCount

Limite recomendado: depende da sua situação

Justificativa do limite: defina o valor limite com base na análise de dados históricos para determinar qual é a contagem de solicitações de linha de base esperada para sua API. Definir o limite em um valor muito alto pode fazer com que o alarme seja muito sensível em períodos de baixo tráfego normal e esperado. Por outro lado, configurá-lo em um valor muito baixo pode fazer com que o alarme não perceba quedas menores e anômalas no volume de tráfego.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: LESS_THAN_THRESHOLD

Contagem

Dimensões: ApiId, estágio, recurso, método

Descrição do alarme: esse alarme ajuda a detectar um baixo volume de tráfego para a rota da API HTTP no estágio. Isso pode indicar um problema com a aplicação que está chamando a API, como o uso de endpoints incorretos. Isso também pode indicar um problema com a configuração ou as permissões da API, tornando-a inacessível para os clientes.

Intenção: esse alarme pode detectar um volume de tráfego inesperadamente baixo para a rota da API HTTP no estágio. Recomendamos que você crie esse alarme se sua API receber um número previsível e consistente de solicitações em condições normais. Esse alarme não é recomendado para APIs que não esperam tráfego constante e consistente.

Estatística: SampleCount

Limite recomendado: depende da sua situação

Justificativa do limite: defina o valor limite com base na análise de dados históricos para determinar qual é a contagem de solicitações de linha de base esperada para sua API. Definir o limite em um valor muito alto pode fazer com que o alarme seja muito sensível em períodos de baixo tráfego normal e esperado. Por outro lado, configurá-lo em um valor muito baixo pode fazer com que o alarme não perceba quedas menores e anômalas no volume de tráfego.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: LESS_THAN_THRESHOLD

IntegrationLatency

Dimensões: ApiId, estágio

Descrição do alarme: esse alarme ajuda a detectar se há alta latência de integração para as solicitações de API em um estágio. Você pode correlacionar o valor da métrica `IntegrationLatency` com a métrica de latência correspondente do seu back-end, como a métrica `Duration` para integrações do Lambda. Isso ajuda a determinar se o back-end da API está demorando mais para processar as solicitações dos clientes devido a problemas de performance ou se há alguma outra sobrecarga da inicialização ou da inicialização a frio. Além disso, considere a possibilidade de ativar o CloudWatch Logs para sua API e verificar se há erros nos registros que possam estar causando os problemas de alta latência. Além disso, considere a possibilidade de ativar as métricas detalhadas do CloudWatch para obter uma visão dessa métrica por rota, o que ajudará você a restringir a origem da latência da integração.

Intenção: esse alarme pode detectar quando as solicitações do API Gateway em um estágio têm uma alta latência de integração. Recomendamos esse alarme para APIs de WebSocket e o consideramos opcional para APIs HTTP porque elas já têm recomendações de alarme separadas para a métrica de latência. Caso as métricas detalhadas do CloudWatch estejam ativadas e você tenha diferentes requisitos de performance de latência de integração por rota, recomendamos que você crie alarmes alternativos para ter um monitoramento mais refinado da latência de integração para cada rota.

Estatística: p90

Limite recomendado: 2000,0

Justificativa do limite: o valor limite sugerido não funciona para todas as workloads da API. No entanto, você pode usá-lo como um ponto de partida para o limite. Em seguida, você pode escolher diferentes valores limite com base na workload e nos requisitos aceitáveis de latência, performance e SLA da API. Se for aceitável que a API tenha uma latência mais alta em geral, defina um valor limite mais alto para tornar o alarme menos sensível. No entanto, se for esperado que a API forneça respostas quase em tempo real, defina um valor limite mais baixo. Você também pode analisar os dados históricos para determinar a latência de linha de base esperada para a workload da aplicação e, em seguida, usá-la para ajustar o valor limite adequadamente.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

IntegrationLatency

Dimensões: ApiId, estágio, rota

Descrição do alarme: esse alarme ajuda a detectar se há alta latência de integração para as solicitações da API de WebSocket para uma rota em um estágio. Você pode correlacionar o valor da métrica `IntegrationLatency` com a métrica de latência correspondente do seu back-end, como a métrica `Duration` para integrações do Lambda. Isso ajuda a determinar se o back-end da API está demorando mais para processar solicitações de clientes devido a problemas de performance ou se há alguma outra sobrecarga de inicialização ou inicialização a frio. Além disso, considere a possibilidade de ativar o CloudWatch Logs para sua API e verificar se há erros nos registros que possam estar causando os problemas de alta latência.

Intenção: esse alarme pode detectar quando as solicitações do API Gateway para uma rota em um estágio têm alta latência de integração.

Estatística: p90

Limite recomendado: 2000,0

Justificativa do limite: o valor limite sugerido não funciona para todas as workloads da API. No entanto, você pode usá-lo como um ponto de partida para o limite. Em seguida, você pode escolher diferentes valores limite com base na workload e nos requisitos aceitáveis de latência, performance e SLA da API. Se for aceitável que a API tenha uma latência maior em geral,

você pode definir um valor limite mais alto para tornar o alarme menos sensível. No entanto, se for esperado que a API forneça respostas quase em tempo real, defina um valor limite mais baixo. Você também pode analisar os dados históricos para determinar a latência de linha de base esperada para a workload da aplicação e, em seguida, usá-la para ajustar o valor limite adequadamente.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latência

Dimensões: ApiName, estágio

Descrição do alarme: esse alarme detecta alta latência em um estágio. Encontre o valor da métrica `IntegrationLatency` para verificar a latência do back-end da API. Se as duas métricas estiverem alinhadas em sua maior parte, o back-end da API será a fonte de maior latência e você deve investigar se há problemas. Considere também ativar o CloudWatch Logs e verificar se há erros que possam estar causando a alta latência. Além disso, considere a possibilidade de ativar as métricas detalhadas do CloudWatch para visualizar essa métrica por recurso e método e restringir a origem da latência. Se aplicável, consulte os guias de solução de problemas com o Lambda ou de solução de problemas para endpoints de API otimizados para borda.

Intenção: esse alarme pode detectar quando as solicitações do API Gateway em um estágio têm alta latência. Se você tiver as métricas detalhadas do CloudWatch ativadas e tiver diferentes requisitos de performance de latência para cada método e recurso, recomendamos que crie alarmes alternativos para ter um monitoramento mais detalhado da latência de cada recurso e método.

Estatística: p90

Limite recomendado: 2500,0

Justificativa do limite: o valor limite sugerido não funciona para todas as workloads da API. No entanto, você pode usá-lo como um ponto de partida para o limite. Em seguida, você pode escolher diferentes valores limite com base na workload e nos requisitos aceitáveis de latência, performance e SLA da API. Se for aceitável que a API tenha uma latência maior em geral, você

pode definir um valor limite mais alto para tornar o alarme menos sensível. No entanto, se for esperado que a API forneça respostas quase em tempo real, defina um valor limite mais baixo. Você também pode analisar dados históricos para determinar qual é a latência de linha de base esperada para a workload da aplicação e, em seguida, ajustar o valor limite adequadamente.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latência

Dimensões: ApiName, estágio, recurso, método

Descrição do alarme: esse alarme detecta alta latência para um recurso e método em um estágio. Encontre o valor da métrica `IntegrationLatency` para verificar a latência do back-end da API. Se as duas métricas estiverem alinhadas, o back-end da API será a fonte de maior latência e você deve investigar se há problemas de performance. Considere também ativar o CloudWatch Logs e verificar se há algum erro que possa estar causando a alta latência. Você também pode consultar os guias de solução de problemas com o Lambda ou de solução de problemas para endpoints de API otimizados para borda, se aplicável.

Intenção: esse alarme pode detectar quando as solicitações do API Gateway para um recurso e método em um estágio têm alta latência.

Estatística: p90

Limite recomendado: 2500,0

Justificativa do limite: o valor limite sugerido não funciona para todas as workloads da API. No entanto, você pode usá-lo como um ponto de partida para o limite. Em seguida, você pode escolher diferentes valores limite com base na workload e nos requisitos aceitáveis de latência, performance e SLA da API. Se for aceitável que a API tenha uma latência maior em geral, você pode definir um valor limite mais alto para tornar o alarme menos sensível. No entanto, se for esperado que a API forneça respostas quase em tempo real, defina um valor limite mais baixo. Você também pode analisar dados históricos para determinar a latência de linha de base esperada para a workload da aplicação e, em seguida, ajustar o valor limite adequadamente.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latência

Dimensões: ApiId, estágio

Descrição do alarme: esse alarme detecta alta latência em um estágio. Encontre o valor da métrica `IntegrationLatency` para verificar a latência do back-end da API. Se as duas métricas estiverem alinhadas, o back-end da API será a fonte de maior latência e você deve investigar se há problemas de performance. Considere também ativar o CloudWatch Logs e verificar se há algum erro que possa estar causando a alta latência. Além disso, considere a possibilidade de ativar as métricas detalhadas do CloudWatch para visualizar essa métrica por rota e restringir a origem da latência. Você também pode consultar o guia de solução de problemas com integrações do Lambda, se aplicável.

Intenção: esse alarme pode detectar quando as solicitações do API Gateway em um estágio têm alta latência. Caso as métricas detalhadas do CloudWatch estejam ativadas e você tenha diferentes requisitos de performance de latência por rota, recomendamos que você crie alarmes alternativos para ter um monitoramento mais detalhado da latência de cada rota.

Estatística: p90

Limite recomendado: 2500,0

Justificativa do limite: o valor limite sugerido não funciona para todas as workloads da API. No entanto, ele pode ser usado como ponto de partida para o limite. Em seguida, você pode escolher diferentes valores limite com base na workload e na latência aceitável, na performance e nos requisitos de SLA da API. Se for aceitável que a API tenha uma latência mais alta em geral, você poderá definir um valor limite mais alto para torná-la menos sensível. No entanto, se for esperado que a API forneça respostas quase em tempo real, defina um valor limite mais baixo. Você também pode analisar dados históricos para determinar a latência de linha de base esperada para a workload da aplicação e, em seguida, ajustar o valor limite adequadamente.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latência

Dimensões: ApiId, estágio, recurso, método

Descrição do alarme: esse alarme detecta alta latência para uma rota em um estágio. Encontre o valor da métrica `IntegrationLatency` para verificar a latência do back-end da API. Se as duas métricas estiverem mais alinhadas, o back-end da API será a fonte de maior latência e deve-se investigar quanto a problemas de performance. Considere também ativar o CloudWatch Logs e verificar se há algum erro que possa estar causando a alta latência. Você também pode consultar o guia de solução de problemas com integrações do Lambda, se aplicável.

Intenção: esse alarme é usado para detectar quando as solicitações do API Gateway para uma rota em um estágio têm alta latência.

Estatística: p90

Limite recomendado: 2500,0

Justificativa do limite: o valor limite sugerido não funciona para todas as workloads da API. No entanto, ele pode ser usado como ponto de partida para o limite. Em seguida, você pode escolher diferentes valores limite com base na workload e nos requisitos aceitáveis de latência, performance e SLA da API. Se for aceitável que a API tenha uma latência maior em geral, você pode definir um valor limite mais alto para tornar o alarme menos sensível. No entanto, se for esperado que a API forneça respostas quase em tempo real, defina um valor limite mais baixo. Você também pode analisar dados históricos para determinar a latência de linha de base esperada para a workload da aplicação e, em seguida, ajustar o valor limite adequadamente.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

4xx

Dimensões: ApiId, estágio

Descrição do alarme: esse alarme detecta uma alta taxa de erros do lado do cliente. Isso pode indicar um problema na autorização ou nos parâmetros da solicitação do cliente. Isso também

pode significar que uma rota foi removida ou que um cliente está solicitando uma rota que não existe na API. Considere a possibilidade de ativar o CloudWatch Logs e verificar se há algum erro que possa estar causando os erros 4xx. Além disso, considere a possibilidade de ativar as métricas detalhadas do CloudWatch para visualizar essa métrica por rota, o que ajudará você a restringir a origem dos erros. Os erros também podem ser causados por exceder o limite de controle de utilização configurado. Se as respostas e os logs estiverem relatando taxas altas e inesperadas de erros 429, siga este guia para solucionar esse problema.

Intenção: esse alarme pode detectar altas taxas de erros no lado do cliente para as solicitações do API Gateway.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: o limite sugerido detecta quando mais de 5% do total de solicitações estão recebendo erros 4xx. No entanto, você pode ajustar o limite para se adequar ao tráfego das solicitações, bem como às taxas de erro aceitáveis. Você também pode analisar dados históricos para determinar a taxa de erro aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente. Os erros 4xx que ocorrem com frequência precisam receber um alarme. No entanto, definir um valor muito baixo para o limite pode fazer com que o alarme seja muito sensível.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

5xx

Dimensões: ApiId, estágio

Descrição do alarme: esse alarme detecta uma alta taxa de erros do lado do cliente. Isso pode indicar que há algo errado no back-end da API, na rede ou na integração entre o gateway da API e a API de back-end. Essa documentação pode ajudar você a solucionar a causa dos erros 5xx.

Intenção: esse alarme pode detectar altas taxas de erros no lado do servidor para as solicitações do API Gateway.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: o limite sugerido detecta quando mais de 5% do total de solicitações estão recebendo erros 5xx. No entanto, você pode ajustar o limite para se adequar ao tráfego das solicitações, bem como às taxas de erro aceitáveis. Você também pode analisar dados históricos para determinar qual é a taxa de erro aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente. Os erros 5xx que ocorrem com frequência precisam receber um alarme. No entanto, definir um valor muito baixo para o limite pode fazer com que o alarme seja muito sensível.

Período: 60

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: GREATER_THAN_THRESHOLD

MessageCount

Dimensões: ApiId, estágio

Descrição do alarme: esse alarme ajuda a detectar baixo volume de tráfego para o estágio da API de WebSocket. Isso pode indicar um problema quando os clientes chamam a API, como o uso de endpoints incorretos ou problemas com o back-end que envia mensagens aos clientes. Isso também pode indicar um problema com a configuração ou as permissões da API, tornando-a inacessível para os clientes.

Intenção: esse alarme pode detectar um volume de tráfego inesperadamente baixo para o estágio da API de WebSocket. Recomendamos que você crie esse alarme se a sua API receber e enviar um número previsível e consistente de mensagens em condições normais. Caso as métricas detalhadas do CloudWatch estejam ativadas e você possa prever o volume normal de tráfego por rota, é melhor criar alarmes alternativos a esse para ter um monitoramento mais refinado das quedas no volume de tráfego para cada rota. Não recomendamos esse alarme para APIs que não esperam tráfego constante e consistente.

Estatística: SampleCount

Limite recomendado: depende da sua situação

Justificativa do limite: defina o valor limite com base na análise de dados históricos para determinar qual é a contagem de mensagens de linha de base esperada para sua API. Definir o limite em um valor muito alto pode fazer com que o alarme seja muito sensível em períodos de tráfego normal e baixo esperado. Por outro lado, a definição de um valor muito baixo pode fazer com que o alarme não perceba quedas menores e anômalas no volume de tráfego.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: LESS_THAN_THRESHOLD

MessageCount

Dimensões: ApiId, estágio, rota

Descrição do alarme: esse alarme ajuda a detectar um baixo volume de tráfego para a rota da API de WebSocket no estágio. Isso pode indicar um problema com os clientes que chamam a API, como o uso de endpoints incorretos, ou problemas com o back-end que envia mensagens aos clientes. Isso também pode indicar um problema com a configuração ou as permissões da API, tornando-a inacessível para os clientes.

Intenção: esse alarme pode detectar um volume de tráfego inesperadamente baixo para a rota da API de WebSocket no estágio. Recomendamos que você crie esse alarme se a sua API receber e enviar um número previsível e consistente de mensagens em condições normais. Não recomendamos esse alarme para APIs que não esperam tráfego constante e consistente.

Estatística: SampleCount

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite com base na análise de dados históricos para determinar qual é a contagem de mensagens de linha de base esperada para sua API. Definir o limite em um valor muito alto pode fazer com que o alarme seja muito sensível em períodos de tráfego normal e baixo esperado. Por outro lado, a definição de um valor muito baixo pode fazer com que o alarme não perceba quedas menores e anômalas no volume de tráfego.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: LESS_THAN_THRESHOLD

ClientError

Dimensões: ApiId, estágio

Descrição do alarme: esse alarme detecta uma alta taxa de erros do cliente. Isso pode indicar um problema nos parâmetros de autorização ou de mensagem. Isso também pode significar que uma rota foi removida ou que um cliente está solicitando uma rota que não existe na API. Considere a possibilidade de ativar o CloudWatch Logs e verificar se há algum erro que possa estar causando os erros 4xx. Além disso, considere a possibilidade de ativar as métricas detalhadas do CloudWatch para visualizar essa métrica por rota, o que ajudará você a restringir a origem dos erros. Os erros também podem ser causados por exceder o limite de controle de utilização configurado. Se as respostas e os logs estiverem relatando taxas altas e inesperadas de erros 429, siga este guia para solucionar esse problema.

Intenção: esse alarme pode detectar altas taxas de erros do cliente para as mensagens do API Gateway de WebSocket.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: o limite sugerido detecta quando mais de 5% do total de solicitações estão recebendo erros 4xx. Você pode ajustar o limite de acordo com o tráfego das solicitações e com as taxas de erro aceitáveis. Você também pode analisar dados históricos para determinar a taxa de erro aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente. Os erros 4xx que ocorrem com frequência precisam receber um alarme. No entanto, definir um valor muito baixo para o limite pode fazer com que o alarme seja muito sensível.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

ExecutionError

Dimensões: ApiId, estágio

Descrição do alarme: esse alarme ajuda a detectar uma alta taxa de erros de execução. Isso pode ser causado por erros 5xx de sua integração, problemas de permissão ou outros fatores que impedem a invocação bem-sucedida da integração, como o fato de a integração ter passado por um controle de utilização ou ter sido excluída. Considere ativar o CloudWatch Logs para sua API e verificar os registros quanto ao tipo e à causa dos erros. Além disso, considere a possibilidade de ativar as métricas detalhadas do CloudWatch para obter uma visão dessa métrica por rota, o que ajudará você a restringir a origem dos erros. Essa documentação também pode ajudá-lo a solucionar a causa de qualquer erro de conexão.

Intenção: esse alarme pode detectar altas taxas de erros de execução para as mensagens da API Gateway de WebSocket.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: o limite sugerido detecta quando mais de 5% do total de solicitações estão recebendo erros de execução. Você pode ajustar o limite para se adequar ao tráfego das solicitações, bem como para se adequar às taxas de erro aceitáveis. Você pode analisar dados históricos para determinar a taxa de erro aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente. Os erros de execução que ocorrem com frequência precisam receber um alarme. No entanto, definir um valor muito baixo para o limite pode fazer com que o alarme seja muito sensível.

Período: 60

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon EC2 Auto Scaling

GroupInServiceCapacity

Dimensões: AutoScalingGroupName

Descrição do alarme: esse alarme ajuda a detectar quando a capacidade do grupo está abaixo da capacidade desejada para sua workload. Para solucionar problemas, verifique se há falhas de

inicialização em suas atividades de escalonamento e confirme se a configuração da capacidade desejada está correta.

Intenção: esse alarme pode detectar uma baixa disponibilidade em seu grupo do Auto Scaling devido a falhas de inicialização ou inicializações suspensas.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite deve ser a capacidade mínima necessária para executar sua workload. Na maioria dos casos, é possível definir isso para corresponder à métrica GroupDesiredCapacity.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: LESS_THAN_THRESHOLD

Amazon CloudFront

5xxErrorRate

Dimensões: DistributionID, Região=Global

Descrição do alarme: esse alarme monitora a porcentagem de respostas de erro 5xx do seu servidor de origem para ajudar você a detectar se o serviço CloudFront está com problemas. Consulte Como solucionar problemas de respostas de erro da sua origem para obter informações que ajudem a entender os problemas com o servidor. Além disso, ative as métricas adicionais para obter métricas de erro detalhadas.

Intenção: esse alarme é usado para detectar problemas com o atendimento de solicitações do servidor de origem ou problemas com a comunicação entre o CloudFront e seu servidor de origem.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme é altamente dependente da tolerância para respostas 5xx. Você pode analisar dados históricos e tendências e, em seguida, definir o limite adequadamente. Como os erros 5xx podem ser causados por problemas transitórios, recomendamos que você defina o limite como um valor maior que 0 para que o alarme não seja muito sensível.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

OriginLatency

Dimensões: DistributionID, Região=Global

Descrição do alarme: o alarme ajuda a monitorar se o servidor de origem está demorando muito para responder. Se o servidor demorar muito para responder, isso pode levar a um tempo limite. Consulte Encontre e corrija respostas atrasadas de aplicações no seu servidor de origem se você tiver valores `OriginLatency` consistentemente altos.

Intenção: esse alarme é usado para detectar problemas com o servidor de origem que está demorando muito para responder.

Estatística: p90

Limite recomendado: depende da sua situação

Justificativa do limite: você deve calcular o valor de cerca de 80% do tempo limite da resposta de origem e usar o resultado como valor limite. Se essa métrica estiver consistentemente próxima do valor de tempo limite da resposta de origem, é possível que você comece a receber erros 504.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

FunctionValidationErrors

Dimensões: DistributionID, FunctionName, Região=Global

Descrição do alarme: esse alarme ajuda a monitorar os erros de validação do CloudFront Functions para que você possa tomar medidas para resolvê-los. Analise os logs do CloudWatch Functions e observe o código da função para encontrar e resolver a causa-raiz do problema. Consulte Restrições nas funções de borda para entender as configurações incorretas comuns do CloudFront Functions.

Intenção: esse alarme é usado para detectar erros de validação do CloudFront Functions.

Estatística: soma

Limite recomendado: 0,0

Justificativa do limite: um valor maior que 0 indica um erro de validação. Recomendamos definir o limite como 0, pois os erros de validação implicam um problema quando o CloudFront Functions é transferido de volta para o CloudFront. Por exemplo, o CloudFront precisa do cabeçalho HTTP Host para processar uma solicitação. Não há nada que impeça um usuário de excluir o cabeçalho Host em seu código do CloudFront Functions. Mas quando o CloudFront recebe a resposta de volta e o cabeçalho Host está ausente, o CloudFront gera um erro de validação.

Período: 60

Pontos de dados para o alarme: 2

Períodos de avaliação: 2

Operador de comparação: GREATER_THAN_THRESHOLD

FunctionExecutionErrors

Dimensões: DistributionID, FunctionName, Região=Global

Descrição do alarme: esse alarme ajuda a monitorar os erros de execução do CloudFront Functions para que você possa tomar medidas para resolvê-los. Analise os logs do CloudWatch Functions e observe o código da função para encontrar e resolver a causa-raiz do problema.

Intenção: esse alarme é usado para detectar erros de execução do CloudFront Functions.

Estatística: soma

Limite recomendado: 0,0

Justificativa do limite: recomendamos definir o limite como 0, pois um erro de execução indica um problema com o código que ocorre no runtime.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

FunctionThrottles

Dimensões: DistributionID, FunctionName, Região=Global

Descrição do alarme: esse alarme ajuda você a monitorar se o CloudFront Functions está com controle de utilização. Caso sua função esteja com um controle de utilização, isso significa que ela está demorando muito para ser executada. Para evitar o controle de utilização de funções, considere otimizar o código da função.

Intenção: esse alarme pode detectar quando o CloudFront Functions está com um controle de utilização, de modo que você possa reagir e resolver o problema para proporcionar uma experiência tranquila ao cliente.

Estatística: soma

Limite recomendado: 0,0

Justificativa do limite: recomendamos definir o limite como 0 para permitir uma resolução mais rápida dos controles de utilização da função.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon Cognito

SignUpThrottles

Dimensões: UserPool, UserPoolClient

Descrição do alarme: esse alarme monitora a contagem de solicitações com controle de utilização. Se os usuários estiverem constantemente com um controle de utilização, você

deverá aumentar o limite solicitando um aumento da cota de serviços. Consulte Cotas no Amazon Cognito para saber como solicitar um aumento de cotas. Para tomar medidas proativas, considere monitorar a cota de uso.

Intenção: esse alarme ajuda a monitorar a ocorrência de solicitações de cadastro com controle de utilização. Isso pode ajudar você a saber quando tomar medidas para atenuar qualquer degradação na experiência de cadastro. O controle de utilização contínuo das solicitações é uma experiência negativa de cadastro do usuário.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: um grupo de usuários bem provisionado não deve encontrar nenhum controle de utilização que se estenda por vários pontos de dados. Portanto, um limite típico para uma workload esperada deve ser zero. Para uma workload irregular com picos frequentes, você pode analisar dados históricos para determinar o controle de utilização aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente. Uma solicitação de controle de utilização deve ser testada novamente para minimizar o impacto na aplicação.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

SignInThrottles

Dimensões: UserPool, UserPoolClient

Descrição do alarme: esse alarme monitora a contagem de solicitações com controle de utilização de autenticação do usuário. Caso os usuários estejam constantemente passando por controle de utilização, talvez seja necessário aumentar o limite solicitando um aumento da cota de serviços. Consulte Cotas no Amazon Cognito para saber como solicitar um aumento de cotas. Para tomar medidas proativas, considere monitorar a cota de uso.

Intenção: esse alarme ajuda a monitorar a ocorrência de solicitações de login com controle de utilização. Isso pode ajudar você a saber quando tomar medidas para atenuar qualquer degradação na experiência de login. O controle de utilização contínuo das solicitações é uma experiência ruim de autenticação do usuário.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: um grupo de usuários bem provisionado não deve encontrar nenhum controle de utilização que se estenda por vários pontos de dados. Portanto, um limite típico para uma workload esperada deve ser zero. Para uma workload irregular com picos frequentes, você pode analisar dados históricos para determinar o controle de utilização aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente. Uma solicitação de controle de utilização deve ser testada novamente para minimizar o impacto na aplicação.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

TokenRefreshThrottles

Dimensões: UserPool, UserPoolClient

Descrição do alarme: você pode definir o valor do limite para se adequar ao tráfego da solicitação, bem como para corresponder ao controle de utilização aceitável para solicitações de atualização de token. O controle de utilização é usado para proteger seu sistema de muitas solicitações. No entanto, é importante monitorar se o provisionamento também é insuficiente para o seu tráfego normal. Você pode analisar dados históricos para encontrar o controle de utilização aceitável para a workload da aplicação e, em seguida, ajustar o limite de alarme para que ele seja maior do que o nível de controle de utilização aceitável. As solicitações de controle de utilização devem ser repetidas pela aplicação/serviço, pois são transitórias. Portanto, um valor muito baixo para o limite pode fazer com que o alarme seja sensível.

Intenção: esse alarme ajuda a monitorar a ocorrência de solicitações de atualização de token com controle de utilização. Isso pode ajudar você a saber quando tomar medidas para atenuar possíveis problemas, para garantir uma experiência de usuário tranquila e a integridade e confiabilidade do seu sistema de autenticação. O controle de utilização contínuo das solicitações é uma experiência ruim de autenticação do usuário.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite também pode ser definido/ajustado para se adequar ao tráfego da solicitação, bem como ao controle de utilização aceitável para solicitações de atualização de token. O controle de utilização existe para proteger seu sistema de muitas solicitações. No entanto, é importante monitorar se o provisionamento para o tráfego normal também está insuficiente e verificar se isso está causando o impacto. Os dados históricos também podem ser analisados para verificar qual é o controle de utilização aceitável para a workload da aplicação, e o limite pode ser ajustado para um nível mais alto do que o controle de utilização aceitável habitual. As solicitações de controle de utilização devem ser repetidas pela aplicação/serviço, pois são transitórias. Portanto, um valor muito baixo para o limite pode fazer com que o alarme seja sensível.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

FederationThrottles

Dimensões: UserPool, UserPoolClient, IdentityProvider

Descrição do alarme: esse alarme monitora a contagem de solicitações de federação de identidades com controle de utilização. Se você observar um controle de utilização constante, isso pode indicar que é necessário aumentar o limite solicitando um aumento da cota de serviços. Consulte Cotas no Amazon Cognito para saber como solicitar um aumento de cotas.

Intenção: esse alarme ajuda a monitorar a ocorrência de solicitações de federação de identidades com controle de utilização. Isso pode ajudar você a responder proativamente a gargalos de performance ou configurações incorretas e garantir uma experiência de autenticação tranquila para seus usuários. O controle de utilização contínuo das solicitações é uma experiência ruim de autenticação do usuário.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: você pode definir o limite para se adequar ao tráfego da solicitação, bem como para corresponder ao controle de utilização aceitável para solicitações de federação de identidades. O controle de utilização é usado para proteger seu sistema de um número excessivo

de solicitações. No entanto, é importante monitorar se o provisionamento também é insuficiente para o seu tráfego normal. Você pode analisar os dados históricos para encontrar o controle de utilização aceitável para a workload da aplicação e, em seguida, definir o limite para um valor acima do nível de controle de utilização aceitável. As solicitações de controle de utilização devem ser repetidas pela aplicação/serviço, pois são transitórias. Portanto, um valor muito baixo para o limite pode fazer com que o alarme seja sensível.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon DynamoDB

AccountProvisionedReadCapacityUtilization

Dimensões: nenhuma

Descrição do alarme: esse alarme detecta se a capacidade de leitura da conta está atingindo o limite provisionado. Se isso ocorrer, você poderá aumentar a cota da conta para utilização da capacidade de leitura. Você pode visualizar suas cotas atuais para unidades de capacidade de leitura e solicitar aumentos usando o Service Quotas.

Intenção: o alarme pode detectar se a utilização da capacidade de leitura da conta está se aproximando da utilização da capacidade de leitura provisionada. Se a utilização atingir seu limite máximo, o DynamoDB começará a realizar o controle de utilização nas solicitações de leitura.

Estatística: máxima

Limite recomendado: 80,0

Justificativa do limite: defina o limite para 80%, de modo que uma medida (como aumentar os limites da conta) possa ser tomada antes de atingir a capacidade total para evitar o controle de utilização.

Período: 300

Pontos de dados para o alarme: 2

Períodos de avaliação: 2

Operador de comparação: GREATER_THAN_THRESHOLD

AccountProvisionedWriteCapacityUtilization

Dimensões: nenhuma

Descrição do alarme: esse alarme detecta se a capacidade de gravação da conta está atingindo o limite provisionado. Se isso ocorrer, você poderá aumentar a cota da conta para utilização da capacidade de gravação. Você pode visualizar suas cotas atuais para unidades de capacidade de gravação e solicitar aumentos usando o Service Quotas.

Intenção: esse alarme pode detectar se a utilização da capacidade de gravação da conta está se aproximando da utilização da capacidade de gravação provisionada. Se a utilização atingir seu limite máximo, o DynamoDB começará a realizar o controle de utilização das solicitações de gravação.

Estatística: máxima

Limite recomendado: 80,0

Justificativa do limite: defina o limite para 80%, para que a medida (como aumentar os limites da conta) possa ser tomada antes de atingir a capacidade total para evitar o controle de utilização.

Período: 300

Pontos de dados para o alarme: 2

Períodos de avaliação: 2

Operador de comparação: GREATER_THAN_THRESHOLD

AgeOfOldestUnreplicatedRecord

Dimensões: TableName, DelegateDoperation

Descrição do alarme: esse alarme detecta o atraso na replicação para um fluxo de dados do Kinesis. Em operação normal, `AgeOfOldestUnreplicatedRecord` deve estar na ordem dos milissegundos. Esse número cresce com base em tentativas de replicação malsucedidas causadas por escolhas de configuração controladas pelo cliente. Exemplos de configurações controladas pelo cliente que levam a tentativas de replicação malsucedidas são uma capacidade de fluxo de dados do Kinesis com provisionamento insuficiente que leva a um controle de

utilização excessivo ou uma atualização manual das políticas de acesso do fluxo de dados do Kinesis que impede o DynamoDB de adicionar dados ao fluxo de dados. Para manter essa métrica o mais baixa possível, você precisa garantir o provisionamento correto da capacidade do fluxo de dados do Kinesis e certificar-se de que as permissões do DynamoDB não foram alteradas.

Intenção: esse alarme pode monitorar tentativas de replicação malsucedidas e o atraso resultante na replicação para o fluxo de dados do Kinesis.

Estatística: máxima

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite de acordo com o atraso de replicação desejado, medido em milissegundos. Esse valor depende dos requisitos de sua workload e da performance esperada.

Período: 300

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: GREATER_THAN_THRESHOLD

FailedToReplicateRecordCount

Dimensões: TableName, DelegateDoperation

Descrição do alarme: esse alarme detecta o número de registros que o DynamoDB não conseguiu replicar para o fluxo de dados do Kinesis. Determinados itens com mais de 34 KB podem se expandir em tamanho para alterar registros de dados maiores que o limite de 1 MB para itens do Kinesis Data Streams. Essa expansão de tamanho ocorre quando os itens com mais de 34 KB contêm um grande número de valores de atributos booleanos ou vazios. Valores de atributos booleanos e vazios são armazenados como 1 byte no DynamoDB, mas expandem até 5 bytes quando são serializados usando JSON padrão para replicação do Kinesis Data Streams. O DynamoDB não consegue replicar esses registros de alteração para o fluxo de dados do Kinesis. O DynamoDB ignora esses registros de dados de alteração e continua automaticamente a replicar registros subsequentes.

Intenção: esse alarme pode monitorar o número de registros que o DynamoDB não conseguiu replicar para o seu fluxo de dados do Kinesis devido ao limite de tamanho de item dos fluxos de dados do Kinesis.

Estatística: soma

Limite recomendado: 0,0

Justificativa do limite: defina o limite como 0 para detectar quaisquer registros que o DynamoDB não conseguiu replicar.

Período: 60

Pontos de dados para o alarme: 1

Períodos de avaliação: 1

Operador de comparação: GREATER_THAN_THRESHOLD

ReadThrottleEvents

Dimensões: TableName

Descrição do alarme: esse alarme detecta se há um grande número de solicitações de leitura passando por um controle de utilização para a tabela do DynamoDB. Para solucionar o problema, consulte Solução de problemas de controle de utilização no Amazon DynamoDB.

Intenção: esse alarme pode detectar o controle de utilização contínuo de solicitações de leitura na tabela do DynamoDB. O controle de utilização contínuo das solicitações de leitura pode afetar negativamente as operações de leitura de sua workload e reduzir a eficiência geral do sistema.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite de acordo com o tráfego de leitura esperado para a tabela do DynamoDB, levando em conta um nível aceitável de controle de utilização. É importante monitorar se você está com provisionamento insuficiente e não está causando um controle de utilização consistente. Você também pode analisar os dados históricos para encontrar o nível de controle de utilização aceitável para a workload da aplicação e, em seguida, ajustar o limite para que seja mais alto do que o nível de controle de utilização habitual. As solicitações com controle de utilização devem ser repetidas pela aplicação ou serviço, pois são transitórias. Portanto, um limite muito baixo pode fazer com que o alarme seja sensível demais, causando transições de estado indesejadas.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

ReadThrottleEvents

Dimensões: TableName, GlobalSecondaryIndexName

Descrição do alarme: esse alarme detecta se há um grande número de solicitações de leitura sendo limitadas para o índice secundário global da tabela do DynamoDB. Para solucionar o problema, consulte Solução de problemas de controle de utilização no Amazon DynamoDB.

Intenção: o alarme pode detectar o controle de utilização contínuo de solicitações de leitura para o índice secundário global da tabela do DynamoDB. O controle de utilização contínuo das solicitações de leitura pode afetar negativamente as operações de leitura de sua workload e reduzir a eficiência geral do sistema.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite de acordo com o tráfego de leitura esperado para a tabela do DynamoDB, levando em conta um nível aceitável de controle de utilização. É importante monitorar se você está com provisionamento insuficiente e não está causando um controle de utilização consistente. Você também pode analisar os dados históricos para encontrar um nível de controle de utilização aceitável para a workload da aplicação e, em seguida, ajustar o limite para que seja mais alto do que o nível de controle de utilização aceitável habitual. As solicitações com controle de utilização devem ser repetidas pela aplicação ou serviço, pois são transitórias. Portanto, um limite muito baixo pode fazer com que o alarme seja sensível demais, causando transições de estado indesejadas.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

ReplicationLatency

Dimensões: TableName, ReceivingRegion

Descrição do alarme: o alarme detecta se a réplica em uma região para a tabela global está atrasada em relação à região de origem. A latência poderá aumentar se uma região da AWS ficar degradada e você tiver uma tabela de réplica nessa região. Nesse caso, você pode redirecionar temporariamente as atividades de leitura e gravação da aplicação para outra região da AWS. Se você estiver usando 2017.11.29 (herdado) de tabelas globais, verifique se as unidades de capacidade de gravação (WCUs) são idênticas para cada uma das tabelas de réplica. Você também pode seguir as recomendações em Práticas recomendadas e requisitos de gerenciamento de capacidade.

Intenção: o alarme pode detectar se a tabela de réplica em uma região está ficando para trás na replicação das alterações de outra região. Isso pode fazer com que sua réplica se desvie das outras réplicas. É útil saber a latência de replicação de cada região da AWS e alertar se essa latência de replicação aumenta continuamente. A replicação da tabela se aplica somente a tabelas globais.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme depende muito do seu caso de uso. Latências de replicação superiores a três minutos geralmente são motivo de investigação. Analise a importância e os requisitos do atraso de replicação e analise as tendências históricas. Em seguida, selecione o limite adequadamente.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

SuccessfulRequestLatency

Dimensões: TableName, operação

Descrição do alarme: esse alarme detecta uma alta latência para a operação da tabela do DynamoDB (indicada pelo valor da dimensão da `Operation` no alarme). Consulte este documento de solução de problemas para solucionar problemas de latência no Amazon DynamoDB.

Intenção: esse alarme pode detectar uma alta latência para a operação da tabela do DynamoDB. A latência mais alta para as operações pode afetar negativamente a eficiência geral do sistema.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: o DynamoDB fornece latência de um dígito de milissegundos, em média, para as operações singleton, como GetItem, PutItem e assim por diante. No entanto, é possível definir o limite com base na tolerância aceitável para a latência do tipo de operação e da tabela envolvida na workload. Você pode analisar os dados históricos dessa métrica para descobrir a latência usual da operação da tabela e, em seguida, definir o limite para um número que represente o atraso crítico da operação.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: GREATER_THAN_THRESHOLD

SystemErrors

Dimensões: TableName

Descrição do alarme: esse alarme detecta um número alto e contínuo de erros de sistema para as solicitações de tabela do DynamoDB. Se você continuar recebendo erros 5xx, abra o AWS Service Health Dashboard para verificar se há problemas operacionais no serviço. Você pode usar esse alarme para receber notificações caso haja um problema prolongado de serviço interno do DynamoDB, e isso ajuda você a se correlacionar com o problema que a aplicação cliente está enfrentando. Consulte Tratamento de erros com o DynamoDB para obter mais informações.

Intenção: esse alarme pode detectar erros de sistema contínuos para as solicitações de tabela do DynamoDB. Os erros do sistema indicam erros de serviço interno do DynamoDB e ajudam a correlacionar com o problema que o cliente está enfrentando.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite de acordo com o tráfego esperado, levando em conta um nível aceitável de erros do sistema. Você também pode analisar dados históricos para encontrar

a contagem de erros aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente. Os erros do sistema devem ser testados novamente pela aplicação/serviço, pois são transitórios. Portanto, um limite muito baixo pode fazer com que o alarme seja sensível demais, causando transições de estado indesejadas.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

ThrottledPutRecordCount

Dimensões: TableName, DelegateDoperation

Descrição do alarme: esse alarme detecta os registros que estão tendo controle de utilização pelo fluxo de dados do Kinesis durante a replicação da captura de dados de alteração para o Kinesis. Esse controle de utilização ocorre devido à capacidade insuficiente do fluxo de dados do Kinesis. Se você perceber controle de utilização excessivo e regular, talvez seja necessário aumentar o número de fragmentos de fluxos do Kinesis proporcionalmente ao throughput de gravação observada da tabela. Para saber mais sobre a determinação do tamanho de um fluxo de dados do Kinesis, consulte Como determinar o tamanho inicial de um fluxo de dados do Kinesis.

Intenção: esse alarme pode monitorar o número de registros com controle de utilização pelo fluxo de dados do Kinesis devido à capacidade insuficiente do fluxo de dados do Kinesis.

Estatística: máxima

Limite recomendado: depende da sua situação

Justificativa do limite: é possível que haja algum controle de utilização durante picos de uso excepcionais, mas os registros com controle de utilização devem permanecer o mais baixo possível para evitar maior latência de replicação (o DynamoDB tenta novamente enviar registros com controle de utilização para o fluxo de dados do Kinesis). Defina o limite para um número que possa ajudar você a detectar o controle de utilização excessivo regular. Você também pode analisar os dados históricos dessa métrica para encontrar as taxas de controle de utilização aceitáveis para a workload da aplicação. Ajuste o limite para um valor que a aplicação possa tolerar com base no seu caso de uso.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: GREATER_THAN_THRESHOLD

UserErrors

Dimensões: nenhuma

Descrição do alarme: esse alarme detecta um número alto e contínuo de erros de usuário para as solicitações da tabela do DynamoDB. Você pode verificar os logs da aplicação cliente durante o período do problema para ver por que as solicitações são inválidas. Você pode verificar o código de status HTTP 400 para ver o tipo de erro que você está recebendo e tomar as medidas necessárias. Talvez seja necessário corrigir a lógica da aplicação para criar solicitações válidas.

Intenção: esse alarme pode detectar erros de usuário contínuos para as solicitações da tabela do DynamoDB. Os erros do usuário para operações solicitadas significam que o cliente está produzindo solicitações inválidas e está falhando.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite como zero para detectar quaisquer erros do lado do cliente. Ou você pode defini-lo com um valor mais alto se quiser evitar o acionamento do alarme para um número muito baixo de erros. Decida com base no seu caso de uso e no tráfego das solicitações.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: GREATER_THAN_THRESHOLD

WriteThrottleEvents

Dimensões: TableName

Descrição do alarme: esse alarme detecta se há um grande número de solicitações de gravação passando por um controle de utilização para a tabela do DynamoDB. Consulte Solução de problemas de controle de utilização no Amazon DynamoDB para solucionar o problema.

Intenção: esse alarme pode detectar o controle de utilização contínuo de solicitações de gravação na tabela do DynamoDB. O controle de utilização contínuo das solicitações de gravação pode afetar negativamente as operações de gravação de sua workload e reduzir a eficiência geral do sistema.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite de acordo com o tráfego de gravação esperado para a tabela do DynamoDB, levando em conta um nível aceitável de controle de utilização. É importante monitorar se você está com provisionamento insuficiente e não está causando um controle de utilização consistente. Você também pode analisar os dados históricos para encontrar o nível aceitável de controle de utilização para a workload da aplicação e, em seguida, ajustar o limite para um valor mais alto do que o nível aceitável de controle de utilização habitual. As solicitações de controle de utilização devem ser repetidas pela aplicação/serviço, pois são transitórias. Portanto, um limite muito baixo pode fazer com que o alarme seja sensível demais, causando transições de estado indesejadas.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

WriteThrottleEvents

Dimensões: TableName, GlobalSecondaryIndexName

Descrição do alarme: esse alarme detecta se há um grande número de solicitações de gravação sendo limitadas para o índice secundário global da tabela do DynamoDB. Consulte Solução de problemas de controle de utilização no Amazon DynamoDB para solucionar o problema.

Intenção: esse alarme pode detectar o controle de utilização contínuo de solicitações de gravação para o índice secundário global da tabela do DynamoDB. O controle de utilização contínuo das solicitações de gravação pode afetar negativamente as operações de gravação de sua workload e reduzir a eficiência geral do sistema.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite de acordo com o tráfego de gravação esperado para a tabela do DynamoDB, levando em conta um nível aceitável de controle de utilização. É importante monitorar se você está com provisionamento insuficiente e não está causando um controle de utilização consistente. Você também pode analisar os dados históricos para encontrar o nível de controle de utilização aceitável para a workload da aplicação e, em seguida, ajustar o limite para um valor mais alto do que o nível de controle de utilização aceitável usual. As solicitações de controle de utilização devem ser repetidas pela aplicação/serviço, pois são transitórias. Portanto, um valor muito baixo pode fazer com que o alarme seja sensível demais, o que causa transições de estado indesejadas.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon EBS

VolumeStalledIOCheck

Dimensões: VolumeId, InstanceId

Descrição do alarme: este alarme ajuda você a monitorar o desempenho de E/S dos seus volumes do Amazon EBS. Essa verificação detecta problemas subjacentes com a infraestrutura do Amazon EBS, como problemas de hardware ou software nos subsistemas de armazenamento subjacentes aos volumes do Amazon EBS, problemas de hardware no host físico que afetam a acessibilidade dos volumes do Amazon EBS da sua instância do Amazon EC2 e pode detectar problemas de conectividade entre a instância e os volumes do Amazon EBS. Se a verificação Stalled IO falhar, você poderá esperar a AWS resolver o problema ou pode tomar medidas, como substituir os volumes afetados ou parar e reiniciar a instância à qual o volume está anexado. Na maioria dos casos, quando essa métrica falha, o Amazon EBS diagnostica e recupera automaticamente o volume em alguns minutos.

Intenção: este alarme pode detectar o status dos seus volumes do Amazon EBS para determinar quando esses volumes estão danificados e não conseguem concluir as operações de E/S.

Estatística: máxima

Limite recomendado: 1,0

Justificativa do limite: quando uma verificação de status falha, o valor dessa métrica é 1. O limite é definido de modo que, sempre que a verificação de status falhar, o alarme estará no estado ALARME.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Amazon EC2

CPUUtilization

Dimensões: InstanceId

Descrição do alarme: esse alarme ajuda a monitorar a utilização da CPU de uma instância do EC2. Dependendo da aplicação, níveis de utilização consistentemente altos podem ser normais. Porém, se a performance for prejudicada e a aplicação não for limitada por E/S de disco, memória ou recursos de rede, uma CPU no limite máximo poderá indicar um gargalo de recursos ou problemas de performance da aplicação. A alta utilização da CPU pode indicar que é necessário fazer um upgrade para uma instância que consome mais CPU. Se o monitoramento detalhado estiver ativado, você poderá alterar o período para 60 segundos em vez de 300 segundos. Para obter mais informações, consulte Habilitar ou desabilitar o monitoramento detalhado para instâncias.

Intenção: esse alarme é usado para detectar a alta utilização da CPU.

Estatística: média

Limite recomendado: 80,0

Justificativa do limite: normalmente, é possível definir o limite de utilização da CPU para 70% a 80%. No entanto, você pode ajustar esse valor com base no seu nível de performance aceitável e nas características da workload. Para alguns sistemas, a utilização consistentemente alta da

CPU pode ser normal e não indicar um problema, enquanto para outros pode ser motivo de preocupação. Analise os dados históricos de utilização da CPU para identificar o uso, descubra qual utilização da CPU é aceitável para seu sistema e defina o limite adequadamente.

Período: 300

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: GREATER_THAN_THRESHOLD

StatusCheckFailed

Dimensões: InstanceId

Descrição do alarme: esse alarme ajuda a monitorar as verificações de status do sistema e as verificações de status da instância. Se qualquer um dos tipos de verificação de status falhar, esse alarme deverá estar no estado ALARME.

Intenção: esse alarme é usado para detectar os problemas subjacentes com as instâncias, incluindo falhas na verificação do status do sistema e falhas na verificação do status da instância.

Estatística: máxima

Limite recomendado: 1,0

Justificativa do limite: quando uma verificação de status falha, o valor dessa métrica é 1. O limite é definido de modo que, sempre que a verificação de status falhar, o alarme estará no estado ALARME.

Período: 300

Pontos de dados para o alarme: 2

Períodos de avaliação: 2

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

StatusCheckFailed_AttachedEBS

Dimensões: InstanceId

Descrição do alarme: esse alarme ajuda a monitorar se os volumes do Amazon EBS anexados a uma instância estão acessíveis e são capazes de concluir operações de E/S. Essa verificação de

status detecta problemas subjacentes na infraestrutura do Amazon EBS ou com a computação, por exemplo:

- Problemas de hardware ou software nos subsistemas de armazenamento subjacentes aos volumes do Amazon EBS
- Problemas de hardware no host físico que afetam a acessibilidade dos volumes do Amazon EBS
- Problemas de conectividade entre a instância e os volumes do Amazon EBS

Quando houver uma falha na verificação de status do EBS anexado, você poderá esperar que a Amazon resolva o problema ou adotar medidas por conta própria, p. ex., substituir os volumes afetados ou interromper e reiniciar a instância.

Intenção: esse alarme é usado para detectar volumes inacessíveis do Amazon EBS conectados a uma instância. Isso pode causar falhas nas operações de E/S.

Estatística: máxima

Limite recomendado: 1,0

Justificativa do limite: quando uma verificação de status falha, o valor dessa métrica é 1. O limite é definido de modo que, sempre que a verificação de status falhar, o alarme estará no estado ALARME.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Amazon ElastiCache

CPUUtilization

Dimensões: CacheClusterId, CacheNodeId

Descrição do alarme: esse alarme ajuda a monitorar a utilização da CPU de toda a instância do ElastiCache, incluindo os processos do mecanismo de banco de dados e outros processos

em execução na instância. O AWS Elasticache é compatível com dois tipos de mecanismo: Memcached e Redis. Quando atingir uma alta utilização da CPU em um nó do Memcached, considere aumentar a escala verticalmente do tipo de instância ou adicionar novos nós de cache. Para o Redis, se sua workload principal for de solicitações de leitura, você deve considerar adicionar mais réplicas de leitura ao cluster de cache. Se a sua workload principal for de solicitações de gravação, considere adicionar mais fragmentos para distribuir a workload em mais nós primários, caso esteja executando no modo em cluster, ou aumentar a escala verticalmente do seu tipo de instância, caso esteja executando o Redis no modo sem cluster.

Intenção: esse alarme é usado para detectar a alta utilização da CPU dos hosts do ElastiCache. É útil obter uma visão ampla do uso da CPU em toda a instância, incluindo processos que não são do mecanismo.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite como a porcentagem que reflete um nível crítico de utilização da CPU para sua aplicação. Para o Memcached, o mecanismo pode usar até núcleos num_threads. Para o Redis, o mecanismo é basicamente de thread único, mas pode usar núcleos adicionais, se disponíveis, para acelerar a E/S. Na maioria dos casos, você pode definir o limite para cerca de 90% da CPU disponível. Como o Redis é de thread único, o valor limite real deve ser calculado como uma fração da capacidade total do nó.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

CurrConnections

Dimensões: CacheClusterId, CacheNodeId

Descrição do alarme: esse alarme detecta uma alta contagem de conexões, o que pode indicar uma carga pesada ou problemas de performance. Um aumento constante de `CurrConnections` pode levar ao esgotamento das 65 mil conexões disponíveis. Isso pode indicar que as conexões foram fechadas incorretamente no lado da aplicação e permaneceram estabelecidas no lado do servidor. É preciso considerar o uso de pooling de conexões ou tempos limite de conexão ociosa

para limitar o número de conexões feitas ao cluster ou, no caso do Redis, considerar o ajuste do tcp-keepalive em seu cluster para detectar e encerrar possíveis pares mortos.

Intenção: o alarme ajuda a identificar altas contagens de conexões que podem afetar a performance e a estabilidade do cluster do ElastiCache.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme depende muito do intervalo aceitável de conexões para o seu cluster. Revise a capacidade e a workload esperada de seu cluster do ElastiCache e analise as contagens históricas de conexões durante o uso regular para estabelecer uma linha de base e, em seguida, selecione um limite adequadamente. Lembre-se de que cada nó comporta até 65 mil conexões simultâneas.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: GREATER_THAN_THRESHOLD

DatabaseMemoryUsagePercentage

Dimensões: CacheClusterId

Descrição do alarme: esse alarme ajuda você a monitorar a utilização da memória do seu cluster. Quando `DatabaseMemoryUsagePercentage` atinge 100%, a política de memória máxima do Redis é acionada e podem ocorrer remoções com base na política selecionada. Se nenhum objeto no cache corresponder à política de remoção, as operações de gravação falharão. Algumas workloads esperam ou dependem de remoções, mas, se não for o caso, você precisará aumentar a capacidade de memória do seu cluster. Você pode escalar verticalmente seu cluster adicionando mais nós primários, ou aumentá-lo usando um tipo de nó maior. Consulte Escalabilidade de clusters do ElastiCache for Redis para obter detalhes.

Intenção: esse alarme é usado para detectar a alta utilização de memória do seu cluster para que você possa evitar falhas ao gravar no cluster. É útil saber quando você precisará aumentar a escala verticalmente do seu cluster caso sua aplicação não espere passar por remoções.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: dependendo dos requisitos de memória da sua aplicação e da capacidade de memória do seu cluster do ElastiCache, você deve definir o limite para a porcentagem que reflete o nível crítico de uso de memória do cluster. Você pode usar dados históricos de uso de memória como referência para o limite aceitável de uso de memória.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

EngineCPUUtilization

Dimensões: CacheClusterId

Descrição do alarme: esse alarme ajuda a monitorar a utilização da CPU de um thread do mecanismo Redis dentro da instância do ElastiCache. Os motivos comuns para o alto nível de CPU do mecanismo são comandos de longa execução que consomem muita CPU, um alto número de solicitações, um aumento de novas solicitações de conexão de cliente em um curto período de tempo e grandes remoções quando o cache não tem memória suficiente para armazenar novos dados. Você deve considerar a Escalabilidade de clusters do ElastiCache for Redis adicionando mais nós ou aumentando a escala verticalmente do seu tipo de instância.

Intenção: esse alarme é usado para detectar a alta utilização da CPU do thread do mecanismo Redis. Ele é útil caso deseje monitorar o uso da CPU do próprio mecanismo de banco de dados.

Estatística: média

Limite recomendado: 90,0

Justificativa do limite: defina o limite como uma porcentagem que reflita o nível crítico de utilização da CPU do mecanismo para a sua aplicação. Você pode fazer um benchmark do seu cluster usando sua aplicação e a workload esperada para correlacionar a utilização do EngineCPUUtilization e a performance como referência e, em seguida, definir o limite adequadamente. Na maioria dos casos, você pode definir o limite para cerca de 90% da CPU disponível.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

ReplicationLag

Dimensões: CacheClusterId

Descrição do alarme: esse alarme ajuda a monitorar a integridade da replicação do cluster do ElastiCache. Um alto atraso de replicação significa que o nó primário ou a réplica não consegue manter o ritmo da replicação. Se a atividade de gravação for muito alta, considere escalar o cluster adicionando mais nós primários ou ampliando-o usando um tipo de nó maior. Consulte Escalabilidade de clusters do ElastiCache for Redis para obter detalhes. Se suas réplicas de leitura estiverem sobrecarregadas pela quantidade de solicitações de leitura, considere adicionar mais réplicas de leitura.

Intenção: esse alarme é usado para detectar um atraso entre as atualizações de dados no nó primário e sua sincronização com o nó de réplica. Ele ajuda a garantir a consistência dos dados de um nó de cluster de réplica de leitura.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite de acordo com os requisitos de sua aplicação e o possível impacto do atraso da replicação. Você deve considerar as taxas de gravação esperadas da sua aplicação e as condições de rede para o atraso de replicação aceitável.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon EC2 (**`AWS/ElasticGPUs`**)

GPUConnectivityCheckFailed

Dimensões: InstanceId, EGPUId

Descrição do alarme: esse alarme ajuda a detectar falhas de conexão entre a instância e o acelerador do Elastic Graphics. O Elastic Graphics usa a rede de instâncias para enviar comandos OpenGL a uma placa gráfica remotamente anexada. Além disso, uma área de trabalho que executa uma aplicação OpenGL com um acelerador do Elastic Graphics geralmente é acessada usando a tecnologia de acesso remoto. É importante distinguir entre um problema de performance relativo à renderização do OpenGL ou à tecnologia de acesso remoto da área de trabalho. Para saber mais sobre o problema, consulte Investigar problemas na performance da aplicação.

Intenção: esse alarme é usado para detectar problemas de conectividade da instância com o acelerador do Elastic Graphics.

Estatística: máxima

Limite recomendado: 0,0

Justificativa do limite: o valor limite de 1 indica que a conectividade falhou.

Período: 300

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: GREATER_THAN_THRESHOLD

GPUHealthCheckFailed

Dimensões: InstanceId, EGPUId

Descrição do alarme: esse alarme ajuda você a saber quando o status do acelerador gráfico do Elastic não está íntegro. Se o acelerador não estiver íntegro, consulte as etapas de solução de problemas em Resolver problemas de status não íntegros.

Intenção: esse alarme é usado para detectar se o acelerador do Elastic Graphics não está íntegro.

Estatística: máxima

Limite recomendado: 0,0

Justificativa do limite: o valor limite de 1 indica uma falha na verificação de status.

Período: 300

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon ECS

CPUReservation

Dimensões: ClusterName

Descrição do alarme: esse alarme ajuda a detectar uma alta reserva de CPU no cluster do ECS. A alta reserva de CPU pode indicar que o cluster está ficando sem CPUs registradas para a tarefa. Para solucionar problemas, você pode adicionar mais capacidade, escalar o cluster ou configurar o ajuste de escala automático.

Intenção: o alarme é usado para detectar se o número total de unidades de CPU reservadas por tarefas no cluster está atingindo o total de unidades de CPU registradas para o cluster. Isso ajuda você a saber quando aumentar a escala do cluster verticalmente. Alcançar o total de unidades de CPU do cluster pode resultar na falta de CPU para tarefas. Se o escalonamento gerenciado dos provedores de capacidade do EC2 estiver ativado ou se você tiver associado o Fargate a provedores de capacidade, esse alarme não é recomendado.

Estatística: média

Limite recomendado: 90,0

Justificativa do limite: defina o limite para reserva de CPU em 90%. Como alternativa, você pode escolher um valor mais baixo com base nas características do cluster.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

CPUUtilization

Dimensões: ClusterName, ServiceName

Descrição do alarme: esse alarme ajuda você a detectar uma alta utilização da CPU do serviço do ECS. Se não houver nenhuma implantação de ECS em andamento, a utilização máxima da CPU poderá indicar um gargalo de recursos ou problemas de performance da aplicação. Para solucionar o problema, você pode aumentar o limite da CPU.

Intenção: esse alarme é usado para detectar a alta utilização da CPU no serviço do ECS. A alta utilização consistente da CPU pode indicar um gargalo de recursos ou problemas de performance da aplicação.

Estatística: média

Limite recomendado: 90,0

Justificativa do limite: as métricas de serviço para utilização da CPU podem exceder 100% de utilização. No entanto, recomendamos que você monitore a métrica quanto à alta utilização da CPU para evitar o impacto em outros serviços. Defina o limite para cerca de 90% a 95%. Recomendamos que você atualize suas definições de tarefas para refletir o uso real a fim de evitar problemas futuros com outros serviços.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

MemoryReservation

Dimensões: ClusterName

Descrição do alarme: esse alarme ajuda você a detectar uma reserva de memória alta no cluster do ECS. A alta reserva de memória pode indicar um gargalo de recursos para o cluster. Para solucionar problemas, analise a performance da tarefa de serviço para ver se a utilização da memória da tarefa pode ser otimizada. Além disso, é possível registrar mais memória ou configurar o ajuste de escala automático.

Intenção: o alarme é usado para detectar se o total de unidades de memória reservadas por tarefas no cluster está atingindo o total de unidades de memória registradas para o cluster. Isso

pode ajudar você a saber quando aumentar a escala verticalmente do cluster. Atingir o total de unidades de memória do cluster pode fazer com que o cluster não consiga iniciar novas tarefas. Se você tiver o escalonamento gerenciado dos provedores de capacidade do EC2 ativado ou se tiver associado o Fargate a provedores de capacidade, esse alarme não é recomendado.

Estatística: média

Limite recomendado: 90,0

Justificativa do limite: defina o limite para reserva de memória em 90%. Você pode ajustar isso para um valor mais baixo com base nas características do cluster.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

HTTPCode_Target_5XX_Count

Dimensões: ClusterName, ServiceName

Descrição do alarme: esse alarme ajuda você a detectar uma alta contagem de erros no lado do servidor para o serviço do ECS. Isso pode indicar que há erros que fazem com que o servidor não consiga atender às solicitações. Para solucionar o problema, verifique os logs da aplicação.

Intenção: esse alarme é usado para detectar uma alta contagem de erros no lado do servidor para o serviço do ECS.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: calcule o valor de cerca de 5% de seu tráfego médio e use esse valor como ponto de partida para o limite. Você pode encontrar o tráfego médio usando a métrica `RequestCount`. Você também pode analisar dados históricos para determinar a taxa de erro aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente. Os erros 5XX que ocorrem com frequência precisam receber um alarme. No entanto, definir um valor muito baixo para o limite pode fazer com que o alarme seja muito sensível.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

TargetResponseTime

Dimensões: ClusterName, ServiceName

Descrição do alarme: esse alarme ajuda você a detectar um tempo de resposta alvo alto para solicitações de serviço do ECS. Isso pode indicar que há problemas que fazem com que o serviço não consiga atender às solicitações a tempo. Para solucionar problemas, verifique a métrica CPUUtilization para verificar se o serviço está ficando sem CPU ou verifique a utilização da CPU de outros serviços downstream dos quais seu serviço depende.

Intenção: esse alarme é usado para detectar um tempo de resposta alvo alto para solicitações de serviço do ECS.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme depende muito do seu caso de uso. Analise a criticidade e os requisitos do tempo alvo de resposta do serviço e analise o comportamento histórico dessa métrica para determinar níveis de limite adequados.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon ECS com o Container Insights

EphemeralStorageUtilized

Dimensões: ClusterName, ServiceName

Descrição do alarme: este alarme ajuda a detectar a alta utilização do armazenamento temporário no cluster do Fargate. Se o uso do armazenamento temporário for consistentemente alto, você poderá verificá-lo e aumentá-lo.

Intenção: este alarme será usado para detectar a alta utilização do armazenamento temporário para o cluster do Fargate. A alta utilização de um armazenamento temporário de forma consistente pode indicar que o disco está cheio e pode resultar em falhas do contêiner.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite como cerca de 90% do tamanho do armazenamento temporário. É possível ajustar esse valor com base na utilização aceitável do armazenamento temporário do cluster do Fargate. Para alguns sistemas, a alta utilização de um armazenamento temporário de forma consistente pode ser normal, enquanto para outros, pode resultar em falhas do contêiner.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

RunningTaskCount

Dimensões: ClusterName, ServiceName

Descrição do alarme: este alarme ajuda a detectar uma baixa contagem de tarefas em execução do serviço ECS. Caso a contagem de tarefas em execução seja muito baixa, isso pode indicar que a aplicação não consegue lidar com a carga de serviço e pode resultar em problemas de performance. Caso não haja tarefas em execução, o serviço Amazon ECS pode estar indisponível ou pode haver problemas de implantação.

Intenção: este alarme é usado para detectar se o número de tarefas em execução está muito baixo. Uma baixa contagem de tarefas em execução de forma consistente pode indicar problemas de implantação ou de performance do serviço ECS.

Estatística: média

Limite recomendado: 0,0

Justificativa do limite: é possível ajustar o limite com base na contagem mínima de tarefas em execução do serviço ECS. Se a contagem de tarefas em execução for 0, o serviço do Amazon ECS estará indisponível.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: LESS_THAN_OR_EQUAL_TO_THRESHOLD

instance_filesystem_utilization

Dimensões: InstanceId, ContainerInstanceId e ClusterName

Descrição do alarme: este alarme ajuda a detectar uma alta utilização do sistema de arquivos do cluster do ECS. Se a utilização do sistema de arquivos for consistentemente alta, verifique o uso do disco.

Intenção: este alarme é usado para detectar uma alta utilização do sistema de arquivos para o cluster do Amazon ECS. Uma alta utilização do sistema de arquivos de forma consistente pode indicar um gargalo de recursos ou problemas de performance da aplicação e pode impedir a execução de novas tarefas.

Estatística: média

Limite recomendado: 90,0

Justificativa do limite: é possível definir o limite de utilização do sistema de arquivos para cerca de 90 a 95%. Você pode ajustar esse valor com base no nível aceitável de capacidade para o sistema de arquivos do cluster do Amazon ECS. Para alguns sistemas, uma alta utilização do sistema de arquivos de forma consistente pode ser normal e não indicar problemas, enquanto que para outros, pode ser um motivo de preocupação e pode resultar em problemas de performance e impedir a execução de novas tarefas.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon EFS

PercentIOLimit

Dimensões: FileSystemID

Descrição do alarme: esse alarme ajuda a garantir que a workload permaneça dentro do limite de E/S disponível para o sistema de arquivos. Se a métrica atingir o limite de E/S de forma consistente, considere transferir a aplicação para um sistema de arquivos que use a performance máxima de E/S como modo. Para solucionar o problema, verifique os clientes que estão conectados ao sistema de arquivos e as aplicações dos clientes que controlam a utilização do sistema de arquivos.

Intenção: esse alarme é usado para detectar o quanto o sistema de arquivos está próximo de atingir o limite de E/S do modo de performance de uso geral. Uma porcentagem consistentemente alta de E/S pode ser um indicador de que o sistema de arquivos não pode ser escalonado com relação a solicitações de E/S suficientes, e o sistema de arquivos pode ser um gargalo de recursos para as aplicações que usam o sistema de arquivos.

Estatística: média

Limite recomendado: 100,0

Justificativa do limite: quando o sistema de arquivos atinge seu limite de E/S, ele pode responder mais lentamente às solicitações de leitura e gravação. Portanto, é recomendável que a métrica seja monitorada para evitar o impacto nas aplicações que usam o sistema de arquivos. O limite pode ser definido em torno de 100%. No entanto, esse valor pode ser ajustado para um valor menor com base nas características do sistema de arquivos.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

BurstCreditBalance

Dimensões: FileSystemID

Descrição do alarme: esse alarme ajuda a garantir que haja um saldo de crédito de estouro disponível para o uso do sistema de arquivos. Quando não houver crédito de burst disponível, o acesso das aplicações ao sistema de arquivos será limitado devido ao baixo throughput. Se a métrica cair para 0 de forma consistente, considere alterar o modo de throughput para o modo de throughput Elástico ou Provisionado.

Intenção: esse alarme é usado para detectar o baixo saldo de crédito de burst do sistema de arquivos. Um saldo de crédito de burst baixo e consistente pode ser um indicador da diminuição do throughput e do aumento da latência de E/S.

Estatística: média

Limite recomendado: 0,0

Justificativa do limite: quando o sistema de arquivos fica sem créditos de burst e mesmo que o throughput de linha de base seja menor, o EFS continua a fornecer um throughput medido de 1 MiBps para todos os sistemas de arquivos. No entanto, recomenda-se que a métrica seja monitorada quanto ao baixo saldo de crédito de burst para evitar que o sistema de arquivos atue como gargalo de recursos para as aplicações. O limite pode ser definido em torno de 0 bytes.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: LESS_THAN_OR_EQUAL_TO_THRESHOLD

Amazon EKS com o Container Insights

node_cpu_utilization

Dimensões: ClusterName

Descrição do alarme: esse alarme ajuda a detectar a alta utilização da CPU nos nós de processamento do cluster do EKS. Se a utilização for consistentemente alta, isso pode indicar a necessidade de substituir os nós de processamento por instâncias que tenham mais CPU ou a necessidade de escalar o sistema horizontalmente.

Intenção: esse alarme ajuda a monitorar a utilização da CPU dos nós de processamento no cluster do EKS para que a performance do sistema não diminua.

Estatística: máxima

Limite recomendado: 80,0

Justificativa do limite: recomenda-se definir o limite como menor ou igual a 80% para permitir tempo suficiente para depurar o problema antes que o sistema comece a sofrer impacto.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

node_filesystem_utilization

Dimensões: ClusterName

Descrição do alarme: esse alarme ajuda a detectar a alta utilização do sistema de arquivos nos nós de processamento do cluster do EKS. Se a utilização for consistentemente alta, talvez seja necessário atualizar os nós de processamento para que tenham um volume de disco maior, ou talvez seja necessário escalar horizontalmente.

Descrição do alarme: esse alarme ajuda a monitorar a utilização do sistema de arquivos dos nós de processamento no cluster do EKS. Se a utilização atingir 100%, isso pode levar à falha da aplicação, a gargalos de E/S do disco, à remoção de pods ou ao fato de o nó deixar de responder completamente.

Estatística: máxima

Limite recomendado: depende da sua situação

Justificativa do limite: se houver pressão suficiente no disco (o que significa que o disco está ficando cheio), os nós serão marcados como não íntegros e os pods serão removidos do nó. Os pods em um nó com pressão de disco são removidos quando o sistema de arquivos disponível é menor do que os limites de remoção definidos no kubelet. Defina o limite do alarme para que você tenha tempo suficiente para reagir antes que o nó seja removido do cluster.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

node_memory_utilization

Dimensões: ClusterName

Descrição do alarme: esse alarme ajuda a detectar a alta utilização de memória nos nós de processamento do cluster do EKS. Se a utilização for consistentemente alta, isso pode indicar a necessidade de escalar o número de réplicas de pod ou otimizar a sua aplicação.

Intenção: esse alarme ajuda a monitorar a utilização da memória dos nós de processamento no cluster do EKS para que a performance do sistema não diminua.

Estatística: máxima

Limite recomendado: 80,0

Justificativa do limite: recomenda-se definir o limite como menor ou igual a 80% para permitir que se tenha tempo suficiente para depurar o problema antes que o sistema comece a sofrer impacto.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

pod_cpu_utilization_over_pod_limit

Dimensões: ClusterName, Namespace, serviço

Descrição do alarme: esse alarme ajuda a detectar a alta utilização da CPU em pods do cluster do EKS. Se a utilização for consistentemente alta, isso poderá indicar a necessidade de aumentar o limite da CPU para o pod afetado.

Intenção: esse alarme ajuda a monitorar a utilização da CPU dos pods pertencentes a um Serviço do Kubernetes no cluster do EKS para que você possa identificar rapidamente se o pod de um serviço está consumindo mais CPU do que o esperado.

Estatística: máxima

Limite recomendado: 80,0

Justificativa do limite: recomenda-se definir o limite como menor ou igual a 80% para permitir que se tenha tempo suficiente para depurar o problema antes que o sistema comece a sofrer impacto.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

pod_memory_utilization_over_pod_limit

Dimensões: ClusterName, Namespace, serviço

Descrição do alarme: esse alarme ajuda a detectar a alta utilização de memória em pods do cluster do EKS. Se a utilização for consistentemente alta, isso poderá indicar a necessidade de aumentar o limite de memória para o pod afetado.

Intenção: esse alarme ajuda a monitorar a utilização da memória dos pods no cluster do EKS para que a performance do sistema não diminua.

Estatística: máxima

Limite recomendado: 80,0

Justificativa do limite: recomenda-se definir o limite como menor ou igual a 80% para permitir que se tenha tempo suficiente para depurar o problema antes que o sistema comece a sofrer impacto.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon Kinesis Data Streams

GetRecords.IteratorAgeMilliseconds

Dimensões: StreamName

Descrição do alarme: esse alarme pode detectar se a idade máxima do iterador é muito alta. Para aplicações de processamento de dados em tempo real, configure a retenção de dados de acordo com a tolerância do atraso. Isso geralmente ocorre em minutos. Para aplicações que processam dados históricos, use essa métrica para monitorar a velocidade de recuperação. Uma solução rápida para impedir a perda de dados é aumentar o período de retenção enquanto você soluciona o problema. Também é possível aumentar o número de trabalhadores que processam registros em sua aplicação do consumidor. As causas mais comuns para o aumento gradual da idade do iterador são recursos físicos insuficientes ou lógica de processamento de registros que não foi escalonada com um aumento no throughput do fluxo. Consulte o link para obter mais detalhes.

Intenção: esse alarme é usado para detectar se os dados no seu fluxo vão expirar por terem sido preservados por muito tempo ou porque o processamento de registros está muito lento. Ele ajuda a evitar a perda de dados após atingir 100% do tempo de retenção do fluxo.

Estatística: máxima

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme depende muito do período de retenção do fluxo e da tolerância de atraso no processamento dos registros. Revise seus requisitos e analise as tendências históricas e, em seguida, defina o limite para o número de milissegundos que representa um atraso crítico no processamento. Se a idade de um iterador ultrapassar 50% do período de retenção (por padrão, 24 horas, configurável até 365 dias), haverá um risco de perda de dados devido à expiração do registro. Você pode monitorar a métrica para garantir que nenhum dos seus fragmentos se aproxime desse limite.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

GetRecords.Success

Dimensões: StreamName

Descrição do alarme: essa métrica é incrementada sempre que os consumidores leem com êxito os dados do fluxo. O `GetRecords` não retorna nenhum dado quando lança uma exceção. A exceção mais comum é `ProvisionedThroughputExceededException`, pois a taxa

de solicitação do fluxo é muito alta ou porque o throughput disponível já foi atendido para o segundo em questão. Reduzir a frequência ou o tamanho de suas solicitações. Para obter mais informações, consulte Limites de fluxos no Guia do desenvolvedor do Amazon Kinesis Data Streams e Repetições de erro e recuo exponencial na AWS.

Intenção: esse alarme pode detectar se a recuperação de registros do fluxo pelos consumidores está falhando. Ao definir um alarme para essa métrica, você pode detectar proativamente qualquer problema com o consumo de dados, como o aumento das taxas de erro ou a diminuição das recuperações bem-sucedidas. Isso permite que você tome medidas oportunas para resolver possíveis problemas e mantenha um pipeline de processamento de dados regular.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: dependendo da importância da recuperação de registros do fluxo, defina o limite com base na tolerância da sua aplicação para registros com falha. O limite deve ser a porcentagem correspondente de operações bem-sucedidas. Você pode usar dados históricos da métrica GetRecords como referência para a taxa de falha aceitável. Também é preciso considerar as novas tentativas ao definir o limite, pois os registros com falha podem ser tentados novamente. Isso ajuda a evitar que picos transitórios acionem alertas desnecessários.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: LESS_THAN_THRESHOLD

PutRecord.Success

Dimensões: StreamName

Descrição do alarme: esse alarme detecta quando o número de operações de `PutRecord` com falha ultrapassa o limite. Investigue os logs do produtor de dados para encontrar as causas principais das falhas. O motivo mais comum é o throughput provisionado insuficiente no fragmento que causou o `ProvisionedThroughputExceededException`. Isso acontece porque a taxa de solicitação do fluxo é muito alta ou o throughput tentado para ser ingerido no fragmento é muito alto. Reduzir a frequência ou o tamanho de suas solicitações. Para obter mais informações, consulte Limites de fluxos e Repetições de erro e recuo exponencial na AWS.

Intenção: esse alarme pode detectar se a ingestão de registros no fluxo está falhando. Ele ajuda a identificar problemas na gravação de dados no fluxo. Ao definir um alarme para essa métrica, é possível detectar proativamente quaisquer problemas de produtores na publicação de dados no fluxo, como aumento das taxas de erro ou diminuição dos registros publicados com êxito. Isso permite que você tome medidas oportunas para resolver possíveis problemas e mantenha um processo de ingestão de dados confiável.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: dependendo da importância da ingestão de dados e do processamento para o seu serviço, defina o limite com base na tolerância da sua aplicação para registros com falha. O limite deve ser a porcentagem correspondente de operações bem-sucedidas. É possível usar dados históricos da métrica PutRecord como referência para a taxa de falha aceitável. Também é preciso considerar as novas tentativas ao definir o limite, pois os registros com falha podem ser tentados novamente.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: LESS_THAN_THRESHOLD

PutRecords.FailedRecords

Dimensões: StreamName

Descrição do alarme: esse alarme detecta quando o número de operações `PutRecords` com falha excede o limite. O fluxo de dados do Kinesis tenta processar todos os registros em cada solicitação `PutRecords`, mas uma única falha de registro não interrompe o processamento dos registros subsequentes. O principal motivo dessas falhas é exceder o throughput de um fluxo ou de um fragmento individual. As causas comuns são picos de tráfego e latências de rede que fazem com que os registros cheguem ao fluxo de forma desigual. Você deve detectar registros processados sem sucesso e tentar novamente em uma chamada subsequente. Consulte Tratamento de falhas ao usar PutRecords para obter mais detalhes.

Intenção: esse alarme pode detectar falhas consistentes ao usar a operação em lote para colocar registros no seu fluxo. Ao definir um alarme para essa métrica, é possível detectar proativamente um aumento no número de registros com falha, o que permite tomar medidas oportunas para

resolver os problemas subjacentes e garantir um processo de ingestão de dados regular e confiável.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite para o número de registros com falha que reflete a tolerância da aplicação para registros com falha. Você pode usar dados históricos como referência para o valor de falha aceitável. Também é preciso considerar as novas tentativas ao definir o limite, pois os registros com falha podem ser tentados novamente em chamadas PutRecords subsequentes.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

ReadProvisionedThroughputExceeded

Dimensões: StreamName

Descrição do alarme: o alarme rastreia o número de registros que resultam no controle de utilização da capacidade de throughput de leitura. Se você perceber que o controle de utilização está sendo constante, considere adicionar mais fragmentos ao seu fluxo para aumentar o throughput de leitura provisionado. Se houver mais de uma aplicação de consumo em execução no fluxo e elas compartilharem o limite do `GetRecords`, recomendamos que você registre novas aplicações de consumo via distribuição aprimorada. Se a adição de mais fragmentos não reduzir o número de controles de utilização, é possível que haja um fragmento "quente" que esteja sendo lido mais do que os outros fragmentos. Ative a distribuição aprimorada, localize o fragmento "quente" e divida-o.

Intenção: esse alarme pode detectar se os consumidores passam por um controle de utilização quando excedem o throughput de leitura provisionado (determinado pelo número de fragmentos que você tem). Nesse caso, não será possível fazer a leitura do fluxo, e o fluxo pode começar a fazer o backup.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: normalmente, as solicitações com controle de utilização podem ser repetidas e, portanto, definir o limite como zero torna o alarme muito sensível. No entanto, o controle de utilização consistente pode afetar a leitura do fluxo e deve acionar o alarme. Defina o limite como uma porcentagem de acordo com as solicitações com controle de utilização para as configurações da aplicação e de repetição.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

SubscribeToShardEvent.MillisBehindLatest

Dimensões: StreamName, ConsumerName

Descrição do alarme: esse alarme detecta quando o atraso do processamento de registros na aplicação ultrapassa o limite. Problemas transitórios, como falhas na operação da API em uma aplicação downstream, podem causar um aumento repentino na métrica. É necessário investigar se isso ocorre de forma consistente. Uma causa comum é que o consumidor não está processando registros com rapidez suficiente devido a recursos físicos insuficientes ou à lógica de processamento de registros que não foi escalonada com um aumento no throughput do fluxo. O bloqueio de chamadas no caminho crítico geralmente é a causa de lentidão no processamento de registros. Você pode aumentar o paralelismo aumentando o número de fragmentos. Você também deve confirmar se os nós de processamento subjacentes têm recursos físicos suficientes durante o pico de demanda.

Intenção: esse alarme pode detectar atraso na assinatura do evento de fragmento do fluxo. Isso indica um atraso no processamento e pode ajudar a identificar possíveis problemas com a performance da aplicação do consumidor ou com a integridade geral do fluxo. Quando o atraso no processamento torna-se significativo, você deve investigar e resolver quaisquer gargalos ou ineficiências das aplicações do consumidor para garantir o processamento de dados em tempo real e minimizar o acúmulo de dados.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme depende muito do atraso que sua aplicação pode tolerar. Analise os requisitos da sua aplicação e analise

as tendências históricas e, em seguida, selecione um limite adequadamente. Quando a chamada SubscribeToShard for bem-sucedida, o consumidor começará a receber eventos SubscribeToShardEvent pela conexão persistente por até cinco minutos, após os quais será necessário chamar o SubscribeToShard novamente para renovar a assinatura se quiser continuar a receber registros.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

WriteProvisionedThroughputExceeded

Dimensões: StreamName

Descrição do alarme: esse alarme detecta quando o número de registros que resultam no controle de utilização da capacidade de throughput de gravação atingiu o limite. Quando seus produtores excedem o throughput de gravação provisionado (determinado pelo número de fragmentos que você tem), eles passam por um controle de utilização e você não poderá colocar registros no fluxo. Para resolver o controle de utilização consistente, você deve considerar a adição de fragmentos ao seu fluxo. Isso aumenta seu throughput de gravação provisionado e evita o controle de utilização no futuro. Você também deve considerar a escolha da chave de partição ao ingerir registros. A chave de partição aleatória é preferível porque distribui os registros uniformemente entre os fragmentos do fluxo, sempre que possível.

Intenção: esse alarme pode detectar se os seus produtores estão sendo rejeitados para gravar registros devido ao controle de utilização do fluxo ou do fragmento. Se o seu fluxo estiver no modo Provisionado, a configuração desse alarme ajudará você a tomar medidas proativas quando o fluxo de dados atingir seus limites, permitindo otimizar a capacidade provisionada ou tomar medidas de escalonamento adequadas para evitar a perda de dados e manter o processamento de dados regular.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: normalmente, as solicitações com controle de utilização podem ser repetidas, portanto, definir o limite como zero torna o alarme muito sensível. No entanto, o

controle de utilização consistente pode afetar a gravação no fluxo, e você deve definir o limite de alarme para detectar isso. Defina o limite como uma porcentagem de acordo com as solicitações com controle de utilização para as configurações da aplicação e de repetição.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Lambda

ClaimedAccountConcurrency

Dimensões: nenhuma

Descrição do alarme: esse alarme ajuda a monitorar se a simultaneidade de suas funções do Lambda está se aproximando do limite de simultaneidade por região da sua conta. Uma função começa a passar por um controle de utilização ao atingir o limite de simultaneidade. Você pode realizar as ações a seguir para evitar o controle de utilização.

1. Solicitar um aumento de simultaneidade nesta região.
2. Identificar e reduzir qualquer simultaneidade reservada ou simultaneidade provisionada não utilizada.
3. Identificar problemas de desempenho nas funções para aumentar a velocidade de processamento e, portanto, melhorar o throughput.
4. Aumentar o tamanho de lote das funções para que cada invocação de função processe mais mensagens.

Intenção: esse alarme pode detectar proativamente se a simultaneidade de suas funções do Lamba está se aproximando da cota de simultaneidade no nível de região da sua conta para que você possa agir. As funções passam por um controle de utilização se `ClaimedAccountConcurrency` atingir a cota de simultaneidade no nível de região da conta. Se você estiver usando Simultaneidade reservada (RC) ou Simultaneidade provisionada (PC), esse alarme proporcionará mais visibilidade sobre a utilização da simultaneidade em comparação a um alarme em `ConcurrentExecutions`.

Estatística: máxima

Limite recomendado: depende da sua situação

Justificativa do limite: você deve calcular o valor de aproximadamente 90% da cota de simultaneidade definida para a conta na região e usar o resultado como valor limite. Por padrão, sua conta tem uma cota de simultaneidade de mil em todas as funções de uma região. No entanto, verifique a cota da sua conta no painel do Service Quotas.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: GREATER_THAN_THRESHOLD

Erros

Dimensões: FunctionName

Descrição do alarme: esse alarme detecta altas contagens de erros. Os erros incluem as exceções lançadas pelo código, bem como as exceções lançadas pelo runtime do Lambda. Você pode verificar os logs relacionados à função para diagnosticar o problema.

Intenção: o alarme ajuda a detectar altas contagens de erros em invocações de funções.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite como um número maior que zero. O valor exato pode depender da tolerância a erros em sua aplicação. Entenda a criticidade das invocações que a função está tratando. Para algumas aplicações, qualquer erro pode ser inaceitável, enquanto outras aplicações podem permitir uma certa margem de erro.

Período: 60

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: GREATER_THAN_THRESHOLD

Controles de utilização

Dimensões: FunctionName

Descrição do alarme: esse alarme detecta um alto número de solicitações de invocação com controle de utilização. O controle de utilização ocorre quando não há simultaneidade disponível para aumentar a escala verticalmente. Há várias abordagens para resolver esse problema. 1) Solicitar um aumento de simultaneidade do AWS Suporte nesta região. 2) Identificar problemas de performance na função para aumentar a velocidade de processamento e, portanto, melhorar o throughput. 3) Aumentar o tamanho do lote da função para que mais mensagens sejam processadas por cada invocação de função.

Intenção: o alarme ajuda a detectar um alto número de solicitações de invocação com controle de utilização para uma função do Lambda. É importante saber se as solicitações estão sendo constantemente rejeitadas devido ao controle de utilização e se você precisa melhorar a performance da função do Lambda ou aumentar a capacidade de simultaneidade para evitar o controle de utilização constante.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite como um número maior que zero. O valor exato do limite pode depender da tolerância da aplicação. Defina o limite de acordo com seu uso e os requisitos de escalonamento da função.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Duration (Duração)

Dimensões: FunctionName

Descrição do alarme: esse alarme detecta tempos de longa duração para o processamento de um evento por uma função do Lambda. As longas durações podem ser causadas por alterações no código da função, fazendo com que a função demore mais para ser executada, ou que as dependências da função demorem mais.

Intenção: esse alarme pode detectar uma longa duração de funcionamento de uma função do Lambda. A alta duração do runtime indica que uma função está levando mais tempo para ser invocada e também pode afetar a capacidade de simultaneidade da invocação se o Lambda

estiver lidando com um número maior de eventos. É fundamental saber se a função do Lambda está constantemente levando mais tempo de execução do que o esperado.

Estatística: p90

Limite recomendado: depende da sua situação

Justificativa do limite: o limite para a duração depende de sua aplicação, das workloads e de seus requisitos de performance. Para requisitos de alta performance, defina o limite para um tempo mais curto para verificar se a função está atendendo às expectativas. Você também pode analisar dados históricos de métricas de duração para ver se o tempo gasto corresponde à expectativa de performance da função e, em seguida, definir o limite para um tempo maior do que a média histórica. Certifique-se de definir o limite inferior ao tempo limite da função configurada.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

ConcurrentExecutions

Dimensões: FunctionName

Descrição do alarme: esse alarme ajuda a monitorar se a simultaneidade da função está se aproximando do limite de simultaneidade no nível de região da sua conta. Uma função começa a passar por um controle de utilização ao atingir o limite de simultaneidade. Você pode realizar as ações a seguir para evitar o controle de utilização.

1. Solicitar um aumento de simultaneidade nesta região.
2. Identificar problemas de desempenho nas funções para aumentar a velocidade de processamento e, portanto, melhorar o throughput.
3. Aumentar o tamanho de lote das funções para que cada invocação de função processe mais mensagens.

Para obter melhor visibilidade da utilização da simultaneidade reservada e da simultaneidade provisionada, defina um alarme para a nova métrica `ClaimedAccountConcurrency`.

Intenção: esse alarme pode detectar proativamente se a simultaneidade da função está se aproximando da cota de simultaneidade no nível de região da sua conta para que você possa

agir. Uma função passa por um controle de utilização ao atingir a cota de simultaneidade no nível de região da conta.

Estatística: máxima

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite para cerca de 90% da cota de simultaneidade definida para a conta na região. Por padrão, sua conta tem uma cota de simultaneidade de mil em todas as funções de uma região. No entanto, você pode verificar a cota da sua conta, pois ela pode ser aumentada entrando em contato com o suporte da AWS.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: GREATER_THAN_THRESHOLD

Lambda Insights

Recomendamos definir alarmes de práticas recomendadas para as seguintes métricas do Lambda Insights.

memory_utilization

Dimensões: function_name

Descrição do alarme: esse alarme é usado para detectar se a utilização da memória de uma função do Lambda está se aproximando do limite configurado. Para solucionar problemas, você pode tentar 1) Otimizar seu código. 2) Dimensionar corretamente sua alocação de memória, estimando com precisão os requisitos de memória. Você pode consultar o Lambda Power Tuning para realizar essa operação. 3) Use o pooling de conexões. Consulte Using Amazon RDS Proxy with Lambda para o pooling de conexões para o banco de dados do RDS. 4) Você também pode considerar a possibilidade de projetar suas funções para evitar o armazenamento de grandes quantidades de dados na memória entre as invocações.

Descrição do alarme: esse alarme é usado para detectar se a utilização da memória para a função do Lambda está se aproximando do limite configurado.

Estatística: média

Sugestão de limite: 90,0

Justificativa do limite: defina o limite como 90% para receber um alerta quando a utilização da memória exceder 90% da memória alocada. Você pode ajustar isso para um valor mais baixo caso se preocupe com a workload para a utilização da memória. Você também pode verificar os dados históricos dessa métrica e definir o limite adequadamente.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon VPC (**`AWS/NATGateway`**)

ErrorPortAllocation

Dimensões: NatGatewayId

Descrição do alarme: esse alarme ajuda a detectar quando o gateway NAT não consegue alocar portas para novas conexões. Para resolver esse problema, consulte Resolver erros de alocação de portas em um gateway NAT.

Intenção: esse alarme é usado para detectar se o gateway NAT não pôde alocar uma porta de origem.

Estatística: soma

Limite recomendado: 0,0

Justificativa do limite: se o valor de ErrorPortAllocation for maior que zero, isso significa que muitas conexões simultâneas para um único destino popular estão abertas por meio do gateway NAT.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

PacketsDropCount

Dimensões: NatGatewayId

Descrição do alarme: esse alarme ajuda a detectar quando os pacotes são descartados pelo gateway NAT. Isso pode ocorrer devido a um problema com o gateway NAT, portanto, verifique o painel de integridade do serviço da AWS para saber o status do gateway NAT da AWS em sua região. Isso pode ajudar você a correlacionar o problema de rede relacionado ao tráfego usando o gateway NAT.

Intenção: esse alarme é usado para detectar se os pacotes estão sendo descartados pelo gateway NAT.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: você deve calcular o valor de 0,01% do tráfego total no gateway NAT e usar esse resultado como valor limite. Use dados históricos do tráfego no gateway NAT para determinar o limite.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Link privado da AWS (**`AWS/PrivateLinkEndpoints`**)

PacketsDropped

Dimensões: ID da VPC, ID do endpoint da VPC, tipo de endpoint, ID da sub-rede, nome do serviço

Descrição do alarme: esse alarme ajuda a detectar se o endpoint ou o serviço de endpoint não está íntegro por meio do monitoramento do número de pacotes descartados pelo endpoint. Observe que os pacotes maiores que 8500 bytes que chegam ao endpoint da VPC são descartados. Para solução de problemas, consulte Solucionar problemas de conectividade entre um endpoint da VPC de interface e um serviço de endpoint.

Intenção: esse alarme é usado para detectar se o endpoint ou o serviço de endpoint não está íntegro.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite de acordo com o caso de uso. Se você quiser estar ciente do status de não integridade do endpoint ou do serviço de endpoint, defina o limite baixo para ter a chance de corrigir o problema antes de uma grande perda de dados. Você pode usar dados históricos para entender a tolerância de pacotes descartados e definir o limite adequadamente.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Link privado da AWS (**`AWS/PrivateLinkServices`**)

RstPacketsSent

Dimensões: ID do serviço, balanceador de carga Arn, Az

Descrição do alarme: esse alarme ajuda você a detectar alvos não íntegros de um serviço de endpoint com base no número de pacotes de redefinição enviados aos endpoints. Ao depurar erros de conexão com um consumidor do seu serviço, você pode validar se o serviço está redefinindo as conexões com a métrica RstPacketsSent ou se algo mais está falhando no caminho da rede.

Intenção: esse alarme é usado para detectar alvos não íntegros de um serviço de endpoint.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: o limite depende do caso de uso. Se o seu caso de uso puder tolerar que os alvos não sejam íntegros, você poderá definir o limite como alto. Se o caso de uso não tolerar alvos não íntegros, você poderá definir um limite como muito baixo.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon RDS

CPUUtilization

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar a alta utilização da CPU de forma consistente. A utilização da CPU mede o tempo não ocioso. Considere usar o Monitoramento Aprimorado ou o Insights de Performance para analisar qual tempo de espera está consumindo a maior parte do tempo da CPU (`guest`, `irq`, `wait`, `nice` e assim por diante) para o MariaDB, o MySQL, o Oracle e o PostgreSQL. Em seguida, avalie quais consultas consomem a maior quantidade de CPU. Se não for possível ajustar a workload, considere migrar para uma classe de instância de banco de dados maior.

Intenção: este alarme é usado para detectar uma alta utilização da CPU de forma consistente a fim de evitar tempos de resposta muito altos e o atingimento do tempo limite. Se desejar verificar a microexpansão da utilização da CPU, é possível definir um tempo de avaliação inferior para o alarme.

Estatística: média

Limite recomendado: 90,0

Justificativa do limite: os aumentos randômicos no consumo da CPU podem não prejudicar a performance do banco de dados, mas manter uma elevada utilização da CPU pode prejudicar futuras solicitações para o banco de dados. Dependendo da workload geral do banco de dados, a alta utilização da CPU na instância do RDS ou do Aurora pode degradar a performance geral.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

DatabaseConnections

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme detecta um grande número de conexões. Analise as conexões existentes e encerre aquelas que estiverem no estado “em repouso” ou que foram encerradas incorretamente. Considere usar um grupo de conexões para limitar o número de novas conexões. Como alternativa, aumente o tamanho da instância de banco de dados para usar uma classe com mais memória e, portanto, com um valor padrão mais alto para “max_connections”, ou aumente o valor de “max_connections” no RDS e no MySQL e PostgreSQL do Aurora para a classe atual, caso ela seja compatível com sua workload.

Intenção: este alarme é usado para ajudar a evitar conexões rejeitadas quando o número máximo de conexões para o banco de dados é atingido. Esse alarme não é recomendado se você altera a classe da instância de banco de dados com frequência, pois realiza alterações na memória e no número máximo de conexões padrão.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: o número de conexões permitidas depende do tamanho da classe da instância de banco de dados e dos parâmetros específicos para o mecanismo de banco de dados relacionados aos processos ou às conexões. Você deve calcular um valor entre 90 e 95% do número máximo de conexões para seu banco de dados e usar esse resultado como o valor limite.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

EBSByteBalance%

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar uma baixa porcentagem de créditos de throughput restantes. Para solução de problemas, verifique problemas de latência no RDS.

Intenção: este alarme é usado para detectar uma baixa porcentagem de créditos de throughput restantes no bucket de intermitência. A baixa porcentagem de saldo para bytes pode causar problemas de gargalo de throughput. Esse alarme não é recomendado para instâncias do Aurora PostgreSQL.

Estatística: média

Limite recomendado: 10,0

Justificativa do limite: um saldo de créditos de throughput inferior a 10% é considerado ruim e você deve definir o limite adequadamente. Além disso, é possível definir um limite mais baixo se a aplicação puder tolerar um throughput mais baixo para a workload.

Período: 60

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: LESS_THAN_THRESHOLD

EBSIOBalance%

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar uma baixa porcentagem de créditos de IOPS restantes. Para obter uma solução de problemas, consulte problemas de latência no RDS.

Intenção: este alarme é usado para detectar uma baixa porcentagem de créditos de E/S restantes no bucket de intermitência. A baixa porcentagem de saldo para IOPS pode causar problemas de gargalo de IOPS. Esse alarme não é recomendado para instâncias do Aurora.

Estatística: média

Limite recomendado: 10,0

Justificativa do limite: um saldo de créditos de IOPS inferior a 10% é considerado ruim e você pode definir o limite adequadamente. Além disso, é possível definir um limite mais baixo se a aplicação puder tolerar um número de IOPS mais baixo para a workload.

Período: 60

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: LESS_THAN_THRESHOLD

FreeableMemory

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar pouca memória que pode ser liberada, o que pode significar que há um aumento nas conexões do banco de dados ou que a instância pode estar sob alta pressão de memória. Verifique a pressão de memória ao monitorar as métricas do CloudWatch para `SwapUsage`, além de realizar o monitoramento para `FreeableMemory`. Se o consumo de memória da instância for frequentemente muito alto, indica que você deve verificar a workload ou atualizar a classe da instância. Para as instâncias de banco de dados de leitor do Aurora, considere adicionar mais instâncias de banco de dados de leitor ao cluster. Para obter informações sobre como solucionar problemas do Aurora, consulte problemas de memória que pode ser liberada.

Intenção: este alarme é usado para ajudar a evitar a falta de memória, o que pode resultar em conexões rejeitadas.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: dependendo da workload e da classe da instância, valores diferentes para o limite podem ser apropriados. Preferencialmente, a memória disponível não deve ser inferior a 25% da memória total por períodos prolongados. Para o Aurora, é possível definir o limite próximo pa 5%, porque a métrica próxima de zero significa que a instância de banco de dados aumentou a escala verticalmente o máximo possível. Você pode analisar o comportamento histórico dessa métrica para determinar níveis de limite razoáveis.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: LESS_THAN_THRESHOLD

FreeLocalStorage

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar pouco armazenamento local livre. A edição do Aurora compatível com PostgreSQL usa o armazenamento local para armazenar logs de erros e arquivos temporários. O Aurora MySQL usa armazenamento local para armazenar logs de erros, logs gerais, logs de consultas lentas, logs de auditoria e tabelas temporárias que não são do InnoDB. Esses volumes de armazenamento local são apoiados pelo Amazon EBS e podem ser ampliados ao usar uma classe de instância de banco de dados maior. Para obter a solução de problemas, verifique instâncias do Aurora compatíveis com PostgreSQL e compatíveis com MySQL.

Intenção: este alarme é usado para detectar o quão perto a instância de banco de dados do Aurora está de atingir o limite de armazenamento local, se você não usar o Aurora Sem Servidor v2 ou versões posteriores. O armazenamento local pode atingir a capacidade máxima quando você armazena dados não persistentes, como tabelas temporárias e arquivos de log, no armazenamento local. Esse alarme pode evitar um erro de falta de espaço que ocorre quando a instância de banco de dados fica sem o armazenamento local.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: você deve calcular cerca de 10% a 20% da quantidade de armazenamento disponível com base na velocidade e na tendência de uso do volume e, em seguida, usar esse resultado como o valor limite para tomar medidas proativas antes que o volume atinja o limite.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: LESS_THAN_THRESHOLD

FreeStorageSpace

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme monitora uma pequena quantidade de espaço de armazenamento disponível. Considere aumentar a escala verticalmente para o armazenamento do banco de dados caso você frequentemente se aproxime dos limites de capacidade de armazenamento. Inclua algum buffer para acomodar aumentos não previstos na demanda das aplicações. Como alternativa, considere habilitar o ajuste de escala automático para o

armazenamento do RDS. Além disso, considere liberar mais espaço ao excluir dados e logs não utilizados ou desatualizados. Para obter mais informações, verifique o documento do RDS para quando não há mais espaço de armazenamento e o documento sobre problemas de armazenamento do PostgreSQL.

Intenção: este alarme ajuda a evitar problemas de armazenamento cheio. O alarme pode evitar o tempo de inatividade que ocorre quando a instância de banco de dados fica sem armazenamento. Não recomendamos usar esse alarme se você tiver o ajuste de escala automático para o armazenamento habilitado ou caso realize alterações da capacidade de armazenamento da instância de banco de dados com frequência.

Estatística: mínima

Limite recomendado: depende da sua situação

Justificativa do limite: o valor do limite dependerá do espaço de armazenamento atualmente alocado. Normalmente, você deve calcular o valor de 10% do espaço de armazenamento alocado e usar esse resultado como valor limite.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: LESS_THAN_THRESHOLD

MaximumUsedTransactionIDs

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a evitar a conclusão de IDs de transação para o PostgreSQL. Consulte as etapas para a solução de problemas nesta publicação do blog para investigar e resolver o problema. Você também pode consultar esta publicação do blog para se familiarizar ainda mais com os conceitos de autovacuum, os problemas comuns e as práticas recomendadas.

Intenção: este alarme é usado para ajudar a evitar a conclusão de IDs de transação para o PostgreSQL.

Estatística: média

Limite recomendado: 1.0E9

Justificativa do limite: definir este limite para um bilhão deverá fornecer tempo hábil para que você investigue o problema. O valor padrão para autovacuum_freeze_max_age é de 200 milhões. Se o tempo da transação mais antiga for de um bilhão, o autovacuum terá problemas para manter esse limite abaixo da meta de 200 milhões de IDs de transação.

Período: 60

Pontos de dados para o alarme: 1

Períodos de avaliação: 1

Operador de comparação: GREATER_THAN_THRESHOLD

ReadLatency

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar a alta latência de leitura. Se a latência de armazenamento for alta, é porque a workload está excedendo os limites de recursos. É possível analisar a utilização de E/S em relação à configuração da instância e do armazenamento alocado. Consulte a solução de problemas de latência de volumes do Amazon EBS causada por um gargalo de IOPS. Para o Aurora, é possível alternar para uma classe da instância que tenha configuração de armazenamento I/O-Optimized. Consulte Planning I/O in Aurora para obter orientação.

Intenção: este alarme é usado para detectar alta latência de leitura. Geralmente, os discos de banco de dados têm baixa latência de leitura e de gravação, mas podem apresentar problemas que podem causar operações com alta latência.

Estatística: p90

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme depende muito do seu caso de uso. Latências de leitura superiores a 20 milissegundos são provavelmente motivo para uma investigação. Também é possível definir um limite mais alto caso a aplicação possa ter uma latência mais alta para as operações de leitura. Avalie a criticidade e os requisitos para a latência de leitura e analise o comportamento histórico dessa métrica para determinar níveis de limite razoáveis.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

ReplicaLag

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda você a compreender quantos segundos uma réplica tem de diferença quando comparada com a instância primária. Uma réplica de leitura do PostgreSQL relata um atraso de replicação de até cinco minutos caso não haja transações de usuário ocorrendo na instância de banco de dados de origem. Quando a métrica ReplicaLag atinge zero, a réplica alcançou a instância de banco de dados primária. Se a métrica ReplicaLag retornar -1, significa que a replicação não está ativa no momento. Para obter orientações relacionadas ao RDS para PostgreSQL, consulte as práticas recomendadas de replicação. Para solucionar problemas relacionados à métrica `ReplicaLag` e erros relacionados, consulte a solução de problemas para ReplicaLag.

Intenção: este alarme pode detectar o atraso da réplica, que reflete a perda de dados que pode ocorrer em caso de falha da instância primária. Se a réplica tiver uma diferença muito grande quando comparada com a instância primária e ela falhar, faltarão dados na réplica que estavam na instância primária.

Estatística: máxima

Limite recomendado: 60,0

Justificativa do limite: normalmente, o atraso aceitável depende da aplicação. Recomendamos não mais do que 60 segundos.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: GREATER_THAN_THRESHOLD

WriteLatency

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar a alta latência de gravação. Se a latência de armazenamento for alta, é porque a workload está excedendo os limites de recursos. É possível analisar a utilização de E/S em relação à configuração da instância e do armazenamento alocado. Consulte a solução de problemas de latência de volumes do Amazon EBS causada por um gargalo de IOPS. Para o Aurora, é possível alternar para uma classe da instância que tenha configuração de armazenamento I/O-Optimized. Consulte Planning I/O in Aurora para obter orientação.

Intenção: este alarme é usado para detectar alta latência de gravação. Embora, geralmente, os discos de banco de dados tenham baixa latência de leitura e de gravação, eles podem enfrentar problemas que causam operações com alta latência. Monitorar isso garantirá que a latência do disco seja tão baixa quanto o esperado.

Estatística: p90

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme depende muito do seu caso de uso. Latências de gravação superiores a 20 milissegundos são provavelmente motivo para uma investigação. Também é possível definir um limite mais alto caso a aplicação puossa ter uma latência mais alta para as operações de gravação. Avalie a criticidade e os requisitos para a latência de gravação e analise o comportamento histórico dessa métrica para determinar níveis de limite razoáveis.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

DBLoad

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar a alta carga do banco de dados. Se o número de processos exceder o número de vCPUs, os processos começarão a ser colocados em fila. Quando o enfileiramento aumenta, a performance é impactada. Se a carga do banco de dados estiver frequentemente acima da vCPU máxima e o estado de espera primário for a CPU, ela estará sobrecarregada. Nesse caso, é possível monitorar `CPUUtilization`, `DBLoadCPU` e as tarefas enfileiradas no Insights de Performance ou no Monitoramento Aprimorado. Talvez

você deseje realizar o controle de utilização das conexões para a instância, ajustar quaisquer consultas SQL com uma alta carga de CPU ou considerar uma classe da instância maior. As instâncias altas e consistentes de qualquer estado de espera indicam que pode haver problemas de gargalos ou de contenção de recursos que você deve resolver.

Intenção: este alarme é usado para detectar uma alta carga do banco de dados. A alta carga do banco de dados pode causar problemas de performance na instância de banco de dados. Esse alarme não é aplicável para instâncias de banco de dados com tecnologia sem servidor.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: o valor máximo de vCPU é determinado pelo número de núcleos de vCPU (CPU virtual) da instância de banco de dados. Dependendo da vCPU máxima, valores diferentes para o limite podem ser apropriados. Preferencialmente, a carga do banco de dados não deve ultrapassar a linha da vCPU.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

AuroraVolumeBytesLeftTotal

Dimensões: DBClusterIdentifier

Descrição do alarme: este alarme ajuda a monitorar o baixo volume total restante. Quando o volume total restante atinge o limite de tamanho, o cluster relata um erro de falta de espaço. O armazenamento do Aurora é escalado automaticamente com os dados no volume do cluster e é ampliado até 128 TiB ou 64 TiB, dependendo da versão do mecanismo de banco de dados. Considere reduzir o armazenamento ao descartar as tabelas e os bancos de dados que não são mais necessários. Para obter mais informações, consulte a escalabilidade de armazenamento.

Intenção: este alarme é usado para detectar o quão próximo o cluster do Aurora está do limite de tamanho para o volume. Esse alarme pode evitar um erro de falta de espaço que ocorre quando o cluster fica sem espaço. Ele é recomendado somente para o Aurora MySQL.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: você deve calcular de 10% a 20% do limite de tamanho real com base na velocidade e na tendência de aumento de uso do volume e, em seguida, usar esse resultado como o valor limite para tomar medidas proativas antes que o volume atinja o limite.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: LESS_THAN_THRESHOLD

AuroraBinlogReplicaLag

Dimensões: DBClusterIdentifier, Role=WRITER

Descrição do alarme: este alarme ajuda a monitorar o estado de erro da replicação da instância do gravador do Aurora. Para ter mais informações, consulte Como replicar clusters de bancos de dados Amazon Aurora MySQL entre regiões da AWS. Para obter a solução de problemas, consulte Problemas de replicação no Aurora MySQL.

Intenção: este alarme é usado para detectar se a instância do gravador está em um estado de erro e não pode replicar a origem. Ele é recomendado somente para o Aurora MySQL.

Estatística: média

Limite recomendado: -1,0

Justificativa do limite: recomendamos usar -1 como o valor limite porque o Aurora MySQL publicará esse valor se a réplica estiver em um estado de erro.

Período: 60

Pontos de dados para o alarme: 2

Períodos de avaliação: 2

Operador de comparação: LESS_THAN_OR_EQUAL_TO_THRESHOLD

BlockedTransactions

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar uma alta contagem de transações bloqueadas em uma instância de banco de dados do Aurora. As transações bloqueadas podem terminar em uma reversão ou uma confirmação. A alta simultaneidade, a ociosidade nas transações ou as transações de longa execução podem ocasionar transações bloqueadas. Para obter a solução de problemas, consulte a documentação do Aurora MySQL.

Intenção: este alarme é usado para detectar uma alta contagem de transações bloqueadas em uma instância de banco de dados do Aurora com a finalidade de evitar as reversões de transações e a degradação da performance.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: você deve calcular 5% de todas as transações da sua instância usando a métrica `ActiveTransactions` e usar esse resultado como o valor limite. Também é possível avaliar a criticidade e os requisitos para as transações bloqueadas e analisar o comportamento histórico dessa métrica para determinar níveis de limite razoáveis.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

BufferCacheHitRatio

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar uma proporção de ocorrências do cache que esteja baixa e consistente para o cluster do Aurora. Uma alta de acertos baixa indica que as consultas nessa instância de banco de dados estão acessando o disco com maior frequência. Para solucionar problemas, investigue a workload para visualizar quais consultas estão causando esse comportamento e consulte o documento de recomendações de RAM para a instância de banco de dados.

Intenção: este alarme é usado para detectar uma proporção de ocorrências do cache que esteja baixa e consistente a fim de evitar que se mantenha uma diminuição da performance na instância do Aurora.

Estatística: média

Limite recomendado: 80,0

Justificativa do limite: é possível definir o limite para a proporção de ocorrências do cache em buffer como 80%. No entanto, você pode ajustar esse valor com base no seu nível de performance aceitável e nas características da workload.

Período: 60

Pontos de dados para o alarme: 10

Períodos de avaliação: 10

Operador de comparação: LESS_THAN_THRESHOLD

EngineUptime

Dimensões: DBClusterIdentifier, Role=WRITER

Descrição do alarme: este alarme ajuda a monitorar o baixo tempo de inatividade da instância de banco de dados do gravador. A instância de banco de dados do gravador pode se tornar inativa devido a uma reinicialização, uma manutenção, uma atualização ou um failover. Quando o tempo de atividade atinge zero devido a um failover no cluster, e o cluster tem uma ou mais réplicas do Aurora, uma réplica do Aurora é promovida como a instância primária do gravador durante um evento de falha. Para aumentar a disponibilidade do cluster de banco de dados, considere criar uma ou mais réplicas do Aurora em duas ou mais zonas de disponibilidade diferentes. Para obter mais informações, verifique os fatores que influenciam o tempo de inatividade do Aurora.

Intenção: este alarme é usado para detectar se a instância de banco de dados do gravador do Aurora está com um tempo de inatividade. Isso pode evitar falhas de execução prolongada na instância do gravador que ocorrem devido a uma falha ou a um failover.

Estatística: média

Limite recomendado: 0,0

Justificativa do limite: um evento de falha resulta em uma breve interrupção durante a qual as operações de leitura e de gravação falham com uma exceção. No entanto, o serviço é restaurado normalmente em menos de 60 segundos e muitas vezes em menos de 30 segundos.

Período: 60

Pontos de dados para o alarme: 2

Períodos de avaliação: 2

Operador de comparação: LESS_THAN_OR_EQUAL_TO_THRESHOLD

RollbackSegmentHistoryListLength

Dimensões: DBInstanceIdentifier

Descrição do alarme: este alarme ajuda a monitorar um tamanho consistente e amplo para o histórico do segmento de reversão de uma instância do Aurora. Um tamanho amplo para a lista de histórico do InnoDB indica que um grande número de versões antigas de linhas, consultas e desligamentos de banco de dados tornaram-se mais lentos. Para obter mais informações e a solução de problemas, consulte a documentação O tamanho da lista de histórico do InnoDB aumentou significativamente.

Intenção: este alarme é usado para detectar um tamanho amplo e consistente para o histórico do segmento de reversão. Isso pode ajudar a evitar manter a degradação da performance e o alto uso da CPU na instância do Aurora. Ele é recomendado somente para o Aurora MySQL.

Estatística: média

Limite recomendado: 1.000.000,0

Justificativa do limite: definir este limite para um milhão deverá fornecer tempo hábil para que você investigue o problema. No entanto, você pode ajustar esse valor com base no seu nível de performance aceitável e nas características da workload.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

StorageNetworkThroughput

Dimensões: DBClusterIdentifier, Role=WRITER

Descrição do alarme: este alarme ajuda a monitorar o alto throughput da rede de armazenamento. Se o throughput da rede de armazenamento ultrapassar a largura de banda da

rede total da instância do EC2, ele poderá ocasionar uma alta latência de leitura e de gravação, o que pode causar degradação da performance. É possível verificar o tipo de instância do EC2 no Console da AWS. Para solucionar problemas, verifique quaisquer alterações nas latências de gravação e de leitura e avalie se você também acionou um alarme nessa métrica. Se for esse o caso, avalie o padrão de workload durante os horários em que o alarme foi acionado. Isso pode ajudar você a identificar se é possível otimizar a workload para reduzir a quantidade total de tráfego de rede. Se isso não for possível, talvez seja necessário considerar escalar a instância.

Intenção: este alarme é usado para detectar o alto throughput da rede de armazenamento. A detecção de alto throughput pode evitar as quedas de pacotes de rede e a degradação da performance.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: você deve calcular cerca de 80% a 90% da largura de banda da rede total do tipo de instância do EC2 e, em seguida, usar esse resultado como o valor limite para tomar medidas proativas antes que os pacotes de rede sejam afetados. Também é possível avaliar a criticidade e os requisitos para o throughput da rede de armazenamento e analisar o comportamento histórico dessa métrica para determinar níveis de limite razoáveis.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon Route 53 Public Data Plane

HealthCheckStatus

Dimensões: HealthCheckId

Descrição do alarme: esse alarme ajuda a detectar endpoints não íntegros de acordo com os verificadores de integridade. Para entender o motivo de uma falha que resulta em status de não integridade, use a guia Verificadores de integridade no console de verificação de integridade do Route 53 para visualizar o status de cada região, bem como a última falha da verificação de

integridade. A guia de status também exibe o motivo pelo qual o endpoint é relatado como não íntegro. Consulte as etapas de solução de problemas.

Intenção: esse alarme usa os verificadores de integridade do Route 53 para detectar endpoints não íntegros.

Estatística: média

Limite recomendado: 1,0

Justificativa do limite: o status do endpoint é relatado como 1 quando ele está íntegro. Qualquer valor inferior a 1 indica não integridade.

Período: 60

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: LESS_THAN_THRESHOLD

Amazon S3

4xxErrors

Dimensões: BucketName, FilterId

Descrição do alarme: esse alarme nos ajuda a informar o número total de códigos de status de erro 4xx que são criados em resposta a solicitações de clientes. Os códigos de erro 403 podem indicar uma política do IAM incorreta e os códigos de erro 404 podem indicar uma aplicação cliente com comportamento incorreto, por exemplo. Habilitar o registro em log de acesso ao servidor do S3 ajudará você a identificar a origem do problema usando os campos Status HTTP e Código de erro. Para entender mais sobre o código de erro, consulte Respostas de erro.

Intenção: esse alarme é usado para criar uma linha de base para as taxas de erro 4xx típicas, de modo que você possa analisar qualquer anormalidade que possa indicar um problema de configuração.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: o limite recomendado é detectar se mais de 5% do total de solicitações estão recebendo erros 4XX. Os erros 4XX que ocorrem com frequência devem receber um alarme. Entretanto, a definição de um valor muito baixo para o limite pode fazer com que o alarme seja muito sensível. Você também pode ajustar o limite de acordo com a carga das solicitações, levando em conta um nível aceitável de erros 4XX. Você também pode analisar dados históricos para encontrar a taxa de erro aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

5xxErrors

Dimensões: BucketName, FilterId

Descrição do alarme: esse alarme ajuda você a detectar um grande número de erros no lado do servidor. Esses erros indicam que um cliente fez uma solicitação que o servidor não conseguiu concluir. Isso pode ajudar você a correlacionar o problema que sua aplicação está enfrentando por causa do S3. Para obter mais informações para ajudar você a tratar ou reduzir erros de forma eficiente, consulte Optimizing performance design patterns. Os erros também podem ser causados por um problema com o S3. Verifique o painel de integridade do serviço da AWS para saber o status do Amazon S3 em sua região.

Intenção: esse alarme pode ajudar a detectar se a aplicação está apresentando problemas devido a erros 5xx.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: recomendamos definir o limite para detectar se mais de 5% do total de solicitações estão recebendo o 5XXError. No entanto, você pode ajustar o limite para se adequar ao tráfego das solicitações, bem como às taxas de erro aceitáveis. Você também pode analisar dados históricos para ver qual é a taxa de erro aceitável para a workload da aplicação e ajustar o limite adequadamente.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

OperationsFailedReplication

Dimensões: SourceBucket, DestinationBucket, RuleId

Descrição do alarme: esse alarme ajuda a entender uma falha de replicação. Essa métrica rastreia o status de novos objetos replicados usando o S3 CRR ou S3 SRR e também rastreia os objetos existentes replicados usando a replicação em lote do S3. Consulte Solução de problemas de replicação para obter mais detalhes.

Intenção: esse alarme é usado para detectar se houve falha na operação de replicação.

Estatística: máxima

Limite recomendado: 0,0

Justificativa do limite: essa métrica emite um valor de 0 para operações bem-sucedidas e nenhum valor quando não há operações de replicação realizadas para o minuto. Quando a métrica emite um valor maior que 0, a operação de replicação não é bem-sucedida.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

S3ObjectLambda

4xxErrors

Dimensões: AccessPointName, DataSourceARN

Descrição do alarme: esse alarme nos ajuda a informar o número total de códigos de status de erro 4xx que são criados em resposta a solicitações de clientes. Habilitar o registro em log de acesso ao servidor do S3 ajudará você a identificar a origem do problema usando os campos Status HTTP e Código de erro.

Intenção: esse alarme é usado para criar uma linha de base para as taxas de erro 4xx típicas, de modo que você possa analisar qualquer anormalidade que possa indicar um problema de configuração.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: recomendamos definir o limite para detectar se mais de 5% do total de solicitações estão recebendo o 4XXError. Os erros 4XX que ocorrem com frequência devem receber um alarme. Entretanto, a definição de um valor muito baixo para o limite pode fazer com que o alarme seja muito sensível. Você também pode ajustar o limite de acordo com a carga das solicitações, levando em conta um nível aceitável de erros 4XX. Você também pode analisar dados históricos para encontrar a taxa de erro aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

5xxErrors

Dimensões: AccessPointName, DataSourceARN

Descrição do alarme: esse alarme ajuda a detectar um grande número de erros no lado do servidor. Esses erros indicam que um cliente fez uma solicitação que o servidor não conseguiu concluir. Esses erros podem ser causados por um problema com o S3. Verifique o painel de integridade do serviço da AWS para saber o status do Amazon S3 em sua região. Isso pode ajudar você a correlacionar o problema que sua aplicação está enfrentando por causa do S3. Para obter informações que o ajudem a tratar ou reduzir esses erros com eficiência, consulte Optimizing performance design patterns.

Intenção: esse alarme pode ajudar a detectar se a aplicação está apresentando problemas devido a erros 5xx.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: recomendamos definir o limite para detectar se mais de 5% do total de solicitações estão recebendo erros 5XX. No entanto, você pode ajustar o limite para se adequar ao tráfego das solicitações, bem como às taxas de erro aceitáveis. Você também pode analisar dados históricos para ver qual é a taxa de erro aceitável para a workload da aplicação e ajustar o limite adequadamente.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

LambdaResponse4xx

Dimensões: AccessPointName, DataSourceARN

Descrição do alarme: esse alarme ajuda você a detectar e diagnosticar falhas (500s) em chamadas para o S3 Object Lambda. Esses erros podem ser causados por erros ou configurações incorretas na função do Lambda responsável por responder às suas solicitações. Investigar os fluxos de logs do CloudWatch da função do Lambda associada ao ponto de acesso do Object Lambda pode ajudar você a identificar a origem do problema com base na resposta do S3 Object Lambda.

Intenção: esse alarme é usado para detectar erros de cliente 4xx para chamadas WriteGetObjectResponse.

Estatística: média

Limite recomendado: 0,05

Justificativa do limite: recomendamos definir o limite para detectar se mais de 5% do total de solicitações estão recebendo o 4XXError. Os erros 4XX que ocorrem com frequência devem receber um alarme. Entretanto, a definição de um valor muito baixo para o limite pode fazer com que o alarme seja muito sensível. Você também pode ajustar o limite de acordo com a carga das solicitações, levando em conta um nível aceitável de erros 4XX. Você também pode analisar dados históricos para encontrar a taxa de erro aceitável para a workload da aplicação e, em seguida, ajustar o limite adequadamente.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon SNS

NumberOfMessagesPublished

Dimensões: TopicName

Descrição do alarme: esse alarme pode detectar quando o número de mensagens do SNS publicadas é muito baixo. Para solucionar problemas, verifique por que os publicadores estão enviando menos tráfego.

Intenção: esse alarme ajuda você a monitorar e detectar proativamente quedas significativas na publicação de notificações. Isso ajuda você a identificar possíveis problemas com a aplicação ou com os processos comerciais, de modo que você possa tomar as medidas adequadas para manter o fluxo esperado de notificações. Você deve criar esse alarme se você espera que seu sistema tenha um tráfego mínimo a ser atendido.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: o número de mensagens publicadas deve estar de acordo com o número esperado de mensagens publicadas para sua aplicação. Você também pode analisar os dados históricos, as tendências e o tráfego para encontrar o limite correto.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: LESS_THAN_THRESHOLD

NumberOfNotificationsDelivered

Dimensões: TopicName

Descrição do alarme: esse alarme pode detectar quando o número de mensagens do SNS entregues é muito baixo. Isso pode ocorrer devido ao cancelamento não intencional da assinatura de um endpoint ou devido a um evento do SNS que causa atraso nas mensagens.

Intenção: esse alarme ajuda você a detectar uma queda no volume de mensagens entregues. Você deve criar esse alarme se você espera que seu sistema tenha um tráfego mínimo a ser atendido.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: o número de mensagens entregues deve estar de acordo com o número esperado de mensagens produzidas e o número de consumidores. Você também pode analisar os dados históricos, as tendências e o tráfego para encontrar o limite correto.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: LESS_THAN_THRESHOLD

NumberOfNotificationsFailed

Dimensões: TopicName

Descrição do alarme: esse alarme pode detectar quando o número de mensagens do SNS com falha é muito alto. Para solucionar problemas de notificações com falha, ative o registro em log no CloudWatch Logs. A verificação dos logs pode ajudar a descobrir quais assinantes estão apresentando falhas, bem como os códigos de status que estão retornando.

Intenção: esse alarme ajuda você a encontrar proativamente problemas com a entrega de notificações e a tomar as medidas adequadas para resolvê-los.

Estatística: soma

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme depende muito do impacto das notificações com falha. Analise os SLAs fornecidos aos seus usuários finais, a tolerância a falhas e a importância das notificações, e analise os dados históricos, e então selecione um

limite adequadamente. O número de notificações com falha deve ser 0 para tópicos que tenham apenas assinaturas do SQS, Lambda ou Firehose.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

NumberOfNotificationsFilteredOut-InvalidAttributes

Dimensões: TopicName

Descrição do alarme: esse alarme ajuda a monitorar e resolver possíveis problemas com o publicador ou com os assinantes. Verifique se um publicador está publicando mensagens com atributos inválidos ou se um filtro inadequado foi aplicado a um assinante. Você também pode analisar o CloudWatch Logs para ajudar a encontrar a causa-raiz do problema.

Intenção: o alarme é usado para detectar se as mensagens publicadas não são válidas ou se foram aplicados filtros inadequados a um assinante.

Estatística: soma

Limite recomendado: 0,0

Justificativa do limite: atributos inválidos são quase sempre um erro do publicador. Recomendamos definir o limite como 0 porque atributos inválidos não são esperados em um sistema íntegro.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

NumberOfNotificationsFilteredOut-InvalidMessageBody

Dimensões: TopicName

Descrição do alarme: esse alarme ajuda a monitorar e resolver possíveis problemas com o publicador ou com os assinantes. Verifique se um publicador está publicando mensagens com

corpos de mensagem inválidos ou se um filtro inadequado foi aplicado a um assinante. Você também pode analisar o CloudWatch Logs para ajudar a encontrar a causa-raiz do problema.

Intenção: o alarme é usado para detectar se as mensagens publicadas não são válidas ou se foram aplicados filtros inadequados a um assinante.

Estatística: soma

Limite recomendado: 0,0

Justificativa do limite: corpos de mensagem inválidos são quase sempre um erro do publicador. Recomendamos definir o limite como 0 porque corpos de mensagens inválidos não são esperados em um sistema íntegro.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

NumberOfNotificationsRedrivenToDlq

Dimensões: TopicName

Descrição do alarme: esse alarme ajuda a monitorar o número de mensagens que são movidas para uma fila de mensagens não entregues.

Intenção: o alarme é usado para detectar mensagens que foram movidas para uma fila de mensagens não entregues. Recomendamos que você crie esse alarme quando o SNS estiver acoplado ao SQS, Lambda ou Firehose.

Estatística: soma

Limite recomendado: 0,0

Justificativa do limite: em um sistema íntegro de qualquer tipo de assinante, as mensagens não devem ser movidas para a fila de mensagens não entregues. Recomendamos que você receba uma notificação caso alguma mensagem caia na fila para que possa identificar e resolver a causa-raiz e, possivelmente, redirecionar as mensagens na fila de mensagens não entregues para evitar a perda de dados.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

NumberOfNotificationsFailedToRedriveToDlq

Dimensões: TopicName

Descrição do alarme: esse alarme ajuda a monitorar as mensagens que não puderam ser movidas para uma fila de mensagens não entregues. Verifique se a fila de mensagens não entregues existe e se está configurada corretamente. Além disso, verifique se o SNS tem permissões para acessar a fila de mensagens não entregues. Consulte a documentação da fila de mensagens não entregues para saber mais.

Intenção: o alarme é usado para detectar mensagens que não puderam ser movidas para uma fila de mensagens não entregues.

Estatística: soma

Limite recomendado: 0,0

Justificativa do limite: quase sempre é um erro se as mensagens não puderem ser movidas para a fila de mensagens não entregues. A recomendação para o limite é 0, o que significa que todas as mensagens que falharem no processamento deverão ser capazes de alcançar a fila de mensagens não entregues quando a fila tiver sido configurada.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

SMSMonthToDateSpentUSD

Dimensões: TopicName

Descrição do alarme: o alarme ajuda a monitorar se você tem uma cota suficiente em sua conta para que o SNS possa entregar mensagens. Se você atingir sua cota, o SNS não poderá enviar mensagens SMS. Para obter informações sobre como definir sua cota mensal de gastos com

SMS ou sobre como solicitar um aumento da cota de gastos com a AWS, consulte Definição das preferências de mensagens SMS.

Intenção: esse alarme é usado para detectar se você tem uma cota suficiente em sua conta para que as mensagens SMS sejam entregues com êxito.

Estatística: máxima

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite de acordo com a cota (limite de gastos da conta) da conta. Escolha um limite que informe a você com antecedência suficiente que você está atingindo o limite da cota para que você tenha tempo de solicitar um aumento.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

SMSSuccessRate

Dimensões: TopicName

Descrição do alarme: esse alarme ajuda a monitorar a taxa de falhas nas entregas de mensagens SMS. Você pode configurar o Cloudwatch Logs para entender a natureza da falha e tomar medidas com base nisso.

Intenção: esse alarme é usado para detectar falhas na entrega de mensagens SMS.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: defina o limite do alarme de acordo com sua tolerância para falhas na entrega de mensagens SMS.

Período: 60

Pontos de dados para o alarme: 5

Períodos de avaliação: 5

Operador de comparação: GREATER_THAN_THRESHOLD

Amazon SQS

ApproximateAgeOfOldestMessage

Dimensões: QueueName

Descrição do alarme: esse alarme observa a idade da mensagem mais antiga na fila. Você pode usar esse alarme para monitorar se os seus consumidores estão processando mensagens do SQS na velocidade desejada. Considere a possibilidade de aumentar o número de consumidores ou o throughput dos consumidores para reduzir a idade das mensagens. Essa métrica pode ser usada em combinação com `ApproximateNumberOfMessagesVisible` para determinar o tamanho do backlog da fila e a rapidez com que as mensagens estão sendo processadas. Para evitar que as mensagens sejam excluídas antes de serem processadas, considere a possibilidade de configurar a fila de mensagens não entregues para deixar de lado as possíveis mensagens “poison pill”.

Intenção: esse alarme é usado para detectar se a idade da mensagem mais antiga na fila QueueName é muito alta. Uma idade maior pode ser uma indicação de que as mensagens não estão sendo processadas com rapidez suficiente ou de que há algumas mensagens “poison-pill” que estão presas na fila e não podem ser processadas.

Estatística: máxima

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme depende muito do tempo esperado de processamento da mensagem. Você pode usar dados históricos para calcular o tempo médio de processamento de mensagens e, em seguida, definir o limite como 50% maior do que o tempo máximo esperado de processamento de mensagens do SQS pelos consumidores da fila.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

ApproximateNumberOfMessagesNotVisible

Dimensões: QueueName

Descrição do alarme: esse alarme ajuda a detectar um número alto de mensagens em andamento com relação ao `QueueName`. Para solucionar problemas, verifique message backlog decreasing.

Intenção: esse alarme é usado para detectar um número alto de mensagens em andamento na fila. Se os consumidores não excluírem as mensagens dentro do período de tempo limite de visibilidade, quando a fila for pesquisada, as mensagens reaparecerão na fila. Para filas FIFO, pode haver um máximo de 20 mil mensagens em andamento. Se você atingir essa cota, o SQS não retornará nenhuma mensagem de erro. Uma fila FIFO examina as primeiras 20.000 mensagens para determinar os grupos de mensagens disponíveis. Isso significa que, se houver um acúmulo de mensagens em um único grupo de mensagens, não será possível consumir mensagens de outros grupos de mensagens que foram enviadas para a fila posteriormente até que as mensagens do backlog sejam processadas com êxito.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: o valor limite recomendado para esse alarme é altamente dependente do número esperado de mensagens em andamento. Você pode usar dados históricos para calcular o número máximo esperado de mensagens em andamento e definir o limite para 50% acima desse valor. Se os consumidores da fila estiverem processando, mas não excluindo mensagens da fila, esse número aumentará repentinamente.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

ApproximateNumberOfMessagesVisible

Dimensões: QueueName

Descrição do alarme: esse alarme observa se o backlog da fila de mensagens é maior do que o esperado, indicando que os consumidores estão muito lentos ou que não há consumidores

suficientes. Considere aumentar a contagem de consumidores ou acelerar os consumidores, se esse alarme entrar no estado ALARME.

Intenção: esse alarme é usado para detectar se a contagem de mensagens da fila ativa está muito alta e se os consumidores estão demorando para processar as mensagens ou se não há consumidores suficientes para processá-las.

Estatística: média

Limite recomendado: depende da sua situação

Justificativa do limite: um número inesperadamente alto de mensagens visíveis indica que as mensagens não estão sendo processadas por um consumidor na taxa esperada. Você deve considerar os dados históricos ao definir esse limite.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

NumberOfMessagesSent

Dimensões: QueueName

Descrição do alarme: esse alarme ajuda a detectar se não há mensagens sendo enviadas de um produtor com relação ao `QueueName`. Para solucionar problemas, verifique o motivo pelo qual o produtor não está enviando mensagens.

Intenção: esse alarme é usado para detectar quando um produtor para de enviar mensagens.

Estatística: soma

Limite recomendado: 0,0

Justificativa do limite: se o número de mensagens enviadas for 0, o produtor não está enviando nenhuma mensagem. Se essa fila tiver um TPS baixo, aumente o número de EvaluationPeriods adequadamente.

Período: 60

Pontos de dados para o alarme: 15

Períodos de avaliação: 15

Operador de comparação: LESS_THAN_OR_EQUAL_TO_THRESHOLD

AWS VPN

TunnelState

Dimensões: VpnId

Descrição do alarme: esse alarme ajuda você a entender se o estado de um ou mais túneis é INATIVO. Para solucionar problemas, consulte VPN tunnel troubleshooting.

Intenção: esse alarme é usado para detectar se pelo menos um túnel está no estado INATIVO para essa VPN para que você possa solucionar o problema da VPN afetada. Esse alarme sempre estará no estado ALARME para redes que tenham apenas um único túnel configurado.

Estatística: mínima

Limite recomendado: 1,0

Justificativa do limite: um valor menor que 1 indica que pelo menos um túnel está no estado INATIVO.

Período: 300

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: LESS_THAN_THRESHOLD

TunnelState

Dimensões: TunnelIpAddress

Descrição do alarme: esse alarme ajuda você a entender se o estado desse túnel é INATIVO. Para solucionar problemas, consulte VPN tunnel troubleshooting.

Intenção: esse alarme é usado para detectar se o túnel está no estado INATIVO para que você possa solucionar o problema da VPN afetada. Esse alarme sempre estará no estado ALARME para redes que tenham apenas um único túnel configurado.

Estatística: mínima

Limite recomendado: 1,0

Justificativa do limite: um valor menor que 1 indica que o túnel está no estado INATIVO.

Período: 300

Pontos de dados para o alarme: 3

Períodos de avaliação: 3

Operador de comparação: LESS_THAN_THRESHOLD

Geração de alarmes para métricas

As etapas nas seções a seguir explicam como criar alarmes do CloudWatch para métricas.

Criar um alarme do CloudWatch com base em um limite estático

Escolha uma métrica do CloudWatch a ser observada pelo alarme e o limite dessa métrica. O alarme passará para o estado `ALARM` quando a métrica atingir o limite de um número especificado de períodos de avaliação.

Se você estiver criando um alarme em uma conta configurada como uma conta de monitoramento na observabilidade entre contas do CloudWatch, será possível configurar o alarme para observar uma métrica em uma conta de origem vinculada a essa conta de monitoramento. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

Para criar um alarme com base em uma única métrica

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Selecione Criar alarme.
4. Escolha Select metric (Selecionar métrica).
5. Execute um destes procedimentos:
   - Escolha o namespace do serviço que contém a métrica desejada. Continue escolhendo as opções à medida que elas são exibidas para restringir as escolhas. Quando uma lista de métricas for exibida, marque a caixa de seleção ao lado da métrica que você deseja.

- Na caixa de pesquisa, insira o nome de uma métrica, ID de conta, rótulo de conta, dimensão ou ID de recurso. Escolha um dos resultados e continue até uma lista de métricas ser exibida. Marque a caixa de seleção ao lado da métrica que você deseja.

6. Escolha a guia Graphed metrics (Métricas em gráfico).

   a. Em Statistic (Estatística), escolha uma das estatísticas ou percentis predefinidos ou especifique um percentil personalizado (por exemplo, **`p95.45`**).

   b. Em Período, escolha o período de avaliação do alarme. Ao avaliar o alarme, todos os períodos são agregados em um único ponto de dados.

      Também escolha se a legenda do eixo Y é exibida no lado esquerdo ou no lado direito enquanto você está criando o alarme. Essa preferência só é usada enquanto você está criando o alarme.

   c. Escolha Selecionar métrica.

      A página Specify metric and conditions (Especificar métrica e condições) será exibida, mostrando um gráfico e outras informações sobre a métrica e a estatística que você selecionou.

7. Em Conditions (Condições), especifique o seguinte:

   a. Em Whenever ***`metric`*** is (Sempre que a métrica for), especifique se a métrica deve ser maior que, menor que ou igual ao limite. Em than... (que...), especifique o valor limite.

   b. Escolha Additional configuration (Configuração adicional). Em Datapoints to alarm (Pontos de dados para alarme), especifique quantos períodos de avaliação (pontos de dados) devem estar no estado `ALARM` para disparar o alarme. Se os dois valores forem correspondentes, você criará um alarme que passa para o estado `ALARM` se esses períodos consecutivos estiverem violando.

      Para criar um alarme M de N, especifique um número menor para o primeiro valor que especificar para o segundo valor. Para ter mais informações, consulte Avaliar um alarme.

   c. Para o Missing data treatment (Tratamento de dados ausentes), escolha como deseja que o alarme se comporte quando alguns pontos de dados estiverem ausentes. Para ter mais informações, consulte Configurar como os alarmes do CloudWatch tratam dados ausentes.

   d. Se o alarme usar um percentil como estatística monitorada, uma caixa Percentiles with low samples (Percentis com amostras baixas) será exibida. Use-a para escolher se deseja avaliar ou ignorar casos com taxas de amostra baixas. Se você escolher ignore (maintain the alarm state) (ignorar (manter o estado do alarme)), o estado do alarme atual será

sempre mantido quando o tamanho da amostra for muito baixo. Para ter mais informações, consulte Alarmes do CloudWatch baseados em percentual e exemplos de poucos dados.

8. Escolha Próximo.
9. Em Notification (Notificação), selecione um tópico do SNS para notificar quando o alarme estiver no estado `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

   Para que o alarme envie várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

   Na observabilidade entre contas do CloudWatch, você pode optar por enviar notificações para várias contas da AWS. Por exemplo, para a conta de monitoramento e também para a conta de origem.

   Para que o alarme não envie notificações, escolha Remove (Remover).
10. Para que o alarme execute ações do Auto Scaling, EC2, Lambda ou Systems Manager, escolha o botão apropriado e selecione o estado do alarme e a ação a ser executada. Os alarmes só poderão executar ações do Systems Manager ao entrarem no estado ALARM. Para obter mais informações sobre ações do Systems Manager, consulte Configurar o CloudWatch para criar OpsItems a partir de alarmes e Criação de incidentes.

    > **Note**
    >
    > Para criar um alarme que executa uma ação do SSM Incident Manager, é necessário ter determinadas permissões. Para obter mais informações, consulte Exemplos de políticas baseadas em identidade para o AWSSystems Manager Incident Manager.

11. Quando terminar, escolha Next (Próximo).
12. Digite um nome e uma descrição para o alarme. O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação de markdown, que é exibida somente na guia Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos. Em seguida, escolha Próximo.
13. Em Preview and create (Visualizar e criar), confirme se as informações e condições são o que você deseja e escolha Create alarm (Criar alarme).

Também é possível adicionar alarmes a um painel. Para ter mais informações, consulte Adicionar ou remover um widget de alarme em um painel do CloudWatch .

Criar um alarme do Cloudwatch com base em uma expressão matemática de métrica

Para criar um alarme com base em uma expressão matemática métrica, escolha uma ou mais métricas do CloudWatch a serem usadas na expressão. Depois, especifique a expressão, o limite e os períodos de avaliação.

Não é possível criar um alarme com base na expressão SEARCH. Isso ocorre porque as expressões de pesquisa retornam várias séries temporais, e um alarme baseado em uma expressão matemática pode observar apenas uma série temporal.

Para criar um alarme com base em uma expressão matemática de métrica

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes) e depois escolha All alarms (Todos os alarmes).
3. Selecione Criar alarme.
4. Escolha Select Metric (Selecionar métrica) e, em seguida, execute uma das seguintes ações:
   - Selecione um namespace na lista suspensa Namespaces da AWS ou Namespaces personalizados. Depois de selecionar um namespace, continue escolhendo opções até que uma lista de métricas apareça, na qual você deve marcar a caixa de seleção ao lado da métrica correta.
   - Use a caixa de pesquisa para encontrar uma métrica, uma dimensão ou um ID de recurso. Depois de inserir a métrica, a dimensão ou o ID do recurso, continue escolhendo opções até que uma lista de métricas apareça, na qual você deve marcar a caixa de seleção ao lado da métrica correta.
5. (Opcional) Se quiser adicionar outra métrica a uma expressão matemática de métrica, você poderá usar a caixa de pesquisa para encontrar uma métrica específica. Você pode adicionar até dez métricas a uma expressão matemática de métrica.
6. Selecione a guia Graphed metrics (Representar métricas em gráficos). Para cada uma das métricas que você adicionou anteriormente, execute as seguintes ações:

   a. Na coluna Statistics (Estatística), selecione o menu suspenso. No menu suspenso, escolha uma das estatísticas ou percentis predefinidos. Use a caixa de pesquisa no menu suspenso para especificar um percentil personalizado.

b. Na coluna Period (Período), selecione o menu suspenso. No menu suspenso, escolha um dos períodos de avaliação predefinidos.

Ao criar o alarme, você pode especificar se a legenda do eixo Y é exibida no lado esquerdo ou no lado direito do gráfico.

> **Note**
>
> Quando o CloudWatch avalia alarmes, os períodos são agregados em pontos de dados únicos.

7. Escolha o menu suspenso Add math (Adicionar matemática) e, em seguida, selecione Start with an empty expression (Começar com uma expressão vazia) da lista de expressões matemáticas de métrica predefinidas.

   Depois de escolher Start with an empty expression (Começar com uma expressão vazia), uma caixa de expressão matemática aparecerá para que você possa aplicar ou editar expressões matemáticas.

8. Na caixa de expressão matemática, insira sua expressão matemática e, em seguida, escolha Apply (Aplicar).

   Depois de escolher Apply (Aplicar), uma coluna de ID aparece ao lado da coluna Label (Rotular).

   Para usar uma métrica ou o resultado de outra expressão matemática de métricas como parte da fórmula de sua expressão matemática atual, use o valor que é mostrado na coluna ID. Para alterar o valor de ID, selecione o ícone de caneta e papel ao lado do valor atual. O novo valor deve começar com uma letra minúscula e pode incluir números, letras e o símbolo de sublinhado. Alterar o valor do ID para um nome mais significativo também pode tornar o gráfico do alarme mais fácil de entender.

   Para obter informações sobre as funções disponíveis para matemática de métrica, consulte Sintaxe de funções da matemática métricas.

9. (Opcional) Adicione mais expressões matemáticas usando as métricas e os resultados de outras expressões matemáticas nas fórmulas das novas expressões matemáticas.

10. Quando você tiver a expressão a ser usada no alarme, desmarque as caixas de seleção à esquerda de todas as outras expressões e métricas na página. Somente a caixa de seleção ao lado da expressão a ser usada no alarme deve estar marcada. A expressão escolhida para

o alarme deve produzir uma única série temporal e só mostrar uma linha no gráfico. Depois, escolha Select metric (Selecionar métrica).

A página Specify metric and conditions (Especificar métrica e condições) será exibida, mostrando um gráfico e outras informações sobre a expressão matemática que você selecionou.

11. Em Whenever ***`expression`*** is (Sempre que a expressão for), especifique se a expressão deverá ser maior que, menor que ou igual ao limite. Em than... (que...), especifique o valor limite.
12. Escolha Additional configuration (Configuração adicional). Em Datapoints to alarm (Pontos de dados para alarme), especifique quantos períodos de avaliação (pontos de dados) devem estar no estado `ALARM` para disparar o alarme. Se os dois valores forem correspondentes, você criará um alarme que passa para o estado `ALARM` se esses períodos consecutivos estiverem violando.

    Para criar um alarme M de N, especifique um número menor para o primeiro valor que especificar para o segundo valor. Para ter mais informações, consulte Avaliar um alarme.
13. Para o Missing data treatment (Tratamento de dados ausentes), escolha como deseja que o alarme se comporte quando alguns pontos de dados estiverem ausentes. Para ter mais informações, consulte Configurar como os alarmes do CloudWatch tratam dados ausentes.
14. Escolha Próximo.
15. Em Notification (Notificação), selecione um tópico do SNS para notificar quando o alarme estiver no estado `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

    Para que o alarme envie várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

    Para que o alarme não envie notificações, escolha Remove (Remover).
16. Para que o alarme execute ações do Auto Scaling, EC2, Lambda ou Systems Manager, escolha o botão apropriado e selecione o estado do alarme e a ação a ser executada. Se você escolher uma função do Lambda como uma ação de alarme, especifique o nome da função ou o ARN e, opcionalmente, você poderá escolher uma versão específica da função.

    Os alarmes só poderão executar ações do Systems Manager ao entrarem no estado ALARM. Para obter mais informações sobre ações do Systems Manager, consulte Configurar o CloudWatch para criar OpsItems a partir de alarmes e Criação de incidentes.

> **Note**
>
> Para criar um alarme que executa uma ação do SSM Incident Manager, é necessário ter determinadas permissões. Para obter mais informações, consulte Exemplos de políticas baseadas em identidade para o AWSSystems Manager Incident Manager.

17. Quando terminar, escolha Next (Próximo).
18. Digite um nome e uma descrição para o alarme. Em seguida, escolha Próximo.

    O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação de markdown, que é exibida somente na guia Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.
19. Em Preview and create (Visualizar e criar), confirme se as informações e condições são o que você deseja e escolha Create alarm (Criar alarme).

Também é possível adicionar alarmes a um painel. Para ter mais informações, consulte Adicionar ou remover um widget de alarme em um painel do CloudWatch .

Criar um alarme do CloudWatch baseado em uma consulta ao Metrics Insights

Você pode criar um alarme em qualquer consulta ao Metrics Insights que retorne uma única série temporal. Isso pode ser especialmente útil para criar alarmes dinâmicos que monitorem métricas agregadas em toda uma frota da sua infraestrutura ou em todas as suas aplicações. Crie o alarme uma vez e ele se ajustará à medida que recursos forem adicionados ou removidos da frota. Por exemplo, você pode criar um alarme que monitore a utilização da CPU de todas as suas instâncias, e o alarme se ajustará dinamicamente à medida que você adicionar ou remover instâncias.

Para obter instruções completas, consulte Criar alarmes em consultas ao Metrics Insights.

Criação de um alarme com base em uma fonte de dados conectada

É possível criar alarmes que monitorem métricas de fontes de dados que não estejam no CloudWatch. Para obter mais informações sobre como criar conexões com essas outras fontes de dados, consulte Métricas de consulta de outras fontes de dados.

Como criar um alarme para as métricas de uma fonte de dados à qual você está conectado

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Metrics (Métricas), All metrics (Todas as métricas).
3. Escolha a guia Consulta a várias fontes.
4. Em Fonte de dados, selecione a fonte de dados que deseja usar.
5. O construtor de consultas solicita as informações necessárias para que a consulta recupere as métricas a serem usadas para o alarme. O fluxo de trabalho é diferente para cada fonte de dados e é adaptado à fonte de dados. Por exemplo, para as fontes de dados do Amazon Managed Service for Prometheus e do Prometheus, uma caixa do editor de consultas PromQL com um auxiliar de consulta é exibida.
6. Quando você terminar a estrutura da consulta, escolha Representar consulta graficamente.
7. Se o gráfico de amostra tiver a aparência esperada, escolha Criar alarme.
8. A página Especificar métrica e condições é exibida. Se a consulta que você estiver usando produzir mais de uma série temporal, você visualizará um banner de aviso na parte superior da página. Se isso acontecer, selecione uma função a ser usada para agregar a série temporal em Função de agregação.
9. (Opcional) Adicione um Rótulo para o alarme.
10. Para Sempre que ***`your-metric-name`*** for …, escolha Maior, Maior/Igual, Menor/Igual ou Menor. Em seguida, para então …, especifique um número para o valor limite.
11. Escolha Additional configuration (Configuração adicional). Em Datapoints to alarm (Pontos de dados para alarme), especifique quantos períodos de avaliação (pontos de dados) devem estar no estado `ALARM` para disparar o alarme. Se os dois valores forem correspondentes, você criará um alarme que passa para o estado `ALARM` se esses períodos consecutivos estiverem violando.

    Para criar um alarme M de um alarme N, especifique um número para o primeiro valor que seja menor do que o segundo valor. Para ter mais informações, consulte Avaliar um alarme.
12. Em Missing data treatment (Tratamento de dados ausentes), escolha como deseja que o alarme se comporte quando alguns pontos de dados estiverem ausentes. Para ter mais informações, consulte Configurar como os alarmes do CloudWatch tratam dados ausentes.
13. Escolha Próximo.
14. Em Notificação, especifique um tópico do Amazon SNS a ser notificado quando o alarme transitar entre os estados `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

a. (Opcional) Para enviar várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

> **Note**
>
> Recomendamos configurar o alarme para executar ações quando entrar no estado Dados insuficientes, além de para quando entrar no estado Alarme. Isso ocorre porque muitos problemas com a função do Lambda que se conecta à fonte de dados podem fazer com que o alarme transite para Dados insuficientes.

b. (Opcional) Para não enviar notificações do Amazon SNS, escolha Remover.

15. Para que o alarme execute ações do Auto Scaling, EC2, Lambda ou Systems Manager, escolha o botão apropriado e selecione o estado do alarme e a ação a ser executada. Se você escolher uma função do Lambda como uma ação de alarme, especifique o nome da função ou o ARN e, opcionalmente, você poderá escolher uma versão específica da função.

    Os alarmes só poderão executar ações do Systems Manager ao entrarem no estado ALARM. Para obter mais informações sobre ações do Systems Manager, consulte Configurar o CloudWatch para criar OpsItems a partir de alarmes e Criação de incidentes.

    > **Note**
    >
    > Para criar um alarme que executa uma ação do SSM Incident Manager, é necessário ter determinadas permissões. Para obter mais informações, consulte Exemplos de políticas baseadas em identidade para o AWSSystems Manager Incident Manager.

16. Escolha Próximo.

17. Em Name and description (Nome e descrição), insira um nome e uma descrição para o alarme e selecione Next (Próximo). O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação de markdown, que é exibida somente na guia Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.

> **Tip**
>
> O nome do alarme deve conter somente caracteres UTF-8. Ele não pode conter caracteres de controle ASCII.

18. Em Preview and create (Previsualizar e criar), confirme se as informações e condições do seu alarme estão corretas e escolha Create alarm (Criar alarme).

Detalhes sobre alarmes para fontes de dados conectadas

- Quando o CloudWatch avalia um alarme, ele o faz a cada minuto, mesmo que o período para o alarme seja superior a um minuto. Para que o alarme funcione, a função do Lambda deve ser capaz de retornar uma lista de carimbos de data/hora começando em qualquer minuto, e não somente em múltiplos da duração do período. Esses carimbos de data/hora devem ser espaçados por uma duração do período.

  Portanto, se a fonte de dados consultada pelo Lambda puder retornar somente carimbos de data/hora que sejam múltiplos da duração do período, a função deverá disponibilizar uma "nova amostragem" dos dados buscados para corresponder aos carimbos de data/hora esperados pela solicitação `GetMetricData`.

  Por exemplo, um alarme com um período de cinco minutos é avaliado a cada minuto usando janelas de cinco minutos que mudam um minuto de cada vez. Neste caso:

  - Para a avaliação do alarme às 12:15:00, o CloudWatch espera pontos de dados com carimbos de data/hora de `12:00:00`, `12:05:00` e `12:10:00`.
  - Então, para a avaliação do alarme às 12:16:00, o CloudWatch espera pontos de dados com carimbos de data/hora de `12:01:00`, `12:06:00` e `12:11:00`.
- Quando o CloudWatch avalia um alarme, todos os pontos de dados retornados pela função do Lambda que não estão alinhados com os carimbos de data/hora esperados são descartados, e o alarme é avaliado usando os pontos de dados esperados restantes. Por exemplo, quando o alarme é avaliado às `12:15:00`, ele espera dados com carimbos de data/hora de `12:00:00`, `12:05:00` e `12:10:00`. Se ele receber dados com carimbos de data/hora de `12:00:00`, `12:05:00`, `12:06:00` e `12:10:00`, os dados de `12:06:00` serão descartados e o CloudWatch avaliará o alarme usando os outros carimbos de data/hora.

Então, para a próxima avaliação às `12:16:00`, ele espera dados com carimbos de data/hora de `12:01:00`, `12:06:00` e `12:11:00`. Se tiver somente os dados com carimbos de data/hora de `12:00:00`, `12:05:00` e `12:10:00`, todos esses pontos de dados serão ignorados às 12:16:00, e o alarme realizará a transição para o estado de acordo com a forma como você o especificou para tratar dados ausentes. Para ter mais informações, consulte Avaliar um alarme.

- Recomendamos criar esses alarmes para executar ações quando eles realizarem a transição para o estado `INSUFFICIENT_DATA`, porque diversos casos de uso de falha da função do Lambda realizarão a transição do alarme para `INSUFFICIENT_DATA`, independentemente da forma como você o configurou para tratar dados ausentes.
- Se a função do Lambda retornar um erro ou dados parciais:
  - Se houver um problema de permissão ao chamar a função do Lambda, o alarme começará a realizar transições de dados ausentes de acordo com a forma como você o especificou para tratar dados ausentes quando o criou.
  - Se a função do Lambda retornar `'StatusCode' = 'PartialData'`, a avaliação do alarme falhará e o alarme realizará a transição para `INSUFFICIENT_DATA` após três tentativas. Isso demora cerca de três minutos.
  - Qualquer outro erro proveniente da função do Lambda faz com que o alarme realize a transição para `INSUFFICIENT_DATA`.
- Se a métrica solicitada pela função do Lambda tiver algum atraso de modo que o último ponto de dados esteja sempre ausente, você deverá usar uma solução alternativa. É possível criar um alarme “M out of N” ou aumentar o período de avaliação do alarme. Para obter mais informações sobre alarmes “M out of N”, consulte Avaliar um alarme.

Criar um alarme do CloudWatch com base na detecção de anomalias

E possível criar um alarme com base na detecção de anomalias do CloudWatch, que analisa dados de métrica anteriores e cria um modelo de valores esperados. Os valores esperados levam em conta os padrões típicos por hora, dia e semana na métrica.

Defina um valor para o limite de detecção de anomalias, e o CloudWatch usará esse limite com o modelo para determinar o intervalo “normal” de valores para a métrica. Um valor mais alto para o limite produz uma faixa mais larga de valores "normais".

Você pode escolher se o alarme deve ser acionado quando o valor da métrica estiver acima do segmento de valores esperados, abaixo do segmento ou acima ou abaixo do segmento.

Você também pode criar alarmes de detecção de anomalias em métricas únicas e nas saídas de expressões matemáticas métricas. É possível usar essas expressões para criar gráficos de visualização de bandas de detecção de anomalias.

Em uma conta configurada como uma conta de monitoramento para a observabilidade entre contas do CloudWatch, você pode criar detectores de anomalias em métricas em contas de origem, além de métricas na conta de monitoramento.

Para ter mais informações, consulte Usar a detecção de anomalias do CloudWatch.

> **Note**
>
> Se você já estiver usando a detecção de anomalias para fins de visualização de uma métrica no console de métricas e criar um alarme de detecção de anomalias nessa mesma métrica, o limite que você definiu para o alarme não muda o limite já definido para visualização. Para ter mais informações, consulte Criar um gráfico.

Para criar um alarme com base em detecção de anomalias

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Selecione Criar alarme.
4. Escolha Select metric (Selecionar métrica).
5. Execute um destes procedimentos:
   - Escolha o namespace de serviço que contém sua métrica e, em seguida, continue escolhendo as opções conforme elas aparecerem para limitar suas opções. Quando uma lista de métricas for exibida, marque a caixa de seleção ao lado da sua métrica.
   - Na caixa de pesquisa, digite o nome de uma métrica, dimensão ou ID de recurso. Selecione um dos resultados e continue escolhendo as opções apresentadas até que uma lista de métricas seja exibida. Marque a caixa de seleção ao lado de sua métrica.
6. Escolha Métricas em gráficos.

a. (Opcional) Em Estatística, escolha o menu suspenso e selecione uma das estatísticas ou percentis predefinidos. Você pode usar a caixa de pesquisa no menu suspenso para especificar um percentil personalizado, p. ex., **`p95.45`**.

b. (Opcional) Em Período, escolha o menu suspenso e selecione um dos períodos de avaliação predefinidos.

> **Note**
>
> Quando o CloudWatch avalia seu alarme, ele agrega o período em um só ponto de dados. Para alarmes de detecção de anomalias, o período de avaliação deve ser de um minuto ou mais.

7. Escolha Próximo.
8. Em Conditions (Condições), especifique o seguinte:

a. Selecione Anomaly detection (Detecção de anomalias).

Se o modelo para essa métrica e estatística existir, o CloudWatch exibirá uma pré-visualização da faixa de detecção de anomalias no gráfico que se encontra na parte superior da tela. Após a criação do alarme, pode demorar até 15 minutos para que a faixa de detecção de anomalias real apareça no gráfico. Antes disso, a faixa que você visualiza será uma aproximação da faixa de detecção de anomalias.

> **Tip**
>
> Para visualizar o gráfico na parte superior da tela por um período mais longo, escolha Edit (Editar) no canto superior direito da tela.

Se o modelo para essa métrica e estatística ainda não existir, o CloudWatch gerará a faixa de detecção de anomalias após você concluir a criação do alarme. Para novos modelos, pode demorar até três horas para que a faixa de detecção de anomalias real apareça no gráfico. Pode demorar até duas semanas para que o novo modelo seja treinado, então a faixa de detecção de anomalias mostrará valores esperados mais precisos.

b. Em Whenever ***`metric`*** is (Quando a métrica for), especifique quando acionar o alarme. Por exemplo, quando a métrica for maior que, menor que, ou fora da banda (em qualquer direção).

c. Em Anomaly detection threshold (Limite de detecção de anomalias), escolha o número a ser usado para o limite de detecção de anomalias. Um número mais alto cria uma faixa mais espessa de valores "normais" que é mais tolerante a mudanças na métrica. Um número mais baixo cria uma faixa mais fina que entrará no estado `ALARM` com desvios menores na métrica. Não é necessário que o número seja um número inteiro.

d. Escolha Additional configuration (Configuração adicional). Em Datapoints to alarm (Pontos de dados para alarme), especifique quantos períodos de avaliação (pontos de dados) devem estar no estado `ALARM` para disparar o alarme. Se os dois valores forem correspondentes, você criará um alarme que passa para o estado `ALARM` se esses períodos consecutivos estiverem violando.

   Para criar um alarme M de um alarme N, especifique um número para o primeiro valor que seja menor do que o segundo valor. Para ter mais informações, consulte Avaliar um alarme.

e. Em Missing data treatment (Tratamento de dados ausentes), escolha como deseja que o alarme se comporte quando alguns pontos de dados estiverem ausentes. Para ter mais informações, consulte Configurar como os alarmes do CloudWatch tratam dados ausentes.

f. Se o alarme usar um percentil como estatística monitorada, uma caixa Percentiles with low samples (Percentis com amostras baixas) será exibida. Use-a para escolher se deseja avaliar ou ignorar casos com taxas de amostra baixas. Se você escolher Ignore (maintain the alarm state) (Ignorar (manter o estado do alarme)), o estado do alarme atual será sempre mantido quando o tamanho da amostra for muito baixo. Para ter mais informações, consulte Alarmes do CloudWatch baseados em percentual e exemplos de poucos dados.

9. Escolha Próximo.

10. Em Notification (Notificação), selecione um tópico do SNS para notificar quando o alarme estiver no estado `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

    Para enviar várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

    Escolha Remove (Remover) Se não quiser que o alarme envie notificações.

11. É possível configurar o alarme para executar ações do EC2 ou invocar uma função do Lambda quando ele mudar de estado, ou para criar um OpsItem ou incidente do Systems Manager quando ele entrar no estado ALARM. Para fazer isso, escolha o botão apropriado e, em seguida, escolha o estado do alarme e a ação a ser executada.

Se você escolher uma função do Lambda como uma ação de alarme, especifique o nome da função ou o ARN e, opcionalmente, você poderá escolher uma versão específica da função.

Para obter mais informações sobre ações do Systems Manager, consulte Configurar o CloudWatch para criar OpsItems a partir de alarmes e Criação de incidentes.

> **Note**
>
> Para criar um alarme que executa uma ação do AWS Systems Manager Incident Manager, é necessário ter determinadas permissões. Para obter mais informações, consulte Exemplos de políticas baseadas em identidade para o AWSSystems Manager Incident Manager.

12. Escolha Próximo.
13. Em Name and description (Nome e descrição), insira um nome e uma descrição para o alarme e selecione Next (Próximo). O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação de markdown, que é exibida somente na guia Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.

> **Tip**
>
> O nome do alarme deve conter somente caracteres UTF-8 e não pode conter caracteres de controle ASCII

14. Em Preview and create (Previsualizar e criar), confirme se as informações e condições do seu alarme estão corretas e escolha Create alarm (Criar alarme).

Modificar um modelo de detecção de anomalias

Depois de criar um alarme, você pode ajustar o modelo de detecção de anomalias. É possível excluir determinados períodos de tempo para que não sejam usados na criação do modelo. É fundamental excluir eventos incomuns, como interrupções do sistema, implantações e feriados, dos dados de treinamento. Também é possível especificar se deseja ajustar o modelo para alterações de horário de verão.

Para ajustar o modelo de detecção de anomalias para um alarme

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Escolha o nome do alarme. Se necessário, use a caixa de pesquisa para encontrar o alarme.
4. Escolha Analisar, Em métricas.
5. Na coluna Detalhes, escolha ANOMALY_DETECTION_BAND, Editar modelo de detecção de anomalias.
6. Para excluir um período de tempo de ser usado para produzir o modelo, escolha o ícone de calendário para Data de término. Em seguida, selecione ou insira os dias e horários a serem excluídos do treinamento e escolha Apply (Aplicar).
7. Se a métrica for sensível a alterações no horário de verão, selecione o fuso horário apropriado na caixa Metric timezone (Fuso horário da métrica).
8. Escolha Atualizar.

Excluir um modelo de detecção de anomalias

O uso da detecção de anomalias para um alarme gera cobranças na . Como prática recomendada, se o alarme não precisar mais de um modelo de detecção de anomalias, exclua primeiro o alarme e, em seguida, o modelo. Quando os alarmes de detecção de anomalias são avaliados, quaisquer detectores de anomalias ausentes são criados em seu nome. Se você excluir o modelo sem excluir o alarme, ele automaticamente recriará o modelo.

Como excluir um alarme

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Escolha o nome do alarme.
4. Escolha Ações, Excluir.
5. Na caixa de confirmação, escolha Excluir.

Para excluir um modelo de detecção de anomalias que foi usado para um alarme

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

2. No painel de navegação, escolha Metrics (Métricas) e, em seguida, All metrics (Todas as métricas).
3. Escolha Browse (Navegar) e selecione a métrica que contém o modelo de detecção de anomalias. Você pode buscar sua métrica na caixa de pesquisa ou selecionar a métrica escolhendo entre as opções.
    - (Opcional) Se você estiver usando a interface original, escolha All metrics (Todas as métricas) e escolha a métrica que contém o modelo de detecção de anomalias. Você pode buscar sua métrica na caixa de pesquisa ou selecionar a métrica escolhendo entre as opções.
4. Escolha Graphed metrics (Métricas em gráfico).
5. Na guia Graphed metrics (Métricas em gráficos), escolha o nome do modelo de detecção de anomalias que você deseja remover e escolha Delete anomaly detection model (Excluir modelo de detecção de anomalias).
    - (Opcional) Se estiver usando a interface original, escolha Edit model (Editar modelo). Você será direcionado para uma nova tela. Na nova tela, escolha Delete model (Excluir modelo) e escolha Delete (Excluir).

Alarmes nos logs

As etapas nas seções a seguir explicam como criar alarmes do CloudWatch para logs.

Criar um alarme do CloudWatch com base em um filtro de métrica de grupo de logs

O procedimento contido nesta seção descreve como criar um alarme com base em um filtro de métrica de grupo de logs. Com filtros de métrica, você pode procurar termos e padrões em dados de log à medida que os dados são enviados ao CloudWatch Logs. Para obter mais informações, consulte Criar métricas de eventos de logs usando filtros no Guia do usuário do Amazon CloudWatch Logs. Antes de criar um alarme com base em um filtro de métrica de grupo de logs, é necessário concluir as seguintes ações:

- Criar um grupo de logs do Para obter mais informações, consulte Trabalhar com grupos de logs e fluxos de logs no Guia do usuário do Amazon CloudWatch Logs.
- Crie um filtro de métrica. Para obter mais informações, consulte Criar um filtro de métrica para um grupo de logs no Guia do usuário do Amazon CloudWatch Logs.

Para criar um alarme com base em um filtro de métrica de grupo de logs

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Logs e escolha Log groups (Grupos de logs).
3. Escolha o grupo de logs que contém seu filtro de métrica.
4. Escolha Metric filters (Filtros de métrica).
5. Na guia de filtros de métrica, selecione a caixa do filtro de métrica no qual deseja basear seu alarme.
6. Selecione Criar alarme.
7. (Opcional) Em Metric (Métrica), edite Metric name (Nome da métrica), Statistic (Estatística) e Period (Período).
8. Em Conditions (Condições), especifique o seguinte:

    a. Para Threshold type (Tipo de limite), escolha Static (Estático) ou Anomaly detection (Detecção de anomalias).

    b. Em Whenever ***your-metric-name*** is... (Sempre que o nome da métrica for…), escolha Greater (Maior), Greater/Equal (Maior ou igual a), Lower/Equal (Menor ou igual a) ou Lower (Menor).

    c. Em than... (que…), especifique um número para o valor limite.
9. Escolha Additional configuration (Configuração adicional).

    a. Em Data points to alarm (Pontos de dados para alarme), especifique quantos pontos de dados acionam seu alarme para entrar no estado `ALARM`. Se você especificar valores correspondentes, o alarme passará para o estado `ALARM` se existirem nessa quantidade períodos consecutivos violando. Para criar um alarme M de um alarme N, especifique um número para o primeiro valor que seja menor do que o segundo valor que você especificou. Para obter mais informações, consulte Uso de alarmes do Amazon CloudWatch.

    b. Em Missing data treatment (Tratamento de dados ausentes), selecione uma opção para especificar como tratar dados ausentes quando o alarme for avaliado.
10. Escolha Próximo.
11. Em Notification (Notificação), especifique um tópico do Amazon SNS para notificar quando o alarme estiver no estado `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

    a. (Opcional) Para enviar várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

b. (Opcional) Para não enviar notificações, escolha Remove (Remover).

12. Para que o alarme execute ações do Auto Scaling, EC2, Lambda ou Systems Manager, escolha o botão apropriado e selecione o estado do alarme e a ação a ser executada. Se você escolher uma função do Lambda como uma ação de alarme, especifique o nome da função ou o ARN e, opcionalmente, você poderá escolher uma versão específica da função.

    Os alarmes só poderão executar ações do Systems Manager ao entrarem no estado ALARM. Para obter mais informações sobre ações do Systems Manager, consulte Configurar o CloudWatch para criar OpsItems a partir de alarmes e Criação de incidentes.

    > **Note**
    >
    > Para criar um alarme que executa uma ação do SSM Incident Manager, é necessário ter determinadas permissões. Para obter mais informações, consulte Exemplos de políticas baseadas em identidade para o AWSSystems Manager Incident Manager.

13. Escolha Próximo.
14. Em Name e Description (Nome e Descrição), insira um nome e uma descrição para o seu alarme. O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação de markdown, que é exibida somente na guia Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.
15. Em Preview and create (Pré-visualizar e criar), verifique se sua configuração está correta e escolha Create alarm (Criar alarme).

Combinar alarmes

Com o CloudWatch, você pode combinar vários alarmes em um único alarme composto para criar um indicador de integridade resumido e agregado de toda uma aplicação ou grupo de recursos. Os alarmes compostos são alarmes que determinam seu estado monitorando os estados de outros alarmes. Você define regras para combinar o status desses alarmes monitorados usando a lógica booleana.

Você pode usar alarmes compostos para reduzir o ruído do alarme, tomando medidas apenas em um nível agregado. Por exemplo, você pode criar um alarme composto para enviar uma notificação à sua equipe de servidor Web se algum alarme relacionado ao seu servidor Web for acionado.

Quando qualquer um desses alarmes entra no estado ALARME, o alarme composto entra no estado ALARME e envia uma notificação à sua equipe. Se outros alarmes relacionados ao seu servidor Web também entrarem no estado ALARME, sua equipe não será sobrecarregada com novas notificações, pois o alarme composto já notificou a equipe sobre a situação existente.

Você também pode usar alarmes compostos para criar condições de alarme complexas e realizar ações somente quando muitas condições diferentes forem atendidas. Por exemplo, é possível criar um alarme composto que combine um alarme de CPU e um alarme de memória e que só notificaria a sua equipe se os alarmes de CPU e de memória fossem acionados.

Como usar alarmes compostos

Ao usar alarmes compostos, você tem duas opções:

- Configure as ações que você deseja executar somente no nível de alarme composto e crie os alarmes monitorados subjacentes sem ações.
- Configure um conjunto diferente de ações no nível do alarme composto. Por exemplo, as ações de alarme composto podem envolver uma equipe diferente no caso de um problema generalizado.

Os alarmes compostos podem executar apenas as seguintes ações:

- Notificar tópicos do Amazon SNS
- Invocar funções do Lambda
- Criar OpsItems no centro operacional do Systems Manager
- Criar incidentes no Systems Manager Incident Manager

> **Note**
>
> Todos os alarmes subjacentes em seu alarme composto devem estar na mesma conta e na mesma região que seu alarme composto. No entanto, se você configurar um alarme composto em uma conta de monitoramento da observabilidade entre contas do CloudWatch, os alarmes subjacentes poderão observar métricas em contas de origem diferentes e na própria conta de monitoramento. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.
>
> Um único alarme composto pode monitorar 100 alarmes subjacentes, e 150 alarmes compostos podem monitorar um único alarme subjacente.

Expressões de regra

Todos os alarmes compostos contêm expressões de regras. As expressões de regras informam aos alarmes compostos quais outros alarmes devem ser monitorados e determinam seus estados iniciais. Uma expressão de regra pode se referir a alarmes de métrica e a alarmes compostos. Ao fazer referência a um alarme em uma expressão de regra, você designa uma função para o alarme que determina em qual destes três estados o alarme estará:

- ALARME

  ALARM ("alarm-name ou alarm-ARN") será TRUE se o alarme estiver no estado ALARM (ALARME).
- OK

  OK ("alarm-name ou alarm-ARN") será TRUE se o alarme estiver no estado OK.
- INSUFFICIENT_DATA

  INSUFFICIENT_DATA ("alarm-name ou alarm-ARN") será TRUE se o alarme nomeado estiver no estado INSUFFICIENT_DATA (DADOS_INSUFICIENTES).

> **Note**
>
> TRUE (VERDADEIRO) é sempre avaliado como VERDADEIRO, e FALSE (FALSO) é sempre avaliado como FALSO.

Exemplos de expressões

Como o parâmetro da solicitação `AlarmRule` é compatível com o uso dos operadores lógicos `AND`, `OR` e `NOT`, você pode combinar várias funções em uma única expressão. Os exemplos de expressões a seguir mostram como os alarmes subjacentes podem ser configurados no alarme composto:

- `ALARM(CPUUtilizationTooHigh) AND ALARM(DiskReadOpsTooHigh)`

  A expressão especifica que o alarme composto só passará a `ALARM` se `CPUUtilizationTooHigh` e `DiskReadOpsTooHigh` estiverem no estado `ALARM`.
- `ALARM(CPUUtilizationTooHigh) AND NOT ALARM(DeploymentInProgress)`

A expressão especifica que o alarme composto passará a `ALARM` se `CPUUtilizationTooHigh` estiver no estado `ALARM` e `DeploymentInProgress` não estiver no estado `ALARM`. Este é um exemplo de um alarme composto que reduz o ruído do alarme durante uma janela de implantação.

- `(ALARM(CPUUtilizationTooHigh) OR ALARM(DiskReadOpsTooHigh)) AND OK(NetworkOutTooHigh)`

  A expressão especifica que o alarme composto passará a `ALARM` se `(ALARM(CPUUtilizationTooHigh)` ou `(DiskReadOpsTooHigh)` estiver no estado `ALARM` e `(NetworkOutTooHigh)` estiver no estado `OK`. Este é um exemplo de um alarme composto que reduz o ruído do alarme ao não enviar notificações quando um dos alarmes subjacentes não está no estado `ALARM` durante um problema de rede.

Tópicos

Criar um alarme composto

As etapas desta seção explicam como usar o console do CloudWatch para criar um alarme composto. Você também pode usar a API ou a AWS CLI para criar um alarme composto. Para obter mais informações, consulte PutCompositeAlarm ou put-composite-alarm

Como criar um alarme composto

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes) e depois escolha All alarms (Todos os alarmes).
3. Na lista de alarmes, marque a caixa de seleção ao lado de cada um dos alarmes existentes aos quais você deseja fazer referência na expressão da regra e, em seguida, escolha Create composite alarm (Criar alarme composto).
4. Em Specify composite alarm conditions (Especificar condições de alarme composto), especifique a expressão da regra para o novo alarme composto.

> **Note**
>
> Os alarmes que você selecionou na lista de alarmes são automaticamente listados na caixa Conditions (Condições). Por padrão, a função `ALARM` está designada para cada um de seus alarmes, e todos eles são unidos pelo operador lógico `OR`.

Você pode seguir estas subetapas para modificar a expressão da regra:

a. Você pode alterar o estado obrigatório de cada alarme de `ALARM` para `OK` ou `INSUFFICENT_DATA`.

b. O operador lógico na expressão da regra pode ser alterado de `OR` para `AND` ou `NOT` e você pode adicionar parênteses para agrupar funções.

c. Você pode incluir outros alarmes na expressão da regra ou excluir alarmes dela.

Exemplo: expressão de regra com condições

```
(ALARM("CPUUtilizationTooHigh") OR
ALARM("DiskReadOpsTooHigh")) AND
OK("NetworkOutTooHigh")
```

Nesse exemplo de expressão de regra, o alarme composto passa a `ALARM` quando ALARM(“CPUUtilizationTooHigh” ou ALARM(“DiskReadOpsTooHigh”) está no estado `ALARM` ao mesmo tempo em que OK(“NetworkOutTooHigh”) está em `OK`.

5. Quando terminar, escolha Next (Próximo).
6. Em Configure actions (Configurar ações), você pode escolher uma das seguintes opções:

   Em Notification (Notificação)

   - Selecione um tópico de SNS existente ou escolha Create a new SNS topic (Criar um novo tópico do SNS) ou Use a topic ARN (Usar o ARN do tópico) para definir o tópico do SNS que receberá a notificação.
   - Escolha Add notification (Adicionar notificação) para que o alarme possa enviar várias notificações para o mesmo estado ou para estados diferentes.
   - Escolha Remove (Remover) para que o alarme pare de enviar notificações ou executar ações.

(Opcional) Para que o alarme invoque uma função do Lambda quando mudar de estado, escolha Adicionar ação do Lambda. Em seguida, especifique o nome da função ou o ARN e, opcionalmente, escolha uma versão específica da função.

Para Systems Manager action (Ação do Systems Manager)

- Escolha Add Systems Manager action (Adicionar ação do Systems Manager) para que o alarme possa executar uma ação do SSM ao entrar no estado ALARM.

Para saber mais sobre ações do Systems Manager, consulte Configurar o CloudWatch para criar OpsItems de alarmes no Guia do usuário do AWS Systems Manager e Incident creation (Criação de incidentes) no Guia do usuário do Incident Manager. Para criar um alarme que realiza uma ação do SSM Incident Manager, você precisa ter as permissões corretas. Para obter mais informações, consulte Exemplos de políticas baseadas em identidade para o AWS Incident Manager do Systems Manager no Guia do usuário do Incident Manager.

7. Quando terminar, escolha Next (Próximo).
8. Em Add name and description (Adicionar nome e descrição), insira o nome do alarme e uma descrição opcional para seu novo alarme composto. O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação de markdown, que é exibida somente na guia Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.
9. Quando terminar, escolha Next (Próximo).
10. Em Preview and create (Visualizar e criar), confirme as informações e, em seguida, escolha Create composite alarm (Criar alarme composto).

> **Note**
>
> Você pode criar um ciclo de alarmes compostos com dois alarmes compostos que dependem um do outro. Se isso acontecer, os alarmes compostos deixarão de ser avaliados e não poderão ser excluídos por causa da dependência mútua. A maneira mais fácil de quebrar um ciclo de dependência entre alarmes compostos é alterar a função `AlarmRule` em um deles para `False`.

Como suprimir ações de alarme composto

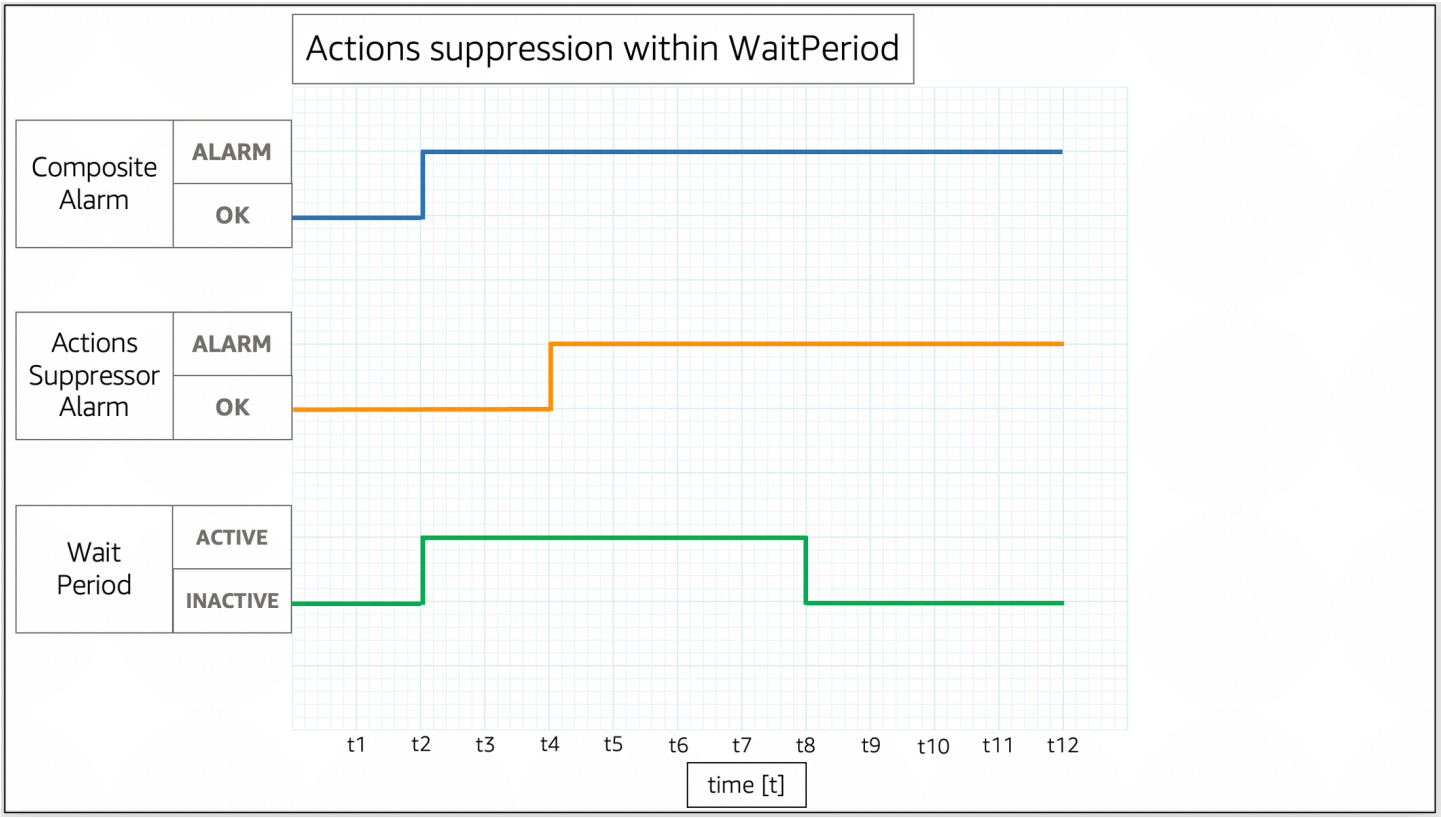
Como os alarmes compostos permitem que você tenha uma visão agregada da sua integridade em vários alarmes, há situações comuns em que é esperado que esses alarmes sejam acionados. Por exemplo, durante uma janela de manutenção da sua aplicação ou quando você investiga um incidente em andamento. Nessas situações, talvez você queira suprimir as ações de seus alarmes compostos para evitar notificações indesejadas ou a criação de novos tíquetes de incidentes.

Com a supressão da ação de alarme composto, você define um alarme como supressor. Os alarmes supressores impedem que os alarmes compostos realizem ações. Por exemplo, você pode especificar um alarme supressor que represente o status de um recurso de suporte. Se o recurso de suporte estiver inativo, o alarme supressor impedirá que o alarme composto envie notificações. A supressão da ação do alarme composto ajuda a reduzir o ruído do alarme. Assim, você leva menos tempo gerenciando alarmes e mais tempo se concentrando em suas operações.

Você especifica alarmes supressores ao configurar alarmes compostos. Qualquer alarme pode funcionar como um alarme supressor. Quando o estado do alarme supressor muda de `OK` para `ALARM`, o alarme composto para de realizar ações. Quando o estado do alarme supressor muda de `ALARM` para `OK`, o alarme composto volta a realizar ações.

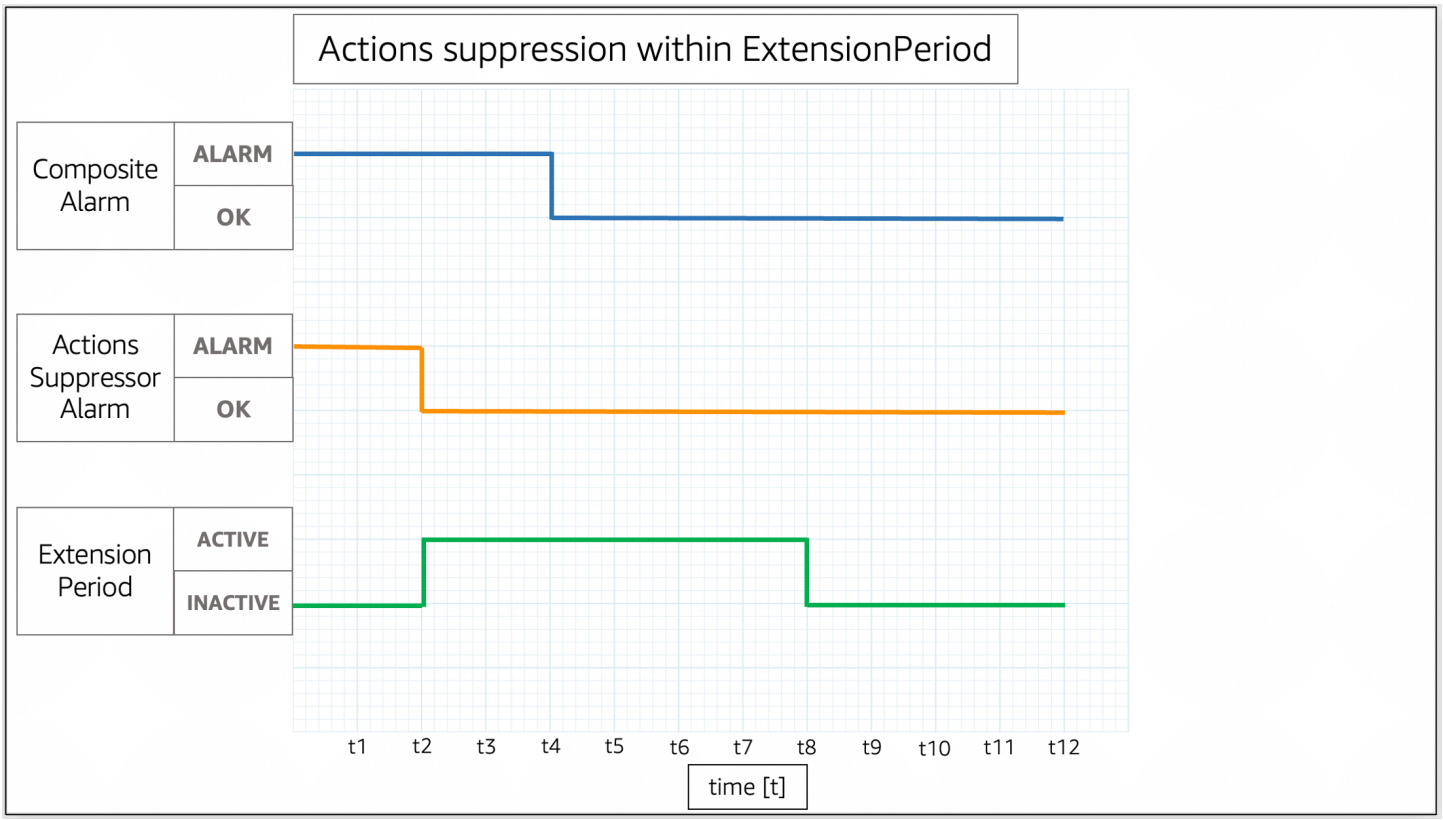
`WaitPeriod` e **`ExtensionPeriod`**

Ao especificar um alarme supressor, você define os parâmetros `WaitPeriod` e `ExtensionPeriod`. Esses parâmetros evitam que alarmes compostos realizem ações inesperadamente quando os alarmes supressores mudam de estado. Use o parâmetro `WaitPeriod` para compensar qualquer atraso que possa ocorrer quando um alarme supressor muda de `OK` para `ALARM`. Por exemplo, se um alarme supressor mudar de `OK` para `ALARM` no período de 60 segundos, defina `WaitPeriod` como 60 segundos.



Na imagem, o alarme composto muda de OK para ALARM em t2. Um `WaitPeriod` começa em t2 e termina em t8. Isso dá ao alarme supressor tempo para mudar o estado de OK para ALARM em t4 antes de suprimir as ações do alarme composto quando o `WaitPeriod` termina em t8.

Use o parâmetro `ExtensionPeriod` para compensar qualquer atraso que possa ocorrer quando um alarme composto muda para OK depois que um alarme supressor muda para OK. Por exemplo, se um alarme composto mudar para OK no período de 60 segundos após a mudança de um alarme supressor para OK, defina `ExtensionPeriod` como 60 segundos.



Na imagem, o alarme supressor muda de `ALARM` para `OK` em t2. Um `ExtensionPeriod` começa em t2 e termina em t8. Isso dá ao alarme composto tempo para mudar de `ALARM` para `OK` antes do final do `ExtensionPeriod` em t8.

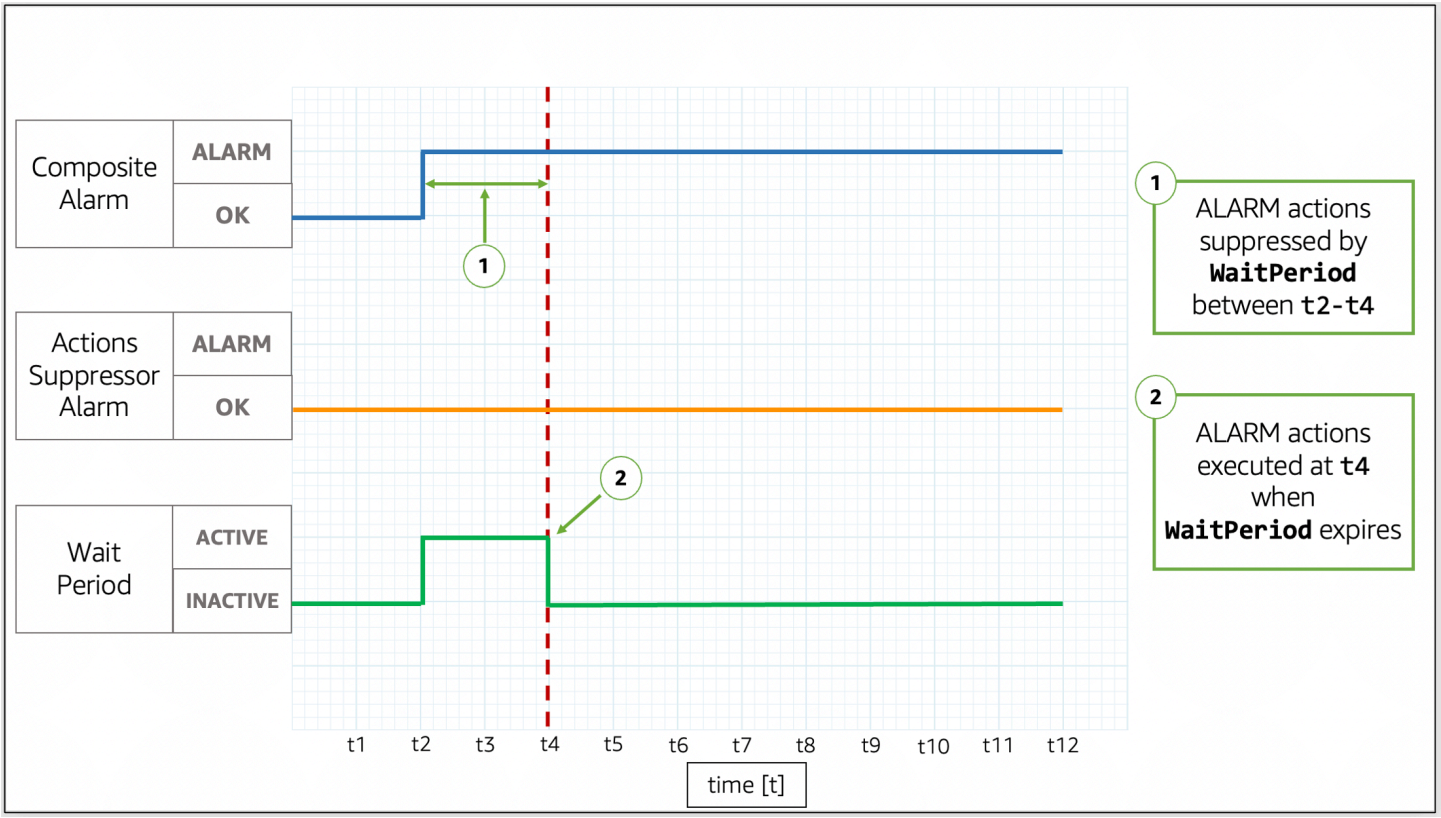
Alarmes compostos não realizam ações quando `WaitPeriod` e `ExtensionPeriod` se tornam ativos. Os alarmes compostos realizam ações baseadas em seus estados atuais quando `ExtensionPeriod` e `WaitPeriod` se tornam inativos. Recomendamos que você defina o valor de cada parâmetro como 60 segundos, pois o CloudWatch avalia os alarmes de métricas a cada minuto. Você pode definir os parâmetros como qualquer número inteiro em segundos.

Os exemplos a seguir descrevem mais detalhadamente como `WaitPeriod` e `ExtensionPeriod` impedem que alarmes compostos realizem ações inesperadas.

> **Note**
>
> Nos exemplos a seguir, `WaitPeriod` está configurado como 2 unidades de tempo e `ExtensionPeriod` está configurado como 3 unidades de tempo.

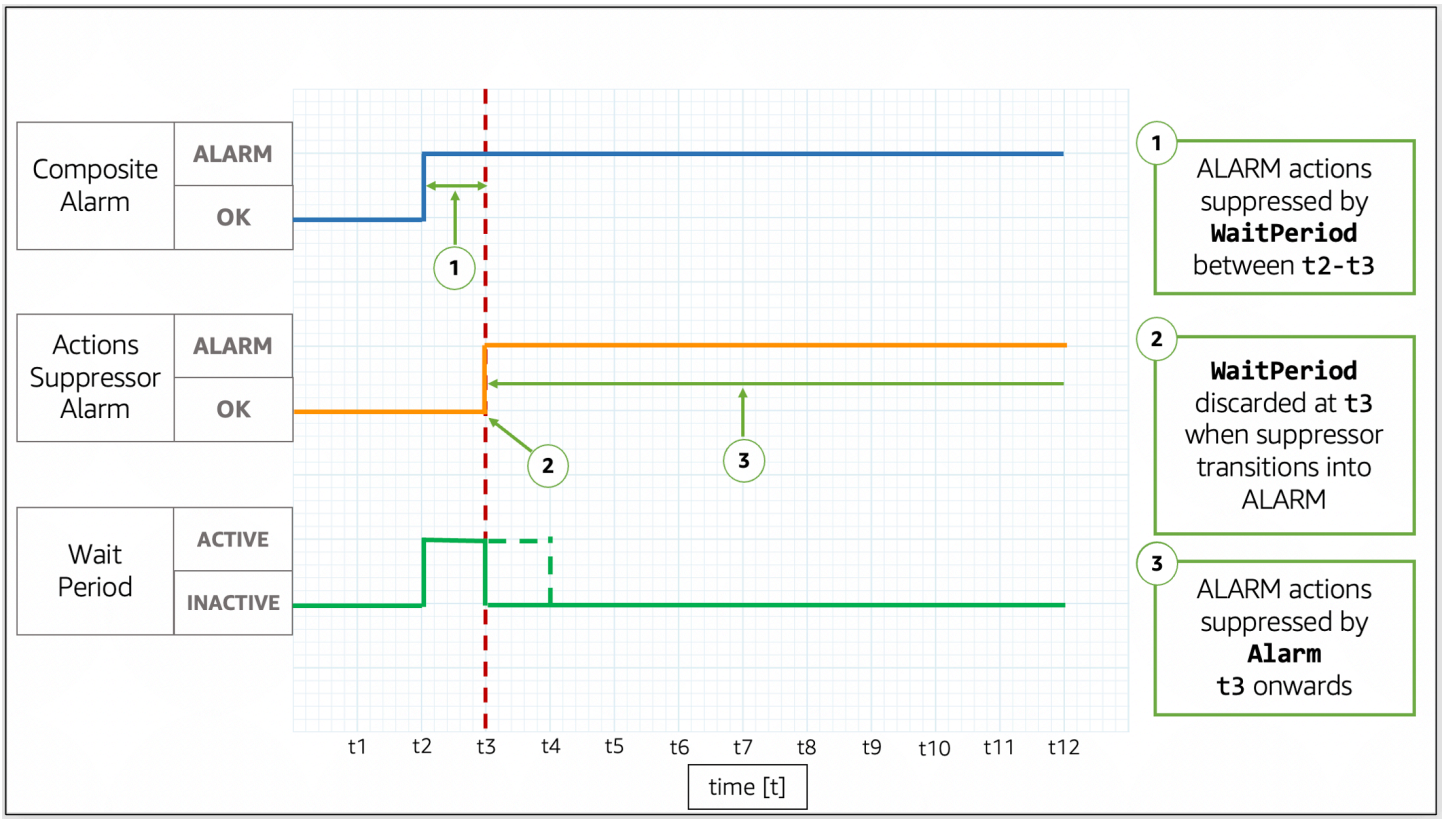
Exemplos

Exemplo 1: as ações não são suprimidas após o **`WaitPeriod`**



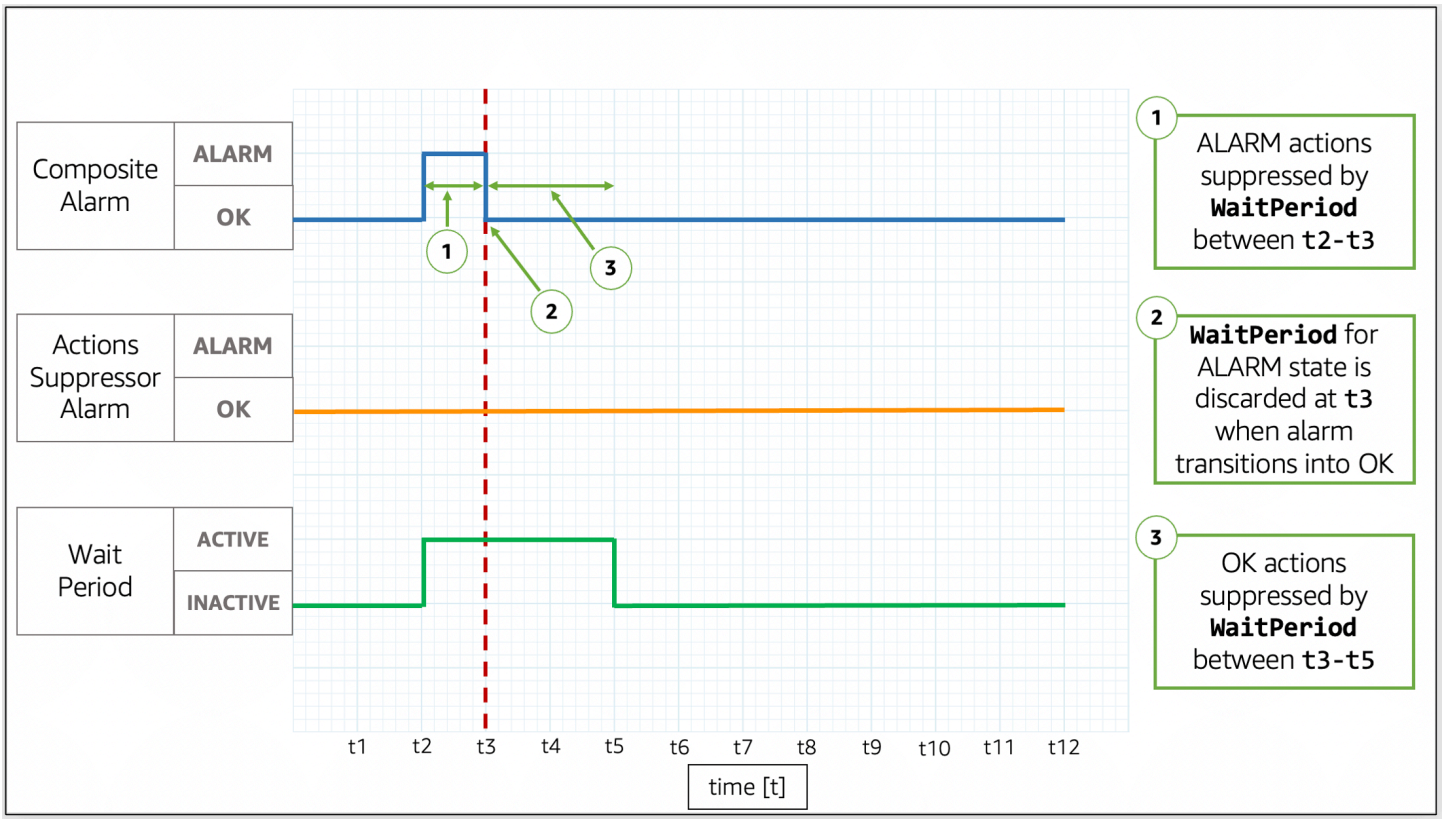
Na imagem, o alarme composto muda do estado de OK para ALARM em t2. Um `WaitPeriod` começa em t2 e termina em t4, para evitar que o alarme composto realize uma ação. Depois que o `WaitPeriod` termina em t4, o alarme composto realiza as ações porque o alarme supressor ainda está no estado OK.

Exemplo 2: as ações são suprimidas pelo alarme antes no final do **`WaitPeriod`**



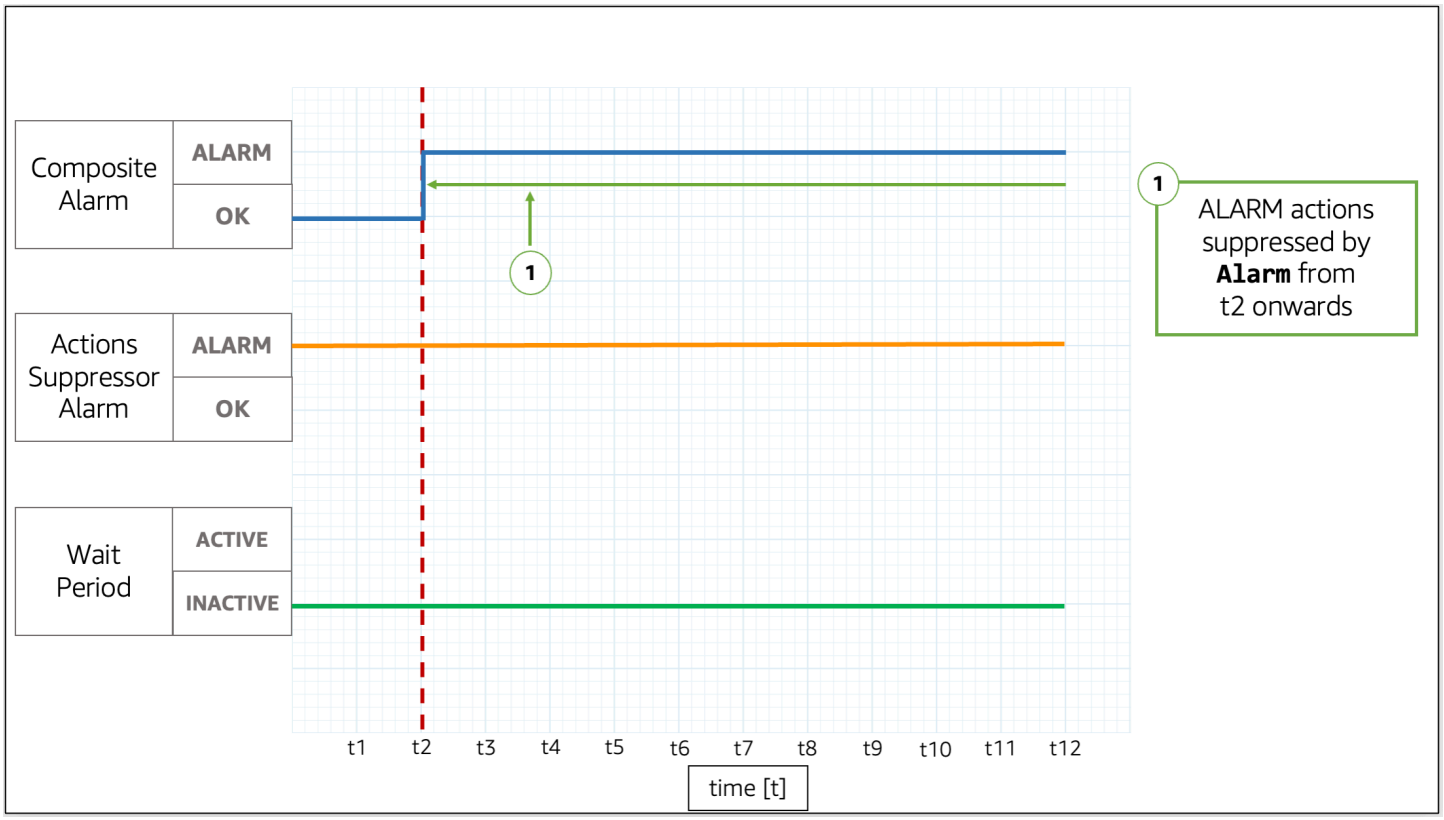
Na imagem, o alarme composto muda do estado de OK para ALARM em t2. Um `WaitPeriod` começa em t2 e termina em t4. Isso dá ao alarme supressor tempo para mudar o estado de OK para ALARM em t3. Como o alarme supressor muda do estado de OK para ALARM em t3, o `WaitPeriod` que começou em t2 é descartado e o alarme supressor agora impede que o alarme composto realize ações.

Exemplo 3: transição de estado quando as ações são suprimidas pelo **`WaitPeriod`**



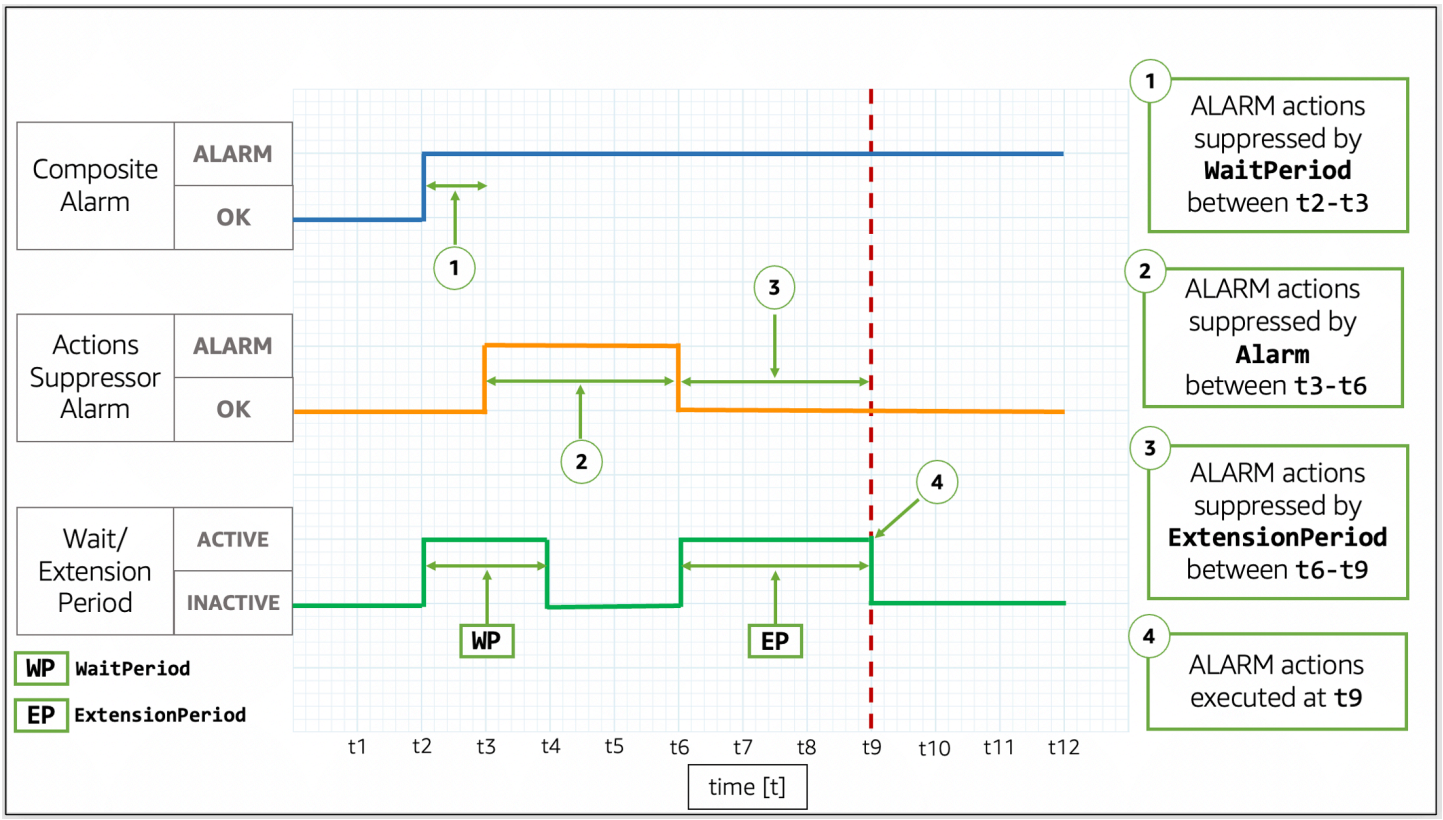
Na imagem, o alarme composto muda do estado de `OK` para `ALARM` em t2. Um `WaitPeriod` começa em t2 e termina em t4. Isso dá ao alarme supressor tempo para mudar de estado. O alarme composto volta para `OK` em t3 e o `WaitPeriod` que começou em t2 é descartado. Um novo `WaitPeriod` começa em t3 e termina em t5. Quando o novo `WaitPeriod` termina em t5, o alarme composto realiza as ações.

Exemplo 4: transição de estado quando as ações são suprimidas pelo alarme



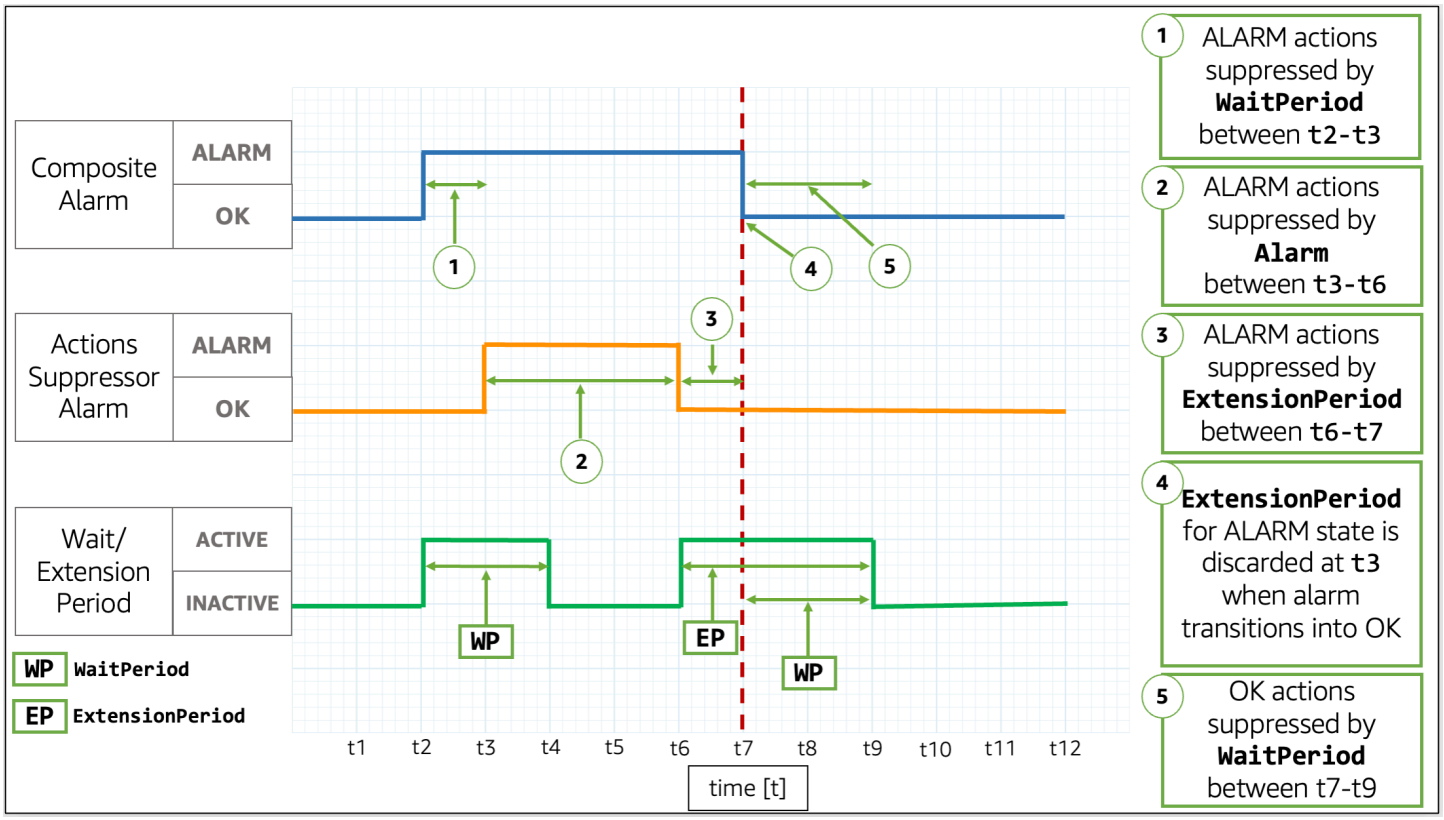
Na imagem, o alarme composto muda do estado de OK para ALARM em t2. O alarme supressor já está no estado ALARM. O alarme supressor impede que o alarme composto realize ações.

Exemplo 5: as ações não são suprimidas após o **`ExtensionPeriod`**



Na imagem, o alarme composto muda do estado de OK para ALARM em t2. Um `WaitPeriod` começa em t2 e termina em t4. Isso dá ao alarme supressor tempo para mudar o estado de OK para ALARM em t3 antes de suprimir as ações do alarme composto até t6. Como o alarme supressor muda do estado de OK para ALARM em t3, o `WaitPeriod` que começou em t2 é descartado. Em t6, o alarme supressor muda para OK. Um `ExtensionPeriod` começa em t6 e termina em t9. Depois que o `ExtensionPeriod` termina, o alarme composto realiza as ações.

Exemplo 6: transição de estado quando as ações são suprimidas pelo **`ExtensionPeriod`**



Na imagem, o alarme composto muda do estado de OK para ALARM em t2. Um `WaitPeriod` começa em t2 e termina em t4. Isso dá ao alarme supressor tempo para mudar o estado de OK para ALARM em t3 antes de suprimir as ações do alarme composto até t6. Como o alarme supressor muda do estado de OK para ALARM em t3, o `WaitPeriod` que começou em t2 é descartado. Em t6, o alarme supressor volta para OK. Um `ExtensionPeriod` começa em t6 e termina em t9. Quando o alarme composto volta para OK em t7, o `ExtensionPeriod` é descartado e um novo `WaitPeriod` começa em t7 e termina em t9.

> **Tip**
>
> Se você substituir o alarme supressor de ação, qualquer `WaitPeriod` ou `ExtensionPeriod` ativo será descartado.

Atuação em mudanças de alarmes

O CloudWatch pode notificar os usuários sobre dois tipos de alterações de alarme: quando um alarme muda de estado e quando a configuração de um alarme é atualizada.

Quando um alarme é avaliado, ele pode mudar de um estado para outro, como ALARM, OK ou INSUFFICIENT_DATA. Essas mudanças no estado do alarme podem sinalizar um possível incidente, um retorno ao normal ou uma métrica indisponível. Nesses casos, talvez você queira engajar ou notificar os usuários usando uma das opções a seguir:

- É possível configurar o alarme para enviar uma notificação para um tópico do SNS como parte das ações de alarme. Um tópico do SNS pode ser configurado para mensagens de aplicação para aplicação (A2A), bem como para notificações de aplicação para pessoa (A2P), incluindo canais como notificações por e-mail e SMS. Todos os destinos definidos para o seu tópico do SNS recebem a notificação de alarme. Para obter mais informações, consulte destinos de eventos do Amazon SNS.
- É possível configurar notificações para eventos de alteração do estado do alarme. AWS As notificações do usuário oferecem uma forma nativa de configurar essas notificações e são a abordagem recomendada.

Além disso, o CloudWatch envia eventos ao Amazon EventBridge sempre que um alarme do CloudWatch muda de estado, e quando os alarmes são criados, atualizados, excluídos ou alterados. É possível escrever regras do EventBridge para realizar ações ou ser notificado quando o EventBridge receber esses eventos.

Tópicos

Notificação dos usuários sobre mudanças de alarmes

Esta seção explica como é possível usar as Notificações de Usuários da AWS ou o Amazon Simple Notification Service para que os usuários sejam notificados sobre alterações no alarme.

Configurando notificações de usuário da AWS

É possível usar notificações de usuário da AWS para configurar canais de entrega para receber notificações sobre eventos de alteração de estado de alarme da e alteração de configuração do CloudWatch. Você recebe uma notificação quando um evento corresponde a uma regra especificada. É possível receber notificações de eventos por meio de vários canais, incluindo email, notificações de chat do Chatbot da AWS ou Notificações por push da aplicação móvel do Console

da AWS. Também é possível ver as notificações na Central de notificações do console. é compatível com agregação, o que pode reduzir o número de notificações recebidas durante eventos específicos.

As configurações de notificação que você cria com as Notificações de Usuários da AWS não contam para o limite do número de ações que podem ser configuradas por estado de alarme alvo. Como as Notificações de Usuários da AWS correspondem aos eventos emitidos para o Amazon EventBridge, elas enviam notificações para todos os alarmes em sua conta e regiões selecionadas, a menos que você especifique um filtro avançado para permitir ou negar alarmes ou padrões específicos.

O exemplo a seguir de um filtro avançado corresponde a uma alteração do estado do alarme de OK para ALARM no alarme chamado `ServerCpuTooHigh`.

```
{
"detail": {
    "alarmName": ["ServerCpuTooHigh"],
    "previousState": { "value": ["OK"] },
    "state": { "value": ["ALARM"] }
  }
}
```

É possível usar qualquer uma das propriedades publicadas por um alarme nos eventos do EventBridge para criar um filtro. Para ter mais informações, consulte Eventos de alarme e o EventBridge.

Configurar notificações do Amazon SNS

É possível usar o Amazon Simple Notification Service para enviar mensagens de aplicação para aplicação (A2A) e mensagens de aplicação para pessoa (A2P), mensagens de texto para celulares (SMS) e mensagens de email. Para obter mais informações, consulte destinos de eventos do Amazon SNS.

Para cada estado que um alarme pode assumir, é possível configurar o alarme para enviar uma mensagem para um tópico do SNS. Cada tópico do Amazon SNS que você configurar para um estado em um determinado alarme contará para o limite do número de ações que poderão ser configuradas para esse alarme e estado. É possível enviar mensagens para o mesmo tópico do Amazon SNS a partir de qualquer alarme em sua conta e usar o mesmo tópico do Amazon SNS para consumidores de aplicações (A2A) e pessoais (A2P). Como essa configuração é feita no nível do alarme, somente os alarmes que você configurou enviam mensagens para o tópico selecionado do Amazon SNS.

Primeiro, crie um tópico e inscreva-se nele. Você também pode publicar uma mensagem de teste para o tópico. Para ver um exemplo, consulte Configurar um tópico do Amazon SNS usando o AWS Management Console. Ou, para obter mais informações, consulte Conceitos básicos do Amazon SNS.

Se preferir, caso você planeje usar o AWS Management Console para criar seu alarme do CloudWatch, poderá ignorar esse procedimento, pois o tópico poderá ser criado junto com o alarme.

Ao criar um alarme do CloudWatch, será possível adicionar ações para qualquer estado de destino em que o alarme entre. Adicione uma notificação do Amazon SNS para o estado sobre o qual você deseja ser notificado e selecione o tópico do Amazon SNS que você criou na etapa anterior para enviar uma notificação por email quando o alarme entrar no estado selecionado.

> **Note**
>
> Ao criar um tópico do Amazon SNS, você pode escolher torná-lo um tópico padrão ou um tópico FIFO. O CloudWatch garante a publicação de todas as notificações de alarme para ambos os tipos de tópicos. No entanto, mesmo que você use um tópico FIFO, em alguns casos raros, o CloudWatch envia as notificações fora de ordem para o tópico. Se você usar um tópico FIFO, o alarme configura o ID do grupo de mensagens das notificações de alarme como um hash do ARN do alarme.

Evitar problemas de representante confuso

Para evitar problemas de segurança de representante confuso entre serviços, recomendamos o uso das chaves de condição globais `aws:SourceArn` e `aws:SourceAccount` na política de recursos do Amazon SNS, que concede permissão ao CloudWatch para acessar os seus recursos do Amazon SNS.

O exemplo de política de recursos a seguir usa a chave de condição `aws:SourceArn` para restringir a permissão `SNS:Publish` de forma que ela seja utilizada apenas por alarmes do CloudWatch na conta definida.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "cloudwatch.amazonaws.com"
```

```
        },
        "Action": "SNS:Publish",
        "Resource": "arn:aws:sns:us-east-2:444455556666:MyTopic",
        "Condition": {
            "ArnLike": {
                "aws:SourceArn": "arn:aws:cloudwatch:us-east-2:111122223333:alarm:*"
            },
            "StringEquals": {
                "aws:SourceAccount": "111122223333"
            }
        }
    }]
}
```

Se um ARN de alarme contiver caracteres não ASCII, utilize somente a chave de condição global `aws:SourceAccount` para limitar as permissões.

Configurar um tópico do Amazon SNS usando o AWS Management Console

Primeiro, crie um tópico e inscreva-se nele. Você também pode publicar uma mensagem de teste para o tópico.

Para criar um tópico do SNS

1. Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No painel do Amazon SNS, em Common actions (Ações comuns), escolha Create Topic (Criar tópico).
3. Na caixa de diálogo Create new topic (Criar novo tópico), em Topic name (Nome do tópico), insira um nome para o tópico (por exemplo, **`my-topic`**).
4. Escolha Criar tópico.
5. Copie o Topic ARN (ARN do tópico) para a próxima tarefa (por exemplo, arn:aws:sns:us-east-1:111122223333:my-topic).

Para se inscrever em um tópico do SNS

1. Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No painel de navegação, escolha Assinaturas, Criar assinatura.
3. Na caixa de diálogo Criar assinatura, em ARN do tópico, cole o ARN do tópico que você criou na tarefa anterior.

4. Em Protocolo, escolha Email.
5. Em Endpoint, insira um endereço de e-mail para receber a notificação e escolha Create subscription (Criar inscrição).
6. No aplicativo de e-mail, abra a mensagem de notificações da AWS e confirme a inscrição.

   O navegador da Web exibe uma resposta de confirmação do Amazon SNS.

Para publicar uma mensagem de teste em um tópico do SNS

1. Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No painel de navegação, escolha Tópicos.
3. Na página Topics (Tópicos), selecione um tópico e escolha Publish to topic (Publicar em um tópico).
4. Na página Publish a message (Publicar uma mensagem), em Subject (Assunto), digite uma linha de assunto para a mensagem e em Message (Mensagem), digite uma breve mensagem.
5. Escolha Publish Message (Publicar mensagem).
6. Verifique seu e-mail para confirmar que recebeu a mensagem.

Configurar um tópico do SNS usando a AWS CLI

Primeiro você cria um tópico do SNS e, depois, publica uma mensagem diretamente no tópico para verificar se ele foi configurado corretamente.

Para configurar um tópico do SNS

1. Crie o tópico usando o comando create-topic da forma a seguir.

   ```
   aws sns create-topic --name my-topic
   ```

   O Amazon SNS retorna um ARN do tópico com o seguinte formato:

   ```
   {
       "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic"
   }
   ```

2. Assine o seu endereço de e-mail para o tópico usando o comando subscribe. Se a solicitação de assinatura for bem-sucedida, você receberá uma mensagem de e-mail de confirmação.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:111122223333:my-topic --
protocol email --notification-endpoint my-email-address
```

O Amazon SNS retorna o seguinte:

```
{
    "SubscriptionArn": "pending confirmation"
}
```

3. No aplicativo de e-mail, abra a mensagem de notificações da AWS e confirme a inscrição.

   O navegador da Web exibe uma resposta de confirmação do Amazon Simple Notification Service.

4. Verifique a assinatura usando o comando list-subscriptions-by-topic.

```
aws sns list-subscriptions-by-topic --topic-arn arn:aws:sns:us-
east-1:111122223333:my-topic
```

O Amazon SNS retorna o seguinte:

```
{
  "Subscriptions": [
    {
        "Owner": "111122223333",
        "Endpoint": "me@mycompany.com",
        "Protocol": "email",
        "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic",
        "SubscriptionArn": "arn:aws:sns:us-east-1:111122223333:my-topic:64886986-
bf10-48fb-a2f1-dab033aa67a3"
    }
  ]
}
```

5. (Opcional) Publique uma mensagem de teste no tópico usando o comando publish.

```
aws sns publish --message "Verification" --topic arn:aws:sns:us-
east-1:111122223333:my-topic
```

O Amazon SNS retorna os resultados a seguir.

```
{
    "MessageId": "42f189a0-3094-5cf6-8fd7-c2dde61a4d7d"
}
```

6. Verifique seu e-mail para confirmar que recebeu a mensagem.

Eventos de alarme e o EventBridge

O CloudWatch envia eventos ao Amazon EventBridge sempre que um alarme do CloudWatch é criado, atualizado, excluído ou altera o estado de um alarme. É possível usar o EventBridge e esses eventos para gravar regras que executam ações, como notificá-lo, quando um alarme mudar de estado. Para obter mais informações, consulte O que é o Amazon EventBridge?

O CloudWatch garante a entrega de eventos de alteração de estado de alarme ao EventBridge.

Exemplos de eventos do CloudWatch

Esta seção inclui exemplos de eventos do CloudWatch.

Alteração de estado do alarme de uma única métrica

```
{
    "version": "0",
    "id": "c4c1c1c9-6542-e61b-6ef0-8c4d36933a92",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:04:40Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
    ],
    "detail": {
        "alarmName": "ServerCpuTooHigh",
        "configuration": {
            "description": "Goes into alarm when server CPU utilization is too high!",
            "metrics": [
                {
                    "id": "30b6c6b2-a864-43a2-4877-c09a1afc3b87",
                    "metricStat": {
                        "metric": {
```

```
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "CPUUtilization",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ]
        },
        "previousState": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [0.0666851903306472 (01/10/19 13:46:00)] was not greater than the threshold (50.0)
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-01T13:56:40.985+0000\",\"startDate\":\"2019-10-01T13:46:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[0.0666851903306472],
\"threshold\":50.0}",
            "timestamp": "2019-10-01T13:56:40.987+0000",
            "value": "OK"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [99.50160229693434 (02/10/19 16:59:00)] was greater than the threshold (50.0) (minimum
 1 datapoint for OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:04:40.985+0000\",\"startDate\":\"2019-10-02T16:59:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[99.50160229693434],
\"threshold\":50.0}",
            "timestamp": "2019-10-02T17:04:40.989+0000",
            "value": "ALARM"
        }
    }
}
```

Alteração de estado de um alarme de matemática de métricas

```
{
    "version": "0",
    "id": "2dde0eb1-528b-d2d5-9ca6-6d590caf2329",
```

```
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:20:48Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "configuration": {
            "description": "Goes into alarm if total network traffic exceeds 10Kb",
            "metrics": [
                {
                    "expression": "SUM(METRICS())",
                    "id": "e1",
                    "label": "Total Network Traffic",
                    "returnData": true
                },
                {
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkIn",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "m2",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkOut",
                            "namespace": "AWS/EC2"
                        },
```

```
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                }
            ]
        },
        "previousState": {
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2019-10-02T17:20:03.642+0000",
            "value": "INSUFFICIENT_DATA"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints [45628.0
 (02/10/19 17:10:00)] was greater than the threshold (10000.0) (minimum 1 datapoint for
 OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:20:48.551+0000\",\"startDate\":\"2019-10-02T17:10:00.000+0000\",
\"period\":300,\"recentDatapoints\":[45628.0],\"threshold\":10000.0}",
            "timestamp": "2019-10-02T17:20:48.554+0000",
            "value": "ALARM"
        }
    }
}
```

Alteração de estado de um alarme de detecção de anomalias

```
{
    "version": "0",
    "id": "daafc9f1-bddd-c6c9-83af-74971fcfc4ef",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-03T16:00:04Z",
    "region": "us-east-1",
    "resources": ["arn:aws:cloudwatch:us-east-1:123456789012:alarm:EC2 CPU Utilization
 Anomaly"],
    "detail": {
        "alarmName": "EC2 CPU Utilization Anomaly",
        "state": {
            "value": "ALARM",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints [0.0
 (03/10/19 15:58:00)] was less than the lower thresholds [0.020599444741798756] or
```

```
 greater than the upper thresholds [0.3006915352732461] (minimum 1 datapoint for OK ->
 ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T16:00:04.650+0000\",\"startDate\":\"2019-10-03T15:58:00.000+0000\",
\"period\":60,\"recentDatapoints\":[0.0],\"recentLowerThresholds\":
[0.020599444741798756],\"recentUpperThresholds\":[0.3006915352732461]}",
            "timestamp": "2019-10-03T16:00:04.653+0000"
        },
        "previousState": {
            "value": "OK",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints
 [0.166666666664241 (03/10/19 15:57:00)] was not less than the lower thresholds
 [0.0206719426210418] or not greater than the upper thresholds [0.30076870222143803]
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T15:59:04.670+0000\",\"startDate\":\"2019-10-03T15:57:00.000+0000\",
\"period\":60,\"recentDatapoints\":[0.166666666664241],\"recentLowerThresholds\":
[0.0206719426210418],\"recentUpperThresholds\":[0.30076870222143803]}",
            "timestamp": "2019-10-03T15:59:04.672+0000"
        },
        "configuration": {
            "description": "Goes into alarm if CPU Utilization is out of band",
            "metrics": [{
                "id": "m1",
                "metricStat": {
                    "metric": {
                        "namespace": "AWS/EC2",
                        "name": "CPUUtilization",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 60,
                    "stat": "Average"
                },
                "returnData": true
            }, {
                "id": "ad1",
                "expression": "ANOMALY_DETECTION_BAND(m1, 0.8)",
                "label": "CPUUtilization (expected)",
                "returnData": true
            }]
        }
    }
```

```
}
```

Alteração de estado de um alarme composto com um alarme supressor

```
{
    "version": "0",
    "id": "d3dfc86d-384d-24c8-0345-9f7986db0b80",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-22T15:57:45Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "actionsSuppressedReason": "Actions suppressed by WaitPeriod",
            "value": "ALARM",
            "reason": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:SuppressionDemo.EventBridge.FirstChild transitioned to ALARM
 at Friday 22 July, 2022 15:57:45 UTC",
            "reasonData": "{\"triggeringAlarms\":[{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:ServerCpuTooHigh\",\"state\":{\"value\":\"ALARM\",\"timestamp
\":\"2022-07-22T15:57:45.394+0000\"}}]}",
            "timestamp": "2022-07-22T15:57:45.394+0000"
        },
        "previousState": {
            "value": "OK",
            "reason": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:SuppressionDemo.EventBridge.Main was created and its alarm
 rule evaluates to OK",
            "reasonData": "{\"triggeringAlarms\":[{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:TotalNetworkTrafficTooHigh\",\"state\":{\"value\":\"OK\",
\"timestamp\":\"2022-07-14T16:28:57.770+0000\"}},{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:ServerCpuTooHigh\",\"state\":{\"value\":\"OK\",\"timestamp\":
\"2022-07-14T16:28:54.191+0000\"}}]}",
            "timestamp": "2022-07-22T15:56:14.552+0000"
        },
        "configuration": {
```

```
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180
        }
    }
}
```

Criação de um alarme composto

```
{

    "version": "0",
    "id": "91535fdd-1e9c-849d-624b-9a9f2b1d09d0",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:06:22Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "create",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:22.289+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "alarmName": "ServiceAggregatedAlarm",
            "description": "Aggregated monitor for instance",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:22.289+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Criação de um alarme composto com um alarme supressor

```
{
    "version": "0",
    "id": "454773e1-09f7-945b-aa2c-590af1c3f8e0",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T13:59:46Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "create",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:46.425+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Atualização de uma métrica de alarme

```
{

    "version": "0",
    "id": "bc7d3391-47f8-ae47-f457-1b4d06118d50",
    "detail-type": "CloudWatch Alarm Configuration Change",
```

```
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:06:34Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
    ],
    "detail": {
        "alarmName": "ServerCpuTooHigh",
        "operation": "update",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:13.757+0000"
        },
        "configuration": {
            "evaluationPeriods": 1,
            "threshold": 80,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [
                {
                    "id": "86bfa85f-b14c-ebf7-8916-7da014ce23c0",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "CPUUtilization",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ],
            "alarmName": "ServerCpuTooHigh",
            "description": "Goes into alarm when server CPU utilization is too high!",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:34.267+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        },
```

```
        "previousConfiguration": {
            "evaluationPeriods": 1,
            "threshold": 70,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [
                {
                    "id": "d6bfa85f-893e-b052-a58b-4f9295c9111a",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "CPUUtilization",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ],
            "alarmName": "ServerCpuTooHigh",
            "description": "Goes into alarm when server CPU utilization is too high!",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:13.757+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Atualização de um alarme composto com um alarme supressor

```
{
    "version": "0",
    "id": "4c6f4177-6bd5-c0ca-9f05-b4151c54568b",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T13:59:56Z",
    "region": "us-east-1",
```

```
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "update",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "value": "ALARM",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 360,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:56.290+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        },
        "previousConfiguration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:46.425+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Exclusão de um alarme matemático métrico

```
{
```

```
    "version": "0",
    "id": "f171d220-9e1c-c252-5042-2677347a83ed",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:07:13Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "operation": "delete",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:17.672+0000"
        },
        "configuration": {
            "evaluationPeriods": 1,
            "threshold": 10000,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [{
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "NetworkIn",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "m2",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "NetworkOut",
```

```
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "e1",
                    "expression": "SUM(METRICS())",
                    "label": "Total Network Traffic",
                    "returnData": true
                }
            ],
            "alarmName": "TotalNetworkTrafficTooHigh",
            "description": "Goes into alarm if total network traffic exceeds 10Kb",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:17.672+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []

        }
    }
}
```

Exclusão de um alarme composto com um alarme supressor

```
{
    "version": "0",
    "id": "e34592a1-46c0-b316-f614-1b17a87be9dc",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T14:00:01Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
```

```
        "operation": "delete",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "value": "ALARM",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 360,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:56.290+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Gerenciar alarmes

Como editar ou excluir um alarme do CloudWatch

É possível editar ou excluir um alarme existente.

Não é possível alterar o nome de um alarme existente. Copie o alarme e dê ao novo alarme um nome diferente. Para copiar um alarme, marque a caixa de seleção ao lado do nome do alarme na lista de alarmes e escolha Action (Ação), Copy (Copiar).

Para editar um alarme

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Escolha o nome do alarme.
4. Para adicionar ou remover tags, escolha a guia Tags e, em seguida, escolha Gerenciar tags.
5. Para editar outras partes do alarme, escolha Ações, Editar.

A página Specify metric and conditions (Especificar métrica e condições) será exibida, mostrando um gráfico e outras informações sobre a métrica e a estatística que você selecionou.

6. Para alterar a métrica, escolha Edit (Editar), escolha a guia All metrics (Todas as métricas) e execute uma das seguintes ações:
   - Escolha o namespace do serviço que contém a métrica desejada. Continue escolhendo as opções à medida que elas são exibidas para restringir as escolhas. Quando uma lista de métricas for exibida, marque a caixa de seleção ao lado da métrica que você deseja.
   - Na caixa de pesquisa, digite o nome de uma métrica, uma dimensão ou um ID de recurso e pressione Enter. Escolha um dos resultados e continue até uma lista de métricas ser exibida. Marque a caixa de seleção ao lado da métrica que você deseja.

   Escolha Selecionar métrica.
7. Para alterar outros aspectos do alarme, escolha as opções apropriadas. Para alterar o número de pontos de dados que devem estar violando para o alarme entrar no estado `ALARM` ou para alterar a maneira como os dados ausentes são tratados, escolha Additional configuration (Configuração adicional).
8. Escolha Próximo.
9. Em Notification (Notificação), Auto Scaling action (Ação do Auto Scaling) e EC2 action (Ação do EC2), é possível editar as ações executadas quando o alarme é acionado. Em seguida, escolha Próximo.
10. Opcionalmente, altere a descrição do alarme.

    Não é possível alterar o nome de um alarme existente. Copie o alarme e dê ao novo alarme um nome diferente. Para copiar um alarme, marque a caixa de seleção ao lado do nome do alarme na lista de alarmes e escolha Action (Ação), Copy (Copiar).
11. Escolha Próximo.
12. Em Preview and create (Visualizar e criar), confirme se as informações e condições são o que você deseja e escolha Update alarm (Atualizar alarme).

Para atualizar uma lista de notificações por e-mail que foi criada usando o console do Amazon SNS

1. Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No painel de navegação, escolha Topics (Tópicos) e selecione o ARN para a sua lista de notificações (tópico).

3. Execute um destes procedimentos:

   - Para adicionar um endereço de e-mail, escolha Create subscription (Criar inscrição). Em Protocolo, escolha Email. Em Endpoint, insira o endereço de e-mail do novo destinatário. Selecione Criar assinatura.
   - Para remover um endereço de e-mail, escolha ID da inscrição. Escolha Outras ações de inscrição, Excluir inscrições.

4. Selecione Publicar em um tópico.

Como excluir um alarme

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Alarmes.
3. Marque a caixa de seleção à esquerda do nome do alarme e escolha Ações, Excluir.
4. Escolha Excluir.

Ocultar alarmes do Auto Scaling

Ao visualizar seus alertas no AWS Management Console, você pode ocultar os alarmes relacionados ao Amazon EC2 Auto Scaling e ao Application Auto Scaling. Esse recurso está disponível somente no AWS Management Console.

Para ocultar temporariamente os alarmes do Auto Scaling

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os Alarmes), e selecione Hide Auto Scaling alarms (Ocultar todos os alarmes de AutoScaling).

Casos de uso e exemplos de alarmes

As seções a seguir fornecem exemplos e tutoriais de alarmes para casos de uso comuns.

Criar um alarme de faturamento para monitorar suas cobranças estimadas da AWS

É possível monitorar suas cobranças estimadas da AWS usando o Amazon CloudWatch. Quando você habilita o monitoramento de estimativas de cobrança para sua conta da AWS, as estimativas de cobrança são calculadas e enviadas várias vezes por dia para o CloudWatch como dados de métrica.

Os dados de métrica de faturamento são armazenados na região Leste dos EUA (Norte da Virgínia) e representam cobranças mundiais. Esses dados incluem as estimativas de cobrança para cada serviço da AWS que você usa, além do total geral estimado de suas cobranças da AWS.

O alarme é acionado quando o faturamento da conta excede o limite especificado. Ele é acionado somente quando o faturamento atual excede o limite. Ele não usa projeções com base no seu uso até o momento no mês.

Se você criar um alerta de faturamento quando suas cobranças já tiverem excedido o limite, o alarme mudará para o estado `ALARM` imediatamente.

> **Note**
>
> Para obter informações sobre como analisar as cobranças do CloudWatch nas quais você já incorreu, consulte Faturamento e custos do CloudWatch.

Tarefas

- Habilitar alertas de faturamento
- Criar um alarme de faturamento
- Excluir um alarme de faturamento

Habilitar alertas de faturamento

Para criar um alarme para suas estimativas de despesas, habilite alertas de faturamento para poder monitorar suas estimativas de despesas da AWS e criar um alarme usando dados de métrica de faturamento. Depois que habilitar alertas de faturamento, você não poderá desativar a coleta de dados, mas poderá excluir qualquer alarme de faturamento que tenha criado.

Depois que habilitar alertas de pagamento pela primeira vez, levará cerca de 15 minutos para que você possa visualizar dados de faturamento e definir alertas de pagamento.

Requisitos

- É necessário estar conectado usando as credenciais de usuário-raiz da conta ou como um usuário do IAM que tenha recebido permissão para visualizar as informações de faturamento.
- Para contas de faturamento consolidado, os dados de faturamento para cada conta vinculada podem ser encontrados fazendo login como a conta de pagamento. Você pode visualizar dados de faturamento para o total de cobranças estimadas e cobranças estimadas por serviço para cada conta vinculada, além da conta consolidada.
- Em uma conta de faturamento consolidado, as métricas da conta vinculada ao membro serão capturadas somente se a conta pagante habilitar a preferência Receber alertas de faturamento. Se você alterar qual é a conta de gerenciamento/pagante, será necessário habilitar os alertas de faturamento na nova conta de gerenciamento/pagante.
- A conta não deve fazer parte da Rede de parceiros da Amazon (APN) porque as métricas de faturamento não são publicadas no CloudWatch para contas do APN. Para obter mais informações, consulte Rede de parceiros da AWS.

Para ativar o monitoramento de estimativas de gastos

1. Abra o console do AWS Billing em https://console.aws.amazon.com/billing/.
2. No painel de navegação, selecione Billing Preferences (Preferências de faturamento).
3. Em Preferências de alertas, escolha Editar.
4. Escolha Receber alertas de faturamento do CloudWatch.
5. Selecione Salvar preferências.

Criar um alarme de faturamento

> **Important**
>
> Antes de criar um alarme de faturamento, defina a região como Leste dos EUA (Norte da Virgínia). Os dados de métrica de faturamento são armazenados nessa região e representam as despesas em todo o mundo. É necessário habilitar alertas de faturamento na conta ou na

conta de gerenciamento/pagante (se você estiver usando faturamento consolidado). Para ter mais informações, consulte Habilitar alertas de faturamento.

Neste procedimento, você cria um alarme que envia uma notificação quando suas estimativas de cobrança da AWS excedem um limite definido.

Para criar um alarme de cobrança usando o console do CloudWatch

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes) e depois escolha All alarms (Todos os alarmes).
3. Selecione Criar alarme.
4. Escolha Selecionar métrica. Em Browse (Navegar), escolha Billing (Faturamento) e escolha Total Estimated Charge (Total da cobrança estimada).

   > **Note**
   >
   > Caso não veja a métrica Faturamento/Total da cobrança estimada, habilite os alertas de faturamento e altere a região para Leste dos EUA (Norte da Virgínia). Para ter mais informações, consulte Habilitar alertas de faturamento.

5. Marque a caixa de seleção EstimatedCharges e escolha Select metric (Selecionar métrica).
6. Em Statistic (Estatística), escolha Maximum (Máximo).
7. Em Period (Período), escolha 6 hours (6 horas).
8. Em Tipo de limite, escolha Estático.
9. Em Whenever EstimatedCharges is… (Sempre que EstimatedCharges for…), escolha Greater (Maior).
10. Em que . . ., defina o valor para o qual você deseja que seu alarme seja acionado. Por exemplo, **200** USD.

    Os valores da métrica EstimatedCharges estão somente em dólares americanos (USD), e a conversão da moeda é fornecida pela Amazon Services LLC. Para obter mais informações, consulte O que é o AWS Billing?.

> **Note**
>
> Após definir um valor limite, o gráfico de pré-visualização exibirá suas cobranças estimadas do mês atual.

11. Em Configuração adicional, faça o seguinte:
    - Em Datapoints to alarm (Pontos de dados para alarme), especifique 1 of 1 (1 de 1).
    - Em Missing data treatment (Tratamento de dados ausentes), escolha Treat missing data as missing (Tratar dados ausentes como ausentes).
12. Escolha Próximo.
13. Em Notificação, certifique-se de que a opção Em alarme esteja selecionada. Em seguida, especifique um tópico do Amazon SNS a ser notificado quando o alarme estiver no estado `ALARM`. O tópico do Amazon SNS pode incluir seu endereço de email para que você receba emails quando o valor do faturamento ultrapassar o limite especificado.

    É possível selecionar um tópico existente do Amazon SNS, criar um novo tópico do Amazon SNS ou usar um ARN do tópico para notificar outra conta. Se quiser que o alarme envie várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).
14. Escolha Próximo.
15. Em Name and description (Nome e descrição), insira um nome para o alarme. O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII.
    - (Opcional) Insira uma descrição do alarme. A descrição pode incluir a formatação de markdown, que é exibida somente na guia Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.
16. Escolha Próximo.
17. Em Preview and create (Pré-visualizar e criar), verifique se a configuração está correta e escolha Create alarm (Criar alarme).

Excluir um alarme de faturamento

É possível excluir seu alarme de faturamento quando não precisar mais dele.

Para excluir um alarme de faturamento

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Se necessário, altere a região para US East (N. Virginia) (Leste dos EUA (Norte da Virgínia)). Os dados da métrica de faturamento são armazenados nessa região e refletem os custos em todo o mundo.
3. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
4. Marque a caixa de seleção ao lado do alarme e escolha Actions (Ações), Delete (Excluir).
5. Quando a confirmação for solicitada, escolha Sim, excluir.

Criar um alarme de utilização de CPU

É possível criar um alarme do CloudWatch que envia uma notificação usando o Amazon SNS quando o alarme muda do estado `OK` para `ALARM`.

O alarme muda para o estado `ALARM` quando o uso médio da CPU de uma instância do EC2 ultrapassa um limite especificado por períodos consecutivos especificados.

Configurar um alarme de uso da CPU usando o AWS Management Console

Use estas etapas da AWS Management Console para criar um alarme de utilização de CPU.

Como criar um alarme baseado no uso da CPU

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Selecione Criar alarme.
4. Escolha Selecionar métrica.
5. Na guia Todas as métricas, escolha Métricas do EC2.
6. Escolha uma categoria da métrica (por exemplo, Métricas por instância).
7. Localize a linha com a instância que deseja listar na coluna InstanceId e CPUUtilization na coluna Nome da métrica. Marque a caixa de seleção ao lado dessa linha e escolha Selecionar métrica.
8. Em Especificar métrica e condições, em Estatística, escolha Média e selecione um dos percentis predefinidos ou especifique um percentil personalizado (por exemplo, **`p95.45`**).
9. Escolha um período (por exemplo, **`5 minutes`**).

10. Em Conditions (Condições), especifique o seguinte:

    a. Em Tipo de limite, escolha Estático.

    b. Em Sempre que CPUUtilization for, especifique Maior. Em que..., especifique o limite que deve acionar o alarme para ir para o estado ALARM se a utilização da CPU exceder essa porcentagem. Por exemplo, 70.

    c. Escolha Additional configuration (Configuração adicional). Em Datapoints to alarm (Pontos de dados para alarme), especifique quantos períodos de avaliação (pontos de dados) devem estar no estado `ALARM` para disparar o alarme. Se os dois valores forem correspondentes, você criará um alarme que passa para o estado `ALARM` se esses períodos consecutivos estiverem violando.

       Para criar um alarme M de N, especifique um número menor para o primeiro valor que especificar para o segundo valor. Para ter mais informações, consulte Avaliar um alarme.

    d. Para o Missing data treatment (Tratamento de dados ausentes), escolha como deseja que o alarme se comporte quando alguns pontos de dados estiverem ausentes. Para ter mais informações, consulte Configurar como os alarmes do CloudWatch tratam dados ausentes.

    e. Se o alarme usar um percentil como estatística monitorada, uma caixa Percentiles with low samples (Percentis com amostras baixas) será exibida. Use-a para escolher se deseja avaliar ou ignorar casos com taxas de amostra baixas. Se você escolher ignore (maintain the alarm state) (ignorar (manter o estado do alarme)), o estado do alarme atual será sempre mantido quando o tamanho da amostra for muito baixo. Para ter mais informações, consulte Alarmes do CloudWatch baseados em percentual e exemplos de poucos dados.

11. Escolha Próximo.

12. Em Notificação, escolha Em alarme e selecione um tópico do SNS para notificar quando o alarme estiver no estado `ALARM`

    Para que o alarme envie várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

    Para que o alarme não envie notificações, escolha Remove (Remover).

13. Quando terminar, escolha Next (Próximo).

14. Digite um nome e uma descrição para o alarme. Em seguida, escolha Próximo.

    O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação de markdown, que é exibida somente na guia

Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.

15. Em Preview and create (Visualizar e criar), confirme se as informações e condições são o que você deseja e escolha Create alarm (Criar alarme).

Configurar um alarme de uso da CPU usando o AWS CLI

Use estas etapas da AWS CLI para criar um alarme de utilização de CPU.

Como criar um alarme baseado no uso da CPU

1. Configure um tópico do SNS. Para ter mais informações, consulte Configurar notificações do Amazon SNS.
2. Crie um alarme usando o comando put-metric-alarm da seguinte forma.

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70%" --metric-name CPUUtilization --namespace AWS/EC2 --statistic
 Average --period 300 --threshold 70 --comparison-operator GreaterThanThreshold --
dimensions  Name=InstanceId,Value=i-12345678 --evaluation-periods 2 --alarm-actions
 arn:aws:sns:us-east-1:111122223333:my-topic --unit Percent
```

3. Teste o alarme forçando uma alteração de estado com o comando set-alarm-state.

   a. Altere o estado do alarme de `INSUFFICIENT_DATA` para `OK`.

   ```
   aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason
    "initializing" --state-value OK
   ```

   b. Altere o estado do alarme de `OK` para `ALARM`.

   ```
   aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason
    "initializing" --state-value ALARM
   ```

   c. Verifique se você recebeu uma notificação sobre o alarme.

Criar um alarme de latência do balanceador de carga que envie um email

É possível configurar uma notificação do Amazon SNS e um alarme que monitore a latência que exceda 100 ms para Classic Load Balancer.

Configurar um alarme de latência usando o AWS Management Console

Use estas etapas para usar a AWS Management Console para criar um alarme de latência de load balancer.

Criar um alarme de latência de balanceador de carga

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Selecione Criar alarme.
4. Em Métricas do CloudWatch por categoria, selecione a categoria Métricas do ELB.
5. Selecione a linha com o Classic Load Balancer e a métrica Latency (Latência).
6. Para a estatística, escolha Average (Média), escolha um dos percentis predefinidos ou especifique um percentil personalizado (por exemplo, **`p95.45`**).
7. Para o período, escolha 1 Minute (1 minuto).
8. Escolha Próximo.
9. Em Alarm Threshold (Limite do alarme), insira um nome exclusivo para o alarme (por exemplo, **`myHighCpuAlarm`**) e uma descrição do alarme (por exemplo, **`Alarm when Latency exceeds 100s`**). Os nomes dos alarmes devem conter somente caracteres UTF-8 da e não podem conter caracteres de controle ASCII

   O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação de markdown, que é exibida somente na guia Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.
10. Em Whenever (Sempre que), em is (é), escolha **>** e insira **`0.1`**. Em for (para), insira **`3`**.
11. Em Additional settings (Configurações adicionais), em Treat missing data as (Tratar dados ausentes como), escolha ignore (maintain alarm state) (ignorar (manter estado do alarme)) para que os pontos de dados ausentes não acionem mudanças do estado do alarme.

    Em Percentiles with low samples (Percentis com amostras baixas), escolha ignore (maintain the alarm state) (ignorar [manter estado do alarme]) para que o alarme só avalie situações com números adequados de amostras de dados.
12. Em Actions (Ações), em Whenever this alarm (Sempre que este alarme), escolha State is ALARM (Estado é ALARME). Em Enviar notificação para, escolha um tópico do SNS existente ou crie um novo.

Para criar um tópico do SNS, escolha New list (Nova lista). Em Send notification to (Enviar notificação para), insira um nome para o tópico do SNS (por exemplo, **myHighCpuAlarm**). Em Email list (Lista de e-mails), insira uma lista de endereços de e-mail separados por vírgulas a serem notificados quando o alarme mudar para o estado ALARM. Para cada endereço de e-mail será enviado um e-mail de confirmação da inscrição no tópico. Você deve confirmar a inscrição para que as notificações sejam enviadas.

13. Escolha Create Alarm.

Configurar um alarme de latência usando o AWS CLI

Use estas etapas para usar a AWS CLI para criar um alarme de latência de load balancer.

Criar um alarme de latência de balanceador de carga

1. Configure um tópico do SNS. Para ter mais informações, consulte Configurar notificações do Amazon SNS.
2. Crie o alarme usando o comando put-metric-alarm da seguinte forma:

```
aws cloudwatch put-metric-alarm --alarm-name lb-mon --alarm-description "Alarm
 when Latency exceeds 100s" --metric-name Latency --namespace AWS/ELB --statistic
 Average --period 60 --threshold 100 --comparison-operator GreaterThanThreshold --
dimensions Name=LoadBalancerName,Value=my-server --evaluation-periods 3 --alarm-
actions arn:aws:sns:us-east-1:111122223333:my-topic --unit Seconds
```

3. Teste o alarme forçando uma alteração de estado com o comando set-alarm-state.

   a. Altere o estado do alarme de INSUFFICIENT_DATA para OK.

   ```
   aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
    "initializing" --state-value OK
   ```

   b. Altere o estado do alarme de OK para ALARM.

   ```
   aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
    "initializing" --state-value ALARM
   ```

   c. Verifique se você recebeu uma notificação por e-mail sobre o alarme.

Criar um alarme de throughput de armazenamento que envie emails

É possível configurar uma notificação do SNS e um alarme que é acionado quando o Amazon EBS excede a throughput de 100 MB.

Configurar um alarme de throughput de armazenamento usando o AWS Management Console

Realize estas etapas para usar o AWS Management Console para criar um alarme baseado na throughput do Amazon EBS.

Para criar um alarme de throughput de armazenamento

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Selecione Criar alarme.
4. Em Métricas do EBS, escolha uma categoria de métrica.
5. Selecione a linha com o volume e a métrica VolumeWriteBytes.
6. Para a estatística, escolha Média. Para o período, escolha 5 minutos. Escolha Próximo.
7. Em Alarm Threshold (Limite do alarme), insira um nome exclusivo para o alarme (por exemplo, **`myHighWriteAlarm`**) e uma descrição do alarme (por exemplo, **`VolumeWriteBytes exceeds 100,000 KiB/s`**). O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação de markdown, que é exibida somente na guia Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.
8. Em Whenever (Sempre que), em is (é), escolha **>** e insira **`100000`**. Em for (para), insira **`15`** períodos consecutivos.

   Uma representação gráfica do limite será exibida em Alarm Preview (Visualização do alarme).
9. Em Additional settings (Configurações adicionais), em Treat missing data as (Tratar dados ausentes como), escolha ignore (maintain alarm state) (ignorar (manter estado do alarme)) para que os pontos de dados ausentes não acionem mudanças do estado do alarme.
10. Em Actions (Ações), em Whenever this alarm (Sempre que este alarme), escolha State is ALARM (Estado é ALARME). Em Enviar notificação para, escolha um tópico do SNS existente ou crie um.

Para criar um tópico do SNS, escolha New list (Nova lista). Em Send notification to (Enviar notificação para), insira um nome para o tópico do SNS (por exemplo, **myHighCpuAlarm**). Em Email list (Lista de e-mails), insira uma lista de endereços de e-mail separados por vírgulas a serem notificados quando o alarme mudar para o estado ALARM. Para cada endereço de e-mail será enviado um e-mail de confirmação da inscrição no tópico. Você deve confirmar a assinatura para que as notificações sejam enviadas para um endereço de e-mail.

11. Escolha Create Alarm.

Configurar um alarme de throughput de armazenamento usando o AWS CLI

Realize estas etapas para usar o AWS CLI para criar um alarme baseado na throughput do Amazon EBS.

Para criar um alarme de throughput de armazenamento

1. Criar um tópico do SNS. Para ter mais informações, consulte Configurar notificações do Amazon SNS.
2. Crie o alarme.

```
aws cloudwatch put-metric-alarm --alarm-name ebs-mon --alarm-description "Alarm
 when EBS volume exceeds 100MB throughput" --metric-name VolumeReadBytes --
namespace AWS/EBS --statistic Average --period 300 --threshold 100000000 --
comparison-operator GreaterThanThreshold --dimensions Name=VolumeId,Value=my-
volume-id --evaluation-periods 3 --alarm-actions arn:aws:sns:us-
east-1:111122223333:my-alarm-topic --insufficient-data-actions arn:aws:sns:us-
east-1:111122223333:my-insufficient-data-topic
```

3. Teste o alarme forçando uma alteração de estado com o comando set-alarm-state.

   a. Altere o estado do alarme de INSUFFICIENT_DATA para OK.

   ```
   aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
    "initializing" --state-value OK
   ```

   b. Altere o estado do alarme de OK para ALARM.

   ```
   aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
    "initializing" --state-value ALARM
   ```

c. Altere o estado do alarme de `ALARM` para `INSUFFICIENT_DATA`.

```
aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
 "initializing" --state-value INSUFFICIENT_DATA
```

d. Verifique se você recebeu uma notificação por e-mail sobre o alarme.

Crie um alarme para as métricas do contador do Performance Insights a partir de um banco de dados da AWS

O CloudWatch inclui uma função matemática métrica DB_PERF_INSIGHTS que pode ser usada para trazer métricas de contador do Performance Insights para o CloudWatch a partir do Amazon Relational Database Service e do Amazon DocumentDB (compatível com MongoDB). DB_PERF_INSIGHTS também traz a métrica `DBLoad` em intervalos inferiores a um minuto. Também é possível usar definir alarmes do CloudWatch para essas métricas.

Para obter mais informações sobre o Insights de Performance do Amazon RDS, consulte Monitoramento de carga de banco de dados com o Insights de Performance no Amazon RDS.

Para obter mais informações sobre o Insights de Performance do Amazon DocumentDB, consulte Monitoramento com o Insights de Performance.

Não há suporte para a detecção de anomalias para alarmes baseados na função DB_PERF_INSIGHTS.

> **Note**
>
> Métricas de alta resolução com granularidade de menos de um minuto recuperadas pelo DB_PERF_INSIGHTS são aplicáveis somente à métrica DBLoad ou às métricas do sistema operacional caso você tenha ativado o monitoramento aprimorado em uma resolução maior. Para obter mais informações sobre o monitoramento avançado do Amazon RDS, consulte Monitoramento de métricas do SO com monitoramento avançado.
> É possível criar um alarme de alta resolução usando a função DB_PERF_INSIGHTS. Três horas é o intervalo máximo de avaliação para um alarme de alta resolução. É possível usar o console do CloudWatch para representar graficamente as métricas recuperadas com a função DB_PERF_INSIGHTS para qualquer intervalo de tempo.

Para criar um alarme com base nas métricas do Insights de Performance

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes) e depois escolha All alarms (Todos os alarmes).
3. Selecione Criar alarme.
4. Escolha Select metric (Selecionar métrica).
5. Escolha o menu suspenso Adicionar matemática e, em seguida, selecione Métricas de performance de banco de dados, DB_PERF_INSIGHTS na lista.

   Depois de escolher DB_PERF_INSIGHTS, uma caixa de expressão matemática aparecerá onde você aplica ou edita expressões matemáticas.
6. Na caixa de expressão matemática, insira sua expressão matemática DB_PERF_INSIGHTS e, em seguida, escolha Aplicar.

   Por exemplo, **`DB_PERF_INSIGHTS('RDS', 'db-ABCDEFGHIJKLMNOPQRSTUVWXY1', 'os.cpuUtilization.user.avg')`**

   > **Important**
   >
   > Ao usar a expressão matemática DB_PERF_INSIGHTS, você deve especificar o ID de recurso exclusivo do banco de dados para o banco de dados. Isso é diferente do identificador do banco de dados. Para encontrar o ID de recurso do banco de dados no console do Amazon RDS, escolha a instância de banco de dados para visualizar os detalhes. Em seguida, escolha a guia Configuration (Configuração). O ID de recurso é exibido na seção Configuração.

   Para obter informações sobre a função DB_PERF_INSIGHTS e outras funções disponíveis para matemática de métrica, consulte Sintaxe de funções da matemática métricas.
7. Escolha Selecionar métrica.

   A página Specify metric and conditions (Especificar métrica e condições) será exibida, mostrando um gráfico e outras informações sobre a expressão matemática que você selecionou.
8. Em Whenever ***`expression`*** is (Sempre que a expressão for), especifique se a expressão deverá ser maior que, menor que ou igual ao limite. Em than... (que...), especifique o valor limite.

9. Escolha Additional configuration (Configuração adicional). Em Datapoints to alarm (Pontos de dados para alarme), especifique quantos períodos de avaliação (pontos de dados) devem estar no estado `ALARM` para disparar o alarme. Se os dois valores forem correspondentes, você criará um alarme que passa para o estado `ALARM` se esses períodos consecutivos estiverem violando.

   Para criar um alarme M de N, especifique um número menor para o primeiro valor que especificar para o segundo valor. Para ter mais informações, consulte Avaliar um alarme.
10. Para o Missing data treatment (Tratamento de dados ausentes), escolha como deseja que o alarme se comporte quando alguns pontos de dados estiverem ausentes. Para ter mais informações, consulte Configurar como os alarmes do CloudWatch tratam dados ausentes.
11. Escolha Próximo.
12. Em Notification (Notificação), selecione um tópico do SNS para notificar quando o alarme estiver no estado `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

    Para que o alarme envie várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

    Para que o alarme não envie notificações, escolha Remove (Remover).
13. Para que o alarme execute ações do Auto Scaling, EC2, Lambda ou Systems Manager, escolha o botão apropriado e selecione o estado do alarme e a ação a ser executada. Se você escolher uma função do Lambda como uma ação de alarme, especifique o nome da função ou o ARN e, opcionalmente, você poderá escolher uma versão específica da função.

    Os alarmes só poderão executar ações do Systems Manager ao entrarem no estado ALARM. Para obter mais informações sobre ações do Systems Manager, consulte Configurar o CloudWatch para criar OpsItems a partir de alarmes e Criação de incidentes.

    > **Note**
    >
    > Para criar um alarme que executa uma ação do SSM Incident Manager, é necessário ter determinadas permissões. Para obter mais informações, consulte Exemplos de políticas baseadas em identidade para o AWSSystems Manager Incident Manager.
14. Quando terminar, escolha Next (Próximo).
15. Digite um nome e uma descrição para o alarme. Em seguida, escolha Próximo.

    O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação de markdown, que é exibida somente na guia

Detalhes do alarme no console do CloudWatch. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.

16. Em Preview and create (Visualizar e criar), confirme se as informações e condições são o que você deseja e escolha Create alarm (Criar alarme).

Criar alarmes para interromper, terminar, reinicializar ou recuperar uma instância do EC2

Usando as ações de alarme do Amazon CloudWatch, você cria alarmes que automaticamente interrompem, terminam, reinicializam ou recuperam suas instâncias do EC2. É possível usar as ações de parada ou encerramento para ajudar a economizar dinheiro quando não precisar mais que uma instância seja executada. É possível usar as ações de reinicialização e recuperação para reinicializar automaticamente essas instâncias ou recuperá-las para um novo hardware caso ocorra um problema no sistema.

Há várias situações nas quais é possível querer interromper ou encerrar sua instância automaticamente. Por exemplo, é possível ter instâncias dedicadas a trabalhos de processamento de folha de pagamento em lote ou tarefas de computação científica que são executadas por um período e, em seguida, concluem seu trabalho. Em vez de permitir que essas instâncias permaneçam ociosas (e acumulem cobranças), interrompa ou as encerre, o que ajuda a economizar. A principal diferença entre usar as ações de alarme de interrupção e encerramento é que você poderá reiniciar facilmente uma instância interrompida se precisar reexecutá-la mais tarde. Também mantenha os mesmos ID de instância e volume raiz. No entanto, não é possível reiniciar uma instância encerrada. Em vez disso, é necessário executar uma nova instância.

É possível adicionar as ações de interromper, terminar, reinicializar ou recuperar a qualquer alarme definido em uma métrica por instância do Amazon EC2, incluindo métricas de monitoramento básicas e detalhadas fornecidas pelo Amazon CloudWatch (no namespace AWS/EC2), além de todas as métricas personalizadas que incluem a dimensão “InstanceId=”, desde que o valor de InstanceId se refira a uma instância do Amazon EC2 válida em execução. Você também pode adicionar a ação de recuperar aos alarmes definidos em qualquer métrica por instância do Amazon EC2, exceto `StatusCheckFailed_Instance`.

Para configurar uma ação de alarme do CloudWatch que pode reinicializar, interromper ou terminar uma instância, você deve usar uma função do IAM vinculada ao serviço AWSServiceRoleForCloudWatchEvents. A função do IAM AWSServiceRoleForCloudWatchEvents permite que a AWS execute ações de alarme em seu nome.

Para criar a função vinculada ao serviço para o CloudWatch Events, use o seguinte comando:

```
aws iam create-service-linked-role --aws-service-name events.amazonaws.com
```

Suporte a consoles

Você pode criar alarmes usando o console do CloudWatch ou o console do Amazon EC2. Os procedimentos nesta documentação usam o console do CloudWatch. Para procedimentos que usam o console do Amazon EC2, consulte Criar alarmes que interrompem, terminam, reinicializam ou recuperam uma instância no Manual do usuário do Amazon CloudWatch.

Permissões

Se você estiver usando uma conta do AWS Identity and Access Management (IAM) para criar ou modificar um alarme que executa ações do EC2 ou ações do Systems Manager OpsItem, deverá ter a permissão `iam:CreateServiceLinkedRole`.

Conteúdo

- Adicionar ações de interromper a alarmes do Amazon CloudWatch
- Adicionar ações de terminar a alarmes do Amazon CloudWatch
- Adicionar ações de reinicializar a alarmes do Amazon CloudWatch
- Adicionar ações de recuperar a alarmes do Amazon CloudWatch
- Visualizar o histórico de alarmes disparados e ações

Adicionar ações de interromper a alarmes do Amazon CloudWatch

É possível criar um alarme que pare uma instância do Amazon EC2 quando o limite for atingido. Por exemplo, é possível executar instâncias de desenvolvimento ou teste e ocasionalmente se esquecer de desativá-las. É possível criar um alarme que seja acionado quando o percentual médio de utilização da CPU for inferior a 10% em 24 horas, sinalizando que ela está ociosa e não mais em uso. Você pode ajustar o limite, a duração e o período para atender às suas necessidades, além de poder adicionar uma notificação do SNS para receber um e-mail quando o alarme for disparado.

As instâncias do Amazon EC2 que usam um volume do Amazon Elastic Block Store como dispositivo raiz podem ser interrompidas ou terminadas, enquanto as instâncias que usam o armazenamento de instância como dispositivo raiz só podem ser terminadas.

Para criar um alarme para interromper uma instância ociosa usando o console do Amazon CloudWatch

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Selecione Criar alarme.
4. Escolha Select metric (Selecionar métrica).
5. Para namespaces da AWS, escolha EC2.
6. Faça o seguinte:
   a. Escolha Per-Instance Metrics (Métricas por instância).
   b. Selecione a caixa de seleção na linha com a instância correta e a métrica CPUUtilization.
   c. Escolha a guia Graphed metrics (Métricas em gráfico).
   d. Para a estatística, escolha Média.
   e. Escolha um período (por exemplo, **1 Hour**).
   f. Escolha Selecionar métrica.
7. Na etapa Definir alarme, faça o seguinte:
   a. Em Conditions (Condições), escolha Static (Estático).
   b. Em Whenever CPUUtilization is (Sempre que a CPUUtilization for), escolha Lower (Mais baixo).
   c. Para than (do que), digite **10**.
   d. Escolha Próximo.
   e. Em Notification (Notificação), para Send notification to (Enviar notificação para), escolha um tópico do SNS existente ou crie um novo.

      Para criar um tópico do SNS, escolha New list (Nova lista). Em Send notification to (Enviar notificação para), digite um nome para o tópico do SNS (por exemplo, Stop_EC2_Instance). Para Email list (Lista de e-mails), digite uma lista de endereços de e-mail separados por vírgulas a serem notificados quando o alarme mudar para o estado `ALARM`. Para cada endereço de e-mail será enviado um e-mail de confirmação da inscrição no tópico. Você deve confirmar a assinatura para que as notificações sejam enviadas para um endereço de e-mail.

   f. Escolha Add EC2 Action (Adicionar ação do EC2).

g. Em Alarm state trigger (Gatilho do estado do alarme), escolha In alarm (Em alarme). Em Take the following action (Executar a ação a seguir), escolha Stop this instance (Interromper a instância).

h. Escolha Próximo.

i. Digite um nome e uma descrição para o alarme. O nome deve conter somente caracteres ASCII. Em seguida, escolha Próximo.

j. Em Preview and create (Visualizar e criar), confirme se as informações e condições são o que você deseja e escolha Create alarm (Criar alarme).

Adicionar ações de terminar a alarmes do Amazon CloudWatch

É possível criar um alarme que encerre uma instância do EC2 automaticamente quando um certo limite for atingido (desde que a proteção contra encerramento não esteja ativada para a instância). Por exemplo, você pode encerrar uma instância quando ela tiver concluído seu trabalho e não precisar mais dela. Se você quiser usar a instância posteriormente, pare-a em vez de encerrá-la. Para obter informações sobre habilitar e desabilitar a proteção contra término de uma instância, consulte Habilitar a proteção contra término de uma instância no Manual do usuário do Amazon EC2 para instâncias do Linux.

Para criar um alarme para terminar uma instância ociosa usando o console do Amazon CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms, Create Alarm.
3. Na etapa Selecionar métrica, faça o seguinte:

   a. Em Métricas do EC2, escolha Métricas por instância.

   b. Selecione a linha com a instância e a métrica CPUUtilization.

   c. Para a estatística, escolha Média.

   d. Escolha um período (por exemplo, **1 Hour**).

   e. Escolha Próximo.
4. Na etapa Definir alarme, faça o seguinte:

   a. Em Limite de alarme, digite um nome exclusivo para o alarme (por exemplo, Encerrar instância do EC2) e uma descrição do alarme (por exemplo, Encerrar instância do EC2 quando a CPU estiver ociosa por muito tempo). Os nomes de alarme devem conter somente caracteres ASCII.

b. Em Whenever (Sempre), em is (é), escolha **<** e digite **10**. Em for (para), digite **24** períodos consecutivos.

   Uma representação gráfica do limite será exibida em Alarm Preview (Visualização do alarme).

c. Em Notification (Notificação), para Send notification to (Enviar notificação para), escolha um tópico do SNS existente ou crie um novo.

   Para criar um tópico do SNS, escolha New list (Nova lista). Em Send notification to (Enviar notificação para), digite um nome para o tópico do SNS (por exemplo, Terminate_EC2_Instance) Para Email list (Lista de e-mails), digite uma lista de endereços de e-mail separados por vírgulas a serem notificados quando o alarme mudar para o estado `ALARM`. Para cada endereço de e-mail será enviado um e-mail de confirmação da inscrição no tópico. Você deve confirmar a assinatura para que as notificações sejam enviadas para um endereço de e-mail.

d. Escolha Ação do EC2.

e. Em Sempre que este alarme, escolha Estado é ALARME. Em Tomar esta medida, escolha Encerrar esta instância.

f. Escolha Create Alarm.

Adicionar ações de reinicializar a alarmes do Amazon CloudWatch

É possível criar um alarme do Amazon CloudWatch que monitore uma instância do Amazon EC2 e reinicialize automaticamente a instância. A ação de alarme de reinicialização é recomendada para falhas de verificação de integridade da instância (ao contrário da ação de alarme de recuperação, que é adequado para falhas de verificação de integridade do sistema). Reinicializar a instância equivale a reinicializar o sistema operacional. Na maioria dos casos, leva apenas alguns minutos para reinicializar sua instância. Quando você reinicializa uma instância, ela permanece no mesmo host físico, para que sua instância mantenha seu nome DNS público, o endereço IP privado e os dados em seus volumes de armazenamento de instância.

A reinicialização de uma instância não inicia uma nova hora de faturamento de instância, ao contrário de pará-la e reiniciá-la. Para obter mais informações sobre como reiniciar uma instância, consulte Reiniciar a instância no Manual do usuário do Amazon EC2 para instâncias do Linux.

> **Important**
>
> Para evitar um comportamento de disputa entre as ações de reinicialização e recuperação, evite configurar o mesmo período de avaliação para um alarme de reinicialização e um alarme de recuperação. Recomendamos que você defina alarmes de reinicialização para três períodos de avaliação de um minuto cada.

Para criar um alarme para reinicializar uma instância usando o console do Amazon CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms, Create Alarm.
3. Na etapa Selecionar métrica, faça o seguinte:
   a. Em Métricas do EC2, escolha Métricas por instância.
   b. Selecione a linha com a instância e a métrica StatusCheckFailed_Instance.
   c. Para a estatística, escolha Mínima.
   d. Escolha um período (por exemplo, **`1 Minute`**).
   e. Escolha Próximo.
4. Na etapa Definir alarme, faça o seguinte:
   a. Em Limite de alarme, digite um nome exclusivo para o alarme (por exemplo, Reinicializar instância do EC2) e uma descrição do alarme (por exemplo, Reinicializar instância do EC2 quando a verificação de integridade falhar). Os nomes de alarme devem conter somente caracteres ASCII.
   b. Em Whenever (Sempre), em is (é), escolha **>** e digite **`0`**. Em for (para), digite **`3`** períodos consecutivos.

      Uma representação gráfica do limite será exibida em Alarm Preview (Visualização do alarme).
   c. Em Notification (Notificação), para Send notification to (Enviar notificação para), escolha um tópico do SNS existente ou crie um novo.

      Para criar um tópico do SNS, escolha New list (Nova lista). Em Send notification to (Enviar notificação para), digite um nome para o tópico do SNS (por exemplo, Reboot_EC2_Instance). Para Email list (Lista de e-mails), digite uma lista de endereços de

e-mail separados por vírgulas a serem notificados quando o alarme mudar para o estado `ALARM`. Para cada endereço de e-mail será enviado um e-mail de confirmação da inscrição no tópico. Você deve confirmar a assinatura para que as notificações sejam enviadas para um endereço de e-mail.

d. Escolha Ação do EC2.

e. Em Sempre que este alarme, escolha Estado é ALARME. Para Tomar esta medida, escolha Reinicializar esta instância.

f. Escolha Create Alarm.

Adicionar ações de recuperar a alarmes do Amazon CloudWatch

É possível criar um alarme do Amazon CloudWatch que monitore uma instância do Amazon EC2 e recupere-a automaticamente se ocorrer um problema devido a uma falha de hardware subjacente ou um problema que exija o envolvimento da AWS para repará-lo. Instâncias encerradas não podem ser recuperadas. Uma instância recuperada é idêntica à instância original, incluindo o ID da instância, endereços IP privados, endereços IP elásticos e todos os metadados de instância.

Quando o alarme `StatusCheckFailed_System` for acionado e a ação de recuperação for iniciada, você será notificado pelo tópico do Amazon SNS que escolheu ao criar o alarme e a ação de recuperação associada. Durante a recuperação da instância, a instância será migrada durante uma reinicialização da instância e todos os dados na memória serão perdidos. Quando o processo é concluído, as informações serão publicadas no tópico do SNS que você tiver configurado para o alarme. Qualquer pessoa que estiver inscrita neste tópico do SNS receberá uma notificação por e-mail com o status da tentativa de recuperação e mais instruções. Você perceberá uma reinicialização da instância na instância recuperada.

A ação de recuperação pode ser usada somente com `StatusCheckFailed_System`, não com `StatusCheckFailed_Instance`.

Exemplos de problemas que causam falha nas verificações de status do sistema incluem:

- Perda de conectividade de rede
- Perda de energia do sistema
- Problemas de software no host físico
- Problemas de hardware de host físico que afetam a acessibilidade de rede

A ação de recuperar só é compatível com alguns tipos de instâncias. Para obter mais informações sobre os tipos de instância compatíveis e outros requisitos, consulte Recuperar sua instância e Requisitos.

> **Important**
>
> Para evitar um comportamento de disputa entre as ações de reinicialização e recuperação, evite configurar o mesmo período de avaliação para um alarme de reinicialização e um alarme de recuperação. Recomendamos que você defina alarmes de recuperação para dois períodos de avaliação de um minuto cada e alarmes de reinicialização para três períodos de avaliação de um minuto cada.

Para criar um alarme para recuperar uma instância usando o console do Amazon CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms, Create Alarm.
3. Na etapa Selecionar métrica, faça o seguinte:
   a. Em Métricas do EC2, escolha Métricas por instância.
   b. Selecione a linha com a instância e a métrica StatusCheckFailed_System.
   c. Para a estatística, escolha Mínima.
   d. Escolha um período (por exemplo, **`1 Minute`**).

      > **Important**
      >
      > Para evitar um comportamento de disputa entre as ações de reinicialização e recuperação, evite configurar o mesmo período de avaliação para um alarme de reinicialização e um alarme de recuperação. É recomendável que você defina os alarmes de recuperação para dois períodos de avaliação de um minuto cada.

   e. Escolha Próximo.
4. Na etapa Definir alarme, faça o seguinte:
   a. Em Limite de alarme, digite um nome exclusivo para o alarme (por exemplo, Recuperar instância do EC2) e uma descrição do alarme (por exemplo, Recuperar instância do EC2

quando a verificação de integridade falhar). Os nomes de alarme devem conter somente caracteres ASCII.

b. Em Whenever (Sempre), em is (é), escolha **>** e digite **0**. Em for (para), digite **2** períodos consecutivos.

c. Em Notification (Notificação), para Send notification to (Enviar notificação para), escolha um tópico do SNS existente ou crie um novo.

   Para criar um tópico do SNS, escolha New list (Nova lista). Em Send notification to (Enviar notificação para), digite um nome para o tópico do SNS (por exemplo, Recover_EC2_Instance). Para Email list (Lista de e-mails), digite uma lista de endereços de e-mail separados por vírgulas a serem notificados quando o alarme mudar para o estado `ALARM`. Para cada endereço de e-mail será enviado um e-mail de confirmação da inscrição no tópico. Você deve confirmar a assinatura para que as notificações sejam enviadas para um endereço de e-mail.

d. Escolha Ação do EC2.

e. Em Sempre que este alarme, escolha Estado é ALARME. Para Tomar esta medida, escolha Recuperar esta instância.

f. Escolha Create Alarm.

Visualizar o histórico de alarmes disparados e ações

É possível visualizar o histórico de alarmes e ações no console do Amazon CloudWatch. O Amazon CloudWatch mantém o histórico de alarmes e de ações dos últimos 30 dias.

Para visualizar o histórico de alarmes e ações disparados

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes) e selecione um alarme.
3. Para visualizar a transição de estado mais recente com os valores de tempo e métrica, escolha a guia Detalhes.
4. Para visualizar as entradas mais recentes do histórico, escolha a guia History (Histórico).

Alarmes e marcação

As etiquetas são pares de valores-chave que podem ajudar você a organizar e categorizar os recursos. Você também pode usá-las para definir o escopo de permissões de usuários, concedendo a um usuário permissão para acessar ou alterar apenas recursos com determinados valores de etiqueta. Para obter informações mais gerais sobre a marcação de recursos, consulte Tagging your AWS resources.

A lista apresentada a seguir explica alguns detalhes sobre como a marcação funciona com os alarmes do CloudWatch.

- Para que seja possível definir ou atualizar as etiquetas para um recurso do CloudWatch, é necessário estar conectado a uma conta que tenha a permissão `cloudwatch:TagResource`. Por exemplo, para criar um alarme e definir as etiquetas para ele, é necessário ter a permissão `cloudwatch:TagResource`, além da permissão `the cloudwatch:PutMetricAlarm` . Recomendamos que você se certifique de que qualquer pessoa em sua organização que criará ou atualizará os recursos do CloudWatch tenha a permissão `cloudwatch:TagResource`.
- As etiquetas podem ser usadas para obter um controle de autorização baseado em etiquetas. Por exemplo, as permissões de perfil ou de usuário do IAM podem incluir condições para limitar as chamadas do CloudWatch para recursos específicos com base nas etiquetas. No entanto, considere o seguinte:
  - As etiquetas com nomes que começam com `aws:` não podem ser usadas para obter um controle de autorização baseado em etiquetas.
  - Os alarmes compostos não oferecem suporte para o controle de autorização baseado em etiquetas.

Application Signals

⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Use o CloudWatch Application Signals para instrumentar de forma automática as aplicações na AWS com a finalidade de que você possa monitorar a integridade atual das aplicações e acompanhar a performance delas em longo prazo em relação aos seus objetivos de negócios. O Application Signals fornece uma visualização unificada e centrada em aplicações para aplicações, serviços e dependências, além de ajudar você a monitorar e realizar a triagem da integridade da aplicação.

- Habilite o Application Signals para coletar métricas e rastreamentos das aplicações de forma automática e exibir métricas importantes, como o volume de chamadas, a disponibilidade, a latência, as falhas e os erros. Visualize e realize a triagem da integridade operacional atual com rapidez, bem como acompanhe se as aplicações estão atendendo às metas de performance de longo prazo, sem a necessidade de escrever um código personalizado ou de criar painéis.
- Crie e monitore os objetivos de nível de serviço (SLOs) com o Application Signals. Crie e rastreie com facilidade o status de SLOs relacionados às métricas do CloudWatch, incluindo as novas métricas da aplicação padrão coletadas pelo Application Signals. Visualize e rastreie o status do indicador de nível de serviço (SLI) para os serviços da sua aplicação em uma lista de serviços e um mapeamento de topologia. Crie alarmes para rastrear seus SLOs e acompanhar as novas métricas da aplicação padrão coletadas pelo Application Signals.
- Visualize um mapeamento da topologia da sua aplicação que o Application Signals descobre automaticamente, que fornece uma representação visual das aplicações, das dependências e da conectividade.
- O Application Signals funciona com o CloudWatch RUM, com os canários do CloudWatch Synthetics, com o AWS Service Catalog AppRegistry e com o Amazon EC2 Auto Scaling para exibir as páginas de clientes, os canários do Synthetics e os nomes de aplicações em painéis e mapeamentos.

Uso do Application Signals para realizar o monitoramento diário de aplicações

Use o Application Signals no console do CloudWatch como parte do monitoramento diário de aplicações:

1. Se você criou objetivos de nível de serviço (SLOs) para seus serviços, comece com a página Objetivos de nível de serviço (SLOs). Os objetivos fornecerão uma visualização imediata da integridade dos serviços e das operações mais importantes. Escolha o nome do serviço ou da operação de um SLO para abrir a página Detalhes do serviço e visualizar informações detalhadas sobre o serviço à medida que soluciona problemas.
2. Abra a página Serviços para visualizar um resumo de todos os seus serviços e observar com rapidez os serviços com a maior taxa de falhas ou latência. Se você criou SLOs, consulte a tabela Serviços para visualizar quais serviços têm indicadores de nível de serviço (SLIs) não íntegros. Se um determinado serviço estiver em um estado não íntegro, selecione o serviço para abrir a página Detalhes do serviço e visualizar as operações, as dependências, os canários do Synthetics e as solicitações de clientes do serviço. Selecione um ponto em um gráfico para visualizar os rastreamentos correlacionados para que você possa solucionar e identificar a causa-raiz dos problemas operacionais.
3. Se os novos serviços foram implantados ou as dependências foram alteradas, abra o Mapa de serviços para inspecionar a topologia da aplicação. Visualize um mapeamento das aplicações que mostra o relacionamento entre clientes, os canários do Synthetics, os serviços e as dependências. Obtenha com rapidez a integridade do SLI, visualize as principais métricas, como o volume de chamadas, a taxa de falhas e a latência, e realize uma busca detalhada para consultar informações mais detalhadas na página Detalhes do serviço.

O uso do Application Signals incorre em cobranças. Para obter informações sobre os preços do CloudWatch, consulte Definição de preço do Amazon CloudWatch.

> **Note**
>
> Não é necessário habilitar o Application Signals para usar o CloudWatch Synthetics, o CloudWatch RUM ou o CloudWatch Evidently. No entanto, o Synthetics e o CloudWatch RUM funcionam com o Application Signals para fornecer benefícios quando você usa esses recursos em conjunto.

Linguagens e arquiteturas compatíveis

No momento, o Application Signals é compatível com aplicações em Java e Python.

O Application Signals é compatível e foi testado para o Amazon EKS, Amazon ECS e Amazon EC2. Nos clusters do Amazon EKS, ele descobre automaticamente os nomes dos serviços e dos clusters.

Em outras arquiteturas, você deve fornecer os nomes dos serviços e dos ambientes ao habilitar esses serviços para o Application Signals.

As instruções para habilitar o Application Signals no Amazon EC2 devem funcionar em qualquer arquitetura que ofereça suporte ao agente do CloudWatch e ao AWS Distro para OpenTelemetry. No entanto, as instruções não foram testadas em arquiteturas diferentes do Amazon ECS e do Amazon EC2.

Supported Regions (Regiões compatíveis)

Para essa versão de pré-visualização, o Application Signals é compatível com as regiões apresentadas a seguir.

- Leste dos EUA (Norte da Virgínia)
- Leste dos EUA (Ohio)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Europa (Irlanda)

Pré-visualização do SDK

Uma versão de pré-visualização do SDK está disponível para download.

> **Warning**
>
> As operações e os parâmetros da API estão sujeitos a alterações antes que o Application Signals seja disponibilizado para o público em geral. Essas alterações podem ser alterações significativas. Não use a versão de pré-visualização do SDK para fins de produção.

Para instalar a pré-visualização do SDK, primeiro, é necessário instalar ou atualizar para a versão mais recente da AWS CLI versão 2. Para obter mais informações, consulte Instalar ou atualizar a versão mais recente do AWS CLI.

Em seguida, use os comandos apresentados a seguir para fazer download do arquivo em zip do SDK do bucket do Amazon S3 e extrair seu conteúdo. Cada arquivo em zip do SDK contém as instruções do SDK e a documentação da API.

> **Note**
>
> O SDK é fornecido em várias linguagens de programação para que você possa usar as APIs do Application Signals com qualquer uma delas. No entanto, a instrumentação automática da sua aplicação para enviar dados para o Application Signals é compatível somente com aplicações em Java e Python.

- SDK para Java V2: `aws s3 cp s3://application-signals-preview-sdk/awsJavaSdkV2.zip ./`
- SDK para JavaScript V3: `aws s3 cp s3://application-signals-preview-sdk/jsSdkV3.zip ./`
- SDK para JavaScript V2: `aws s3 cp s3://application-signals-preview-sdk/jsSdkV2.zip ./`
- SDK para Python: `aws s3 cp s3://application-signals-preview-sdk/pythonSdk.zip ./`
- SDK do Kotlin: `aws s3 cp s3://application-signals-preview-sdk/kotlin.zip ./`
- SDK para Android: `aws s3 cp s3://application-signals-preview-sdk/andriod.zip ./`
- SDK para C++: `aws s3 cp s3://application-signals-preview-sdk/awsCppSdk.zip ./`
- SDK para PHP: `aws s3 cp s3://application-signals-preview-sdk/awsSdkPhp.zip ./`
- SDK para Ruby: `aws s3 cp s3://application-signals-preview-sdk/awsSdkRuby.zip ./`
- SDK para Go V2: `aws s3 cp s3://application-signals-preview-sdk/awsSdkGoV2.zip ./`
- SDK para Go V1: `aws s3 cp s3://application-signals-preview-sdk/go.zip ./`
- SDK para iOS: `aws s3 cp s3://application-signals-preview-sdk/iOS.zip ./`

Tópicos

- Objetivos de nível de serviço (SLOs)
- Monitorar a integridade operacional das suas aplicações com o Application Signals
- Coleta de métricas de aplicações padrão
- Uso do monitoramento sintético
- Execução de lançamentos e experimentos A/B com o CloudWatch Evidently
- Usar o CloudWatch RUM

Permissões necessárias para o Application Signals

⚠ O Application Signals está na versão de pré-visualização para Amazon CloudWatch e está sujeito a alterações.

Esta seção explica as permissões necessárias para que você habilite, gerencie e opere o Application Signals.

Permissões para habilitar e gerenciar o Application Signals

Para gerenciar o Application Signals, você deve ter feito logon com as seguintes permissões:

```
{
 "Version": "2012-10-17",
 "Statement": [
    {
        "Sid": "CloudWatchApplicationSignalsFullAccessPermissions",
        "Effect": "Allow",
        "Action": "application-signals:*",
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsAlarmsPermissions",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:DescribeAlarms"
        ],
        "Resource": "*"
    },
    {
```

```
        "Sid": "CloudWatchApplicationSignalsMetricsPermissions",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:GetMetricData",
            "cloudwatch:ListMetrics"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogGroupPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:StartQuery",
            "logs:DescribeMetricFilters"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogsPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:GetQueryResults",
            "logs:StopQuery"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSyntheticsPermissions",
        "Effect": "Allow",
        "Action": [
            "synthetics:DescribeCanaries",
            "synthetics:DescribeCanariesLastRun",
            "synthetics:GetCanaryRuns"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsRumPermissions",
        "Effect": "Allow",
        "Action": [
            "rum:BatchCreateRumMetricDefinitions",
            "rum:BatchDeleteRumMetricDefinitions",
            "rum:BatchGetRumMetricDefinitions",
            "rum:GetAppMonitor",
```

```
            "rum:GetAppMonitorData",
            "rum:ListAppMonitors",
            "rum:PutRumMetricsDestination",
            "rum:UpdateRumMetricDefinition"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsXrayPermissions",
        "Effect": "Allow",
        "Action": [
            "xray:GetTraceSummaries"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsPutMetricAlarmPermissions",
        "Effect": "Allow",
        "Action": "cloudwatch:PutMetricAlarm",
        "Resource": [
            "arn:aws:cloudwatch:*:*:alarm:SLO-AttainmentGoalAlarm-*",
            "arn:aws:cloudwatch:*:*:alarm:SLO-WarningAlarm-*",
            "arn:aws:cloudwatch:*:*:alarm:SLI-HealthAlarm-*"
        ]
    },
    {
        "Sid": "CloudWatchApplicationSignalsCreateServiceLinkedRolePermissions",
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals",
        "Condition": {
            "StringLike": {
                "iam:AWSServiceName": "application-signals.cloudwatch.amazonaws.com"
            }
        }
    },
    {
        "Sid": "CloudWatchApplicationSignalsGetRolePermissions",
        "Effect": "Allow",
        "Action": "iam:GetRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
    },
```

```
    {
        "Sid": "CloudWatchApplicationSignalsSnsWritePermissions",
        "Effect": "Allow",
        "Action": [
            "sns:CreateTopic",
            "sns:Subscribe"
        ],
        "Resource": "arn:aws:sns:*:*:cloudwatch-application-signals-*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSnsReadPermissions",
        "Effect": "Allow",
        "Action": "sns:ListTopics",
        "Resource": "*"
    }
 ]
}
```

Para habilitar o Application Signals no Amazon EC2, no Kubernetes ou em arquiteturas personalizadas, consulte Habilitar o Application Signals em outras plataformas com uma configuração personalizada. Para habilitar e gerenciar o Application Signals no Amazon EKS usando o complemento de observabilidade do EKS do Amazon CloudWatch, as permissões apresentadas a seguir são necessárias.

> **Important**
>
> Essas permissões incluem `iam:PassRole` com `Resource "*”` e `eks:CreateAddon` com `Resource “*”`. Essas são permissões avançadas e você deve ter cuidado ao concedê-las.

```
{
"Version": "2012-10-17",
    "Statement": [
        {
"Sid": "CloudWatchApplicationSignalsEksAddonManagementPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:AccessKubernetesApi",
                "eks:CreateAddon",
                "eks:DescribeAddon",
                "eks:DescribeAddonConfiguration",
```

```
                "eks:DescribeAddonVersions",
                "eks:DescribeCluster",
                "eks:DescribeUpdate",
                "eks:ListAddons",
                "eks:ListClusters",
                "eks:ListUpdates",
                "iam:ListRoles",
                "iam:PassRole"
            ],
            "Resource": "*"
        },
        {
"Sid":
 "CloudWatchApplicationSignalsEksCloudWatchObservabilityAddonManagementPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:DeleteAddon",
                "eks:UpdateAddon"
            ],
            "Resource": "arn:aws:eks:*:*:addon/*/amazon-cloudwatch-observability/*"
        }
    ]
}
```

O painel do Application Signals mostra as aplicações do AppRegistry do AWS Service Catalog às quais seus SLOs estão associados. Para visualizar essas aplicações nas páginas de SLO, você deve ter as seguintes permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsTaggingReadPermissions",
            "Effect": "Allow",
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Como operar o Application Signals

Os operadores de serviços que usam o Application Signals para monitorar serviços e SLOs devem ter feito logon em uma conta com as seguintes permissões somente leitura:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsReadOnlyAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-signals:BatchGet*",
                "application-signals:Get*",
                "application-signals:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsGetRolePermissions",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
        },
        {
        "Sid": "CloudWatchApplicationSignalsLogGroupPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:StartQuery",
            "logs:DescribeMetricFilters"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogsPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:GetQueryResults",
            "logs:StopQuery"
        ],
        "Resource": "*"
    },
```

```
        {
            "Sid": "CloudWatchApplicationSignalsAlarmsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsMetricsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsSyntheticsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "synthetics:DescribeCanaries",
                "synthetics:DescribeCanariesLastRun",
                "synthetics:GetCanaryRuns"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsRumReadPermissions",
            "Effect": "Allow",
            "Action": [
                "rum:BatchGetRumMetricDefinitions",
                "rum:GetAppMonitor",
                "rum:GetAppMonitorData",
                "rum:ListAppMonitors"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsXrayReadPermissions",
            "Effect": "Allow",
            "Action": [
                "xray:GetTraceSummaries"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

Para visualizar quais são as aplicações do AppRegistry do AWS Service Catalog às quais seus SLOs estão associados no painel do Application Signals, você deve ter as seguintes permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsTaggingReadPermissions",
            "Effect": "Allow",
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Para verificar se o Application Signals está habilitado no Amazon EKS usando o complemento de observabilidade do EKS do Amazon CloudWatch, você precisa ter as seguintes permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsEksReadPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:ListAddons",
                "eks:ListClusters"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsEksDescribeAddonReadPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:DescribeAddon"
            ],
            "Resource": "arn:aws:eks:*:*:addon/*/amazon-cloudwatch-observability/*"
```

```
        }
    ]
}
```

Habilitação do Application Signals

> ⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Os tópicos desta seção explicam como habilitar o CloudWatch Application Signals em seu ambiente. O Application Signals é compatível com clusters do Amazon EKS com um fluxo de trabalho de configuração usando o console. Além disso, ele é compatível com outras plataformas, incluindo o Amazon EC2, com um processo de configuração personalizado.

Tópicos

Sistemas compatíveis para o Application Signals

> ⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

O Application Signals é compatível e foi testado para o Amazon EKS, Amazon ECS e Amazon EC2. As instruções para a habilitação do Application Signals no Amazon EC2 devem funcionar em qualquer plataforma compatível com o agente do CloudWatch e o AWS Distro para OpenTelemetry, mas as instruções não foram testadas em outras plataformas.

Compatibilidade com Java

O Application Signals é compatível com as aplicações em Java e as mesmas bibliotecas e estruturas em Java que o AWS Distro para OpenTelemetry. Para obter mais informações, consulte Supported libraries, frameworks, application servers, and JVMs.

As versões 8, 11 e 17 da JVM são compatíveis.

Compatibilidade com Python

O Application Signals oferece suporte às mesmas bibliotecas e estruturas que o AWS Distro para OpenTelemetry. Para obter mais informações, consulte Supported packages em opentelemetry-python-contrib.

As versões 3.8 e posteriores do Python são compatíveis.

Antes de habilitar o Application Signals para suas aplicações em Python, esteja ciente das considerações apresentadas a seguir.

- Em algumas aplicações em contêineres, uma variável de ambiente `PYTHONPATH` ausente pode, às vezes, causar falhas na inicialização da aplicação. Para resolver isso, certifique-se de definir a variável de ambiente `PYTHONPATH` para o local do diretório de trabalho da sua aplicação. Isso ocorre devido a um problema conhecido com a instrumentação automática do OpenTelemetry. Para obter mais informações sobre esse problema, consulte Python autoinstrumentation setting of PYTHONPATH is not compliant.
- Para aplicações em Django, existem configurações adicionais necessárias, descritas na documentação do OpenTelemetry em Python.
  - Use o sinalizador `--noreload` para evitar o recarregamento automático.
  - Defina a variável de ambiente `DJANGO_SETTINGS_MODULE` para o local do arquivo `settings.py` da sua aplicação em Django. Isso garante que o OpenTelemetry possa acessar e se integrar adequadamente às suas configurações do Django.

Considerações sobre a compatibilidade com o OpenTelemetry

⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Para integrar as aplicações com o CloudWatch Application Signals, recomendamos remover completamente quaisquer soluções de monitoramento de performance de aplicações existentes da sua aplicação com antecedência. Isso inclui a remoção de códigos e configurações de instrumentação.

Embora o Application Signals use a instrumentação do OpenTelemetry, a compatibilidade com a instrumentação ou com a configuração do OpenTelemetry existente não é garantida. Na melhor das hipóteses, será possível manter algumas das funcionalidades do OpenTelemetry, como as métricas personalizadas. No entanto, certifique-se de realizar a leitura das seções apresentadas a seguir para obter mais detalhes.

Considerações no caso de você já usar o OpenTelemetry

Se você já estiver usando o OpenTelemetry com a aplicação, o restante desta seção contém informações importantes para a obtenção de compatibilidade com o Application Signals.

- Antes de habilitar a aplicação para o Application Signals, você deve remover a injeção de quaisquer outros agentes de instrumentação automática baseados no OpenTelemetry da aplicação. Isso ajuda a evitar conflitos de configuração. É possível continuar usando a instrumentação manual ao usar as APIs do OpenTelemetry que são compatíveis em conjunto com o Application Signals.
- Se você estiver usando uma instrumentação manual para gerar extensões ou métricas personalizadas da aplicação, dependendo da complexidade da instrumentação, habilitar o Application Signals poderá fazer com que a geração de dados seja interrompida ou causar outro comportamento indesejável. É possível tentar usar algumas das configurações disponíveis no OpenTelemetry (exceto as mencionadas posteriormente na tabela desta seção) para manter o comportamento desejado das métricas ou das extensões existentes. Para obter mais informações sobre essas configurações, consulte SDK Configuration na documentação do OpenTelemetry.

  Por exemplo, ao usar a configuração `OTEL_EXPORTER_OTLP_METRICS_ENDPOINT` e uma instância autogerenciada do OpenTelemetry Collector, é possível continuar a enviar as métricas personalizadas para o destino desejado.
- Algumas variáveis de ambiente ou propriedades de sistemas não devem ser usadas com o Application Signals, enquanto outras podem ser usadas, desde que você siga as orientações apresentadas na tabela. Consulte a tabela a seguir para obter detalhes.

Variável de ambiente	Recomendações para o Application Signals
Variáveis de ambiente gerais	
`OTEL_SDK_DISABLED`	Não deve ser definida como `true`.
`OTEL_TRACES_EXPORTER`	Deve ser definida como `otlp`.
`OTEL_EXPORTER_OTLP_ENDPOINT`	Não deve ser usada.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Não deve ser usada.
`OTEL_ATTRIBUTE_COUNT_LIMIT`	Se definida, deve ter uma configuração suficientemente elevada para incluir, aproximadamente, mais dez atributos de extensões que são adicionados pelo CloudWatch Application Signals.
`OTEL_PROPAGATORS`	Se definida, deve incluir `xray` para o rastreamento final.
`OTEL_TRACES_SAMPLER`	Se definida, deve ser `xray` para usar a amostragem centralizada do X-Ray. Para usar a amostragem local, defina a variável como `parentbased_traceidratio` e especifique a taxa de amostragem em `OTEL_TRACES_SAMPLER_ARG` .
`OTEL_TRACES_SAMPLER_ARG`	Se você estiver usando o padrão para a amostra de rastreamento centralizado do X-Ray, essa variável não deverá ser usada. Se você estiver usando a amostragem local, defina a taxa de amostragem nessa variável. Por exemplo, `0.05` para uma taxa de amostragem de 5%.

Variável de ambiente	Recomendações para o Application Signals
Variáveis de ambiente específicas para Java	
`OTEL_JAVA_ENABLED_RESOURCE_PROVIDERS`	Se definida, deve incluir detectores de recursos da AWS.
Variáveis de ambiente específicas para Python	
`OTEL_PYTHON_CONFIGURATOR`	Se usada, deve ser definida como `aws_configurator` .
`OTEL_PYTHON_DISTRO`	Se usada, deve ser definida como `aws_distro` .

Habilitar o Application Signals em clusters do Amazon EKS

⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

O CloudWatch Application Signals é compatível com aplicações em Java e Python que estão em execução em clusters do Amazon EKS. Para habilitar o Application Signals para aplicações em um cluster do Amazon EKS, você tem duas opções:

- Para habilitar o Application Signals para as suas aplicações em um cluster do Amazon EKS existente, use as etapas apresentadas em Habilitar o Application Signals em um cluster do Amazon EKS com seus serviços.
- Para testar o Application Signals em um ambiente que não seja de produção com uma aplicação de amostra, use as instruções apresentadas em Habilitar o Application Signals em um novo cluster do Amazon EKS com uma aplicação de amostra. Esse fluxo de trabalho usa scripts disponibilizados pela AWS para criar um novo cluster do Amazon EKS e instalar uma aplicação de amostra habilitada para o Application Signals. Isso permite visualizar e testar a funcionalidade completa do Application Signals.

Tópicos

- Habilitar o Application Signals em um cluster do Amazon EKS com seus serviços
- Habilitar o Application Signals em um novo cluster do Amazon EKS com uma aplicação de amostra

Habilitar o Application Signals em um cluster do Amazon EKS com seus serviços

⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Para habilitar o CloudWatch Application Signals nas aplicações em um cluster do Amazon EKS existente, use as instruções apresentadas nesta seção.

⚠ Important

Se você já estiver usando o OpenTelemetry com uma aplicação para a qual pretende habilitar o Application Signals, consulte Considerações sobre a compatibilidade com o OpenTelemetry antes de habilitar o Application Signals.

Como habilitar o Application Signals para suas aplicações em um cluster do Amazon EKS existente

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Serviços.
3. Caso ainda não tenha habilitado o Application Signals nessa conta, você deve conceder as permissões necessárias para o Application Signals descobrir seus serviços. Para fazê-lo, siga as etapas apresentadas a seguir. Você precisa fazer isso somente uma vez para a conta.

   a. Escolha Começar a descobrir os serviços.

   b. Marque a caixa de seleção e escolha Começar a descobrir serviços.

      A conclusão dessa etapa pela primeira vez em sua conta cria a função vinculada ao serviço AWSServiceRoleForCloudWatchApplicationSignals. Essa função concede as seguintes permissões ao Application Signals:

      - `xray:GetServiceGraph`
      - `logs:StartQuery`

- `logs:GetQueryResults`
- `cloudwatch:GetMetricData`
- `cloudwatch:ListMetrics`
- `tag:GetResources`

Para obter mais informações sobre essa função, consulte Permissões de perfis vinculados ao serviço para o CloudWatch Application Signals.

4. Escolha Habilitar o Application Signals.
5. Em Especificar plataforma, escolha EKS.
6. Em Selecionar um cluster do EKS, selecione o cluster no qual você deseja habilitar o Application Signals.
7. Se esse cluster ainda não tiver o complemento Amazon CloudWatch Observability do EKS habilitado, você será solicitado a habilitá-lo. Se for esse o caso, faça o seguinte:

   a. Escolha Adicionar o complemento CloudWatch Observability do EKS. O console do Amazon EKS será exibido.

   b. Marque a caixa de seleção para Amazon CloudWatch Observability e escolha Próximo.

      O complemento CloudWatch Observability do EKS habilita o Application Signals e o CloudWatch Container Insights com observabilidade aprimorada para o Amazon EKS. Para obter mais informações sobre o Container Insights, consulte Container Insights.

   c. Selecione a versão mais recente do complemento para a instalação.

   d. Selecione um perfil do IAM para usar com o complemento. Se você escolher Herdar do nó, anexe as permissões adequadas ao perfil do IAM usado pelos nós de processamento. Substitua *my-worker-node-role* pelo perfil do IAM usado por seus nós de processamento do Kubernetes.

      ```
      aws iam attach-role-policy \
      --role-name my-worker-node-role \
      --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
      --policy-arn arn:aws:iam::aws:policy/AWSXRayWriteOnlyAccess
      ```

   e. Caso deseje criar um perfil de serviço para usar o complemento, consulte Instalar o agente do CloudWatch usando o complemento de observabilidade do EKS do Amazon CloudWatch.

f. Escolha Próximo, confirme as informações na tela e escolha Criar.

g. Na próxima tela, escolha Habilitar o CloudWatch Application Signals para retornar ao console do CloudWatch e finalizar o processo.

8. Existem duas opções para habilitar as aplicações para o Application Signals. Para manter a consistência, recomendamos que você escolha uma opção por cluster.

- A opção Console é mais simples. O uso desse método resulta na reinicialização imediata dos seus pods.
- O método Anotar o arquivo de manifesto oferece mais controle sobre quando os pods serão reiniciados e também pode ajudar no gerenciamento do monitoramento de uma forma mais descentralizada se você não desejar centralizá-lo.

Console

A opção Console usa a configuração avançada do complemento Observability do Amazon CloudWatch para o EKS para configurar o Application Signals para seus serviços. Para obter mais informações sobre o complemento, consulte (Opcional) Configuração adicional.

Se você não vir uma lista de workloads e de namespaces, certifique-se de ter as permissões adequadas para visualizá-la neste cluster. Para obter mais informações, consulte Permissões obrigatórias.

É possível monitorar workloads únicas ou namespaces inteiros.

Para monitorar uma única workload:

1. Marque a caixa de seleção ao lado da workload que você deseja monitorar.
2. Selecione a linguagem da workload. Para aplicações em Python, certifique-se de que a aplicação siga os pré-requisitos obrigatórios antes de prosseguir. Para ter mais informações, consulte A aplicação em Python não é iniciada após a habilitação do Application Signals.
3. Escolha Concluído. O complemento Observability do Amazon CloudWatch para o EKS injetará imediatamente SDKs da instrumentação automática do AWS Distro para OpenTelemetry (ADOT) em seus pods e acionará reinicializações de pods para habilitar a coleta de métricas e de rastreamentos das aplicações.

Para monitorar um namespace inteiro:

1. Marque a caixa de seleção ao lado do namespace que você deseja monitorar.
2. Selecione a linguagem da workload. Isso será aplicado a todas as workloads neste namespace, independentemente de estarem implantadas atualmente ou de serem implantadas no futuro. Para aplicações em Python, certifique-se de que a aplicação siga os pré-requisitos obrigatórios antes de prosseguir. Para ter mais informações, consulte A aplicação em Python não é iniciada após a habilitação do Application Signals.
3. Escolha Concluído. O complemento Observability do Amazon CloudWatch para o EKS injetará imediatamente SDKs da instrumentação automática do AWS Distro para OpenTelemetry (ADOT) em seus pods e acionará reinicializações de pods para habilitar a coleta de métricas e de rastreamentos das aplicações.

Para habilitar o Application Signals em outro cluster do Amazon EKS, escolha Habilitar o Application Signals usando a tela Serviços.

Annotate manifest file

No console do CloudWatch, a seção Monitorar serviços explica que você deve adicionar uma anotação a um manifesto em YAML no cluster. A adição dessa anotação instrumenta automaticamente a aplicação para enviar métricas, rastreamentos e logs para o Application Signals.

Você tem duas opções para a anotação:

- Anotar a workload instrumenta automaticamente uma única workload no cluster.
- Anotar o namespace instrumenta automaticamente todas as workloads implantadas no namespace selecionado.

Escolha uma dessas opções e siga as etapas apropriadas:

- Para anotar uma única workload:
  1. Escolha Anotar a workload.
  2. Cole uma das linhas apresentadas a seguir na seção `PodTemplate` do arquivo de manifesto da workload.
     - Para workloads em Java: `annotations: instrumentation.opentelemetry.io/inject-java: "true"`

- Para workloads em Python: `annotations: instrumentation.opentelemetry.io/inject-python: "true"`

  Para aplicações em Python, existem configurações adicionais necessárias. Para ter mais informações, consulte A aplicação em Python não é iniciada após a habilitação do Application Signals.

3. No seu terminal, insira `kubectl apply -f your_deployment_yaml` para aplicar a alteração.

- Para anotar todas as workloads em um namespace:

  1. Escolha Anotar o namespace.
  2. Cole uma das linhas apresentadas a seguir na seção de metadados do arquivo de manifesto do namespace. Se o namespace incluir workloads em Java e em Python, cole ambas as linhas no arquivo de manifesto do namespace.

     - Se houver workloads em Java no namespace: `annotations: instrumentation.opentelemetry.io/inject-java: "true"`
     - Se houver workloads em Python no namespace: `annotations: instrumentation.opentelemetry.io/inject-python: "true"`

       Para aplicações em Python, existem configurações adicionais necessárias. Para ter mais informações, consulte A aplicação em Python não é iniciada após a habilitação do Application Signals.

  3. No seu terminal, insira `kubectl apply -f your_namespace_yaml` para aplicar a alteração.
  4. No seu terminal, insira um comando para reiniciar todos os pods no namespace. Um exemplo de comando para reiniciar as workloads de implantação é `kubectl rollout restart deployment -n namespace_name`.

9. Escolha Visualizar os serviços na conclusão. Isso direciona você para a visualização dos serviços do Application Signals, um local no qual você pode consultar os dados que o Application Signals está coletando. Pode demorar alguns minutos para que os dados sejam exibidos.

   Para habilitar o Application Signals em outro cluster do Amazon EKS, escolha Habilitar o Application Signals usando a tela Serviços.

   Para obter mais informações sobre a visualização dos Serviços, consulte Monitorar a integridade operacional das suas aplicações com o Application Signals.

> **Note**
>
> Identificamos algumas considerações que você deve ter em mente ao habilitar aplicações em Python para o Application Signals. Para ter mais informações, consulte A aplicação em Python não é iniciada após a habilitação do Application Signals.

Habilitar o Application Signals em um novo cluster do Amazon EKS com uma aplicação de amostra

> **Warning**
>
> O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Para testar o CloudWatch Application Signals em uma aplicação de amostra antes de instrumentar suas próprias aplicações com ele, siga as instruções apresentadas nesta seção. Essas instruções usam scripts para ajudar você a criar um cluster do Amazon EKS, instalar uma aplicação de amostra e instrumentá-la para funcionar com o Application Signals.

A aplicação de amostra corresponde a uma aplicação “Pet Clinic” da Spring que é composta por quatro microsserviços. Esses serviços são executados no Amazon EKS e no Amazon EC2 e utilizam os scripts de habilitação do Application Signals para habilitar o cluster com o agente de instrumentação automática em Java ou em Python.

Requisitos

- No momento, o Application Signals monitora somente aplicações em Java e Python.
- Você deve ter a AWS CLI instalada na instância. Recomendamos a versão 2 da AWS CLI, mas a versão 1 também deve funcionar. Para obter mais informações sobre como instalar a AWS CLI, consulte Instalar ou atualizar para a versão mais recente da AWS CLI.
- Os scripts apresentados nesta seção devem ser executados em ambientes do Linux e do macOS. Para as instâncias do Windows, recomendamos usar um ambiente do AWS Cloud9 para a execução desses scripts. Para obter mais informações sobre o AWS Cloud9, consulte What is AWS Cloud9?.
- Instale uma versão compatível do `kubectl`. Você deve usar uma versão do `kubectl` com uma diferença de versão secundária do ambiente de gerenciamento do cluster do Amazon EKS. Por

exemplo, um cliente `kubectl` 1.26 funciona com clusters do Kubernetes nas versões 1.25, 1.26 e 1.27. Se você já tiver um cluster do Amazon EKS, talvez seja necessário configurar as credenciais da AWS para o `kubectl`. Para obter mais informações, consulte Criar ou atualizar um arquivo `kubeconfig` para um cluster do Amazon EKS.

- Instale o `eksctl`. O `eksctl` usa a AWS CLI para interagir com a AWS, o que significa que ele usa as mesmas credenciais da AWS que a AWS CLI. Para obter mais informações, consulte Installing or updating `eksctl`.
- Instale o `jq`. O `jq` é necessário para executar os scripts de habilitação do Application Signals. Para obter mais informações, consulte Download jq.

Etapa 1: fazer download dos scripts

Para fazer download dos scripts para a configuração do CloudWatch Application Signals com uma aplicação de amostra, é possível fazer o download e descompactar o arquivo compactado do projeto do GitHub em uma unidade local ou clonar o projeto do GitHub.

Para clonar o projeto, abra uma janela de terminal e insira o comando Git apresentado a seguir em um determinado diretório de trabalho.

```
git clone https://github.com/aws-observability/application-signals-demo.git
```

Etapa 2: criar e implantar a aplicação de amostra

Para criar e enviar por push as imagens da aplicação de amostra, siga estas instruções.

Etapa 3: implantar e habilitar o Application Signals e a aplicação de amostra

Certifique-se de ter atendido aos requisitos listados em Habilitar o Application Signals em um novo cluster do Amazon EKS com uma aplicação de amostra antes de concluir as etapas apresentadas a seguir.

Como implantar e habilitar o Application Signals e a aplicação de amostra

1. Insira o comando apresentado a seguir no terminal local no qual você descompactou o script de integração. Substitua *`new-cluster-name`* pelo nome que você deseja usar para o novo cluster. Substitua *`region-name`* pelo nome da região da AWS, como `us-west-1`.

   Esse comando configura a aplicação de amostra em execução em um novo cluster do Amazon EKS com o Application Signals habilitado.

```
# assuming the current working directory is 'onboarding'
# this script sets up a new cluster, enables Application Signals, and deploys the
# sample application
cd application-signals-demo/scripts/eks/appsignals/one-step && ./setup.sh new-
cluster-name region-name
```

O script de configuração demora cerca de 30 minutos para ser executado e faz o seguinte:

- Cria um novo cluster do Amazon EKS na região especificada.
- Cria as permissões do IAM necessárias para o Application Signals (`arn:aws:iam::aws:policy/AWSXrayWriteOnlyAccess` e `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy`).
- Habilita o Application Signals ao instalar o agente do CloudWatch e ao instrumentalizar automaticamente a aplicação de amostra para as métricas do CloudWatch e os rastreamentos do X-Ray.
- Implanta a aplicação de amostra Pet Clinic da Spring no mesmo cluster do Amazon EKS.
- Cria cinco canários do CloudWatch Synthetics, denominados `pc-add-vist`, `pc-create-owners`, `pc-visit-pet`, `pc-visit-vet` e `pc-clinic-traffic`. Esses canários serão executados com uma frequência de um minuto para gerar tráfego artificial para a aplicação de amostra e demonstrar como os canários do Synthetics aparecem no Application Signals.
- Cria quatro objetivos de nível de serviço (SLOs) para a aplicação Pet Clinic com os seguintes nomes:
  - Disponibilidade para pesquisar um proprietário
  - Latência para pesquisar um proprietário
  - Disponibilidade para registrar um proprietário
  - Latência para registrar um proprietário
- Cria o perfil do IAM obrigatório com uma política de confiança personalizada que concede ao Application Signals as seguintes permissões:
  - `cloudwatch:PutMetricData`
  - `cloudwatch:GetMetricData`
  - `xray:GetServiceGraph`
  - `logs:StartQuery`
  - `logs:GetQueryResults`

2. (Opcional) Se desejar analisar o código fonte da aplicação de amostra Pet Clinic, será possível encontrá-lo na pasta raiz.

```
- application-signals-demo
  - spring-petclinic-admin-server
  - spring-petclinic-api-gateway
  - spring-petclinic-config-server
  - spring-petclinic-customers-service
  - spring-petclinic-discovery-server
  - spring-petclinic-vets-service
  - spring-petclinic-visits-service
```

3. Para visualizar a implantação da aplicação de amostra Pet Clinic, execute o seguinte comando para localizar o URL:

```
kubectl get ingress
```

Etapa 4: monitorar a aplicação de amostra

Após concluir as etapas apresentadas na seção anterior para criar o cluster do Amazon EKS e implantar a aplicação de amostra, será possível usar o Application Signals para monitorar a aplicação.

> **Note**
>
> Para que o console do Application Signals comece a ser preenchido, algum tráfego deve chegar à aplicação de amostra. Em uma das etapas anteriores, ocorreu a criação de canários do CloudWatch Synthetics que geram tráfego para a aplicação de amostra.

Monitoramento da integridade do serviço

Depois de habilitado, o CloudWatch Application Signals descobre e preenche automaticamente uma lista de serviços sem requerer configurações adicionais.

Como visualizar a lista de serviços descobertos e monitorar a integridade

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Serviços.

3. Para visualizar um serviço, as operações e as dependências, escolha o nome de um dos serviços da lista.

   Essa visualização unificada e centrada em aplicações ajuda a fornecer uma perspectiva completa de como os usuários estão interagindo com o serviço. Isso pode ajudar você a realizar a triagem de problemas, caso ocorram anomalias de performance. Para obter todos os detalhes sobre a visualização de Serviços, consulte Monitorar a integridade operacional das suas aplicações com o Application Signals.

4. Escolha a guia Operações de serviço para consultar as métricas da aplicação padrão para as operações desse serviço. Por exemplo, as operações são as operações de API que o serviço chama.

   Em seguida, para visualizar os gráficos de uma única operação desse serviço, escolha o nome dessa operação.

5. Escolha a guia Dependências para visualizar as dependências que a aplicação tem, juntamente com as métricas importantes da aplicação para cada dependência. As dependências incluem serviços da AWS e serviços de terceiros que a aplicação chama.

6. Para visualizar rastreamentos correlacionados usando a página de detalhes do serviço, escolha um ponto de dados em um dos três gráficos acima da tabela. Isso preencherá um novo painel com rastreamentos filtrados do período. Esses rastreamentos são classificados e filtrados com base no gráfico escolhido. Por exemplo, se você escolher o gráfico Latência, os rastreamentos serão classificados pelo tempo de resposta do serviço.

7. No painel de navegação do console do CloudWatch, escolha SLOs. Você visualizará os SLOs que o script criou para a aplicação de amostra. Para obter mais informações sobre os SLOs, consulte Objetivos de nível de serviço (SLOs).

(Opcional) Etapa 5: realizar a limpeza

Quando terminar de testar o Application Signals, será possível usar um script fornecido pela Amazon para realizar a limpeza e a exclusão dos artefatos criados em sua conta para a aplicação de amostra. Para realizar a limpeza, insira o comando apresentado a seguir. Substitua *new-cluster-name* pelo nome do cluster que você criou para a aplicação de amostra e substitua *region*-name pelo nome da região da AWS, como `us-west-1`.

```
cd application-signals-demo/scripts/eks/appsignals/one-step && ./cleanup.sh new-
cluster-name region-name
```

Habilitar o Application Signals em outras plataformas com uma configuração personalizada

⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Habilite o CloudWatch Application Signals em plataformas diferentes do Amazon EKS usando as etapas de configuração personalizadas que são apresentadas nestas seções. Nessas arquiteturas, você instala e configura o agente do CloudWatch e o AWS Distro para OpenTelemetry.

Nessas arquiteturas, o Application Signals não descobre automaticamente os nomes dos seus serviços ou os clusters ou hosts deles. Você deve especificar esses nomes durante a configuração personalizada, e os nomes especificados serão exibidos nos painéis do Application Signals.

Tópicos

Uso de uma configuração personalizada para habilitar o Application Signals no Amazon ECS

⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Use essas instruções de configuração personalizadas para integrar suas aplicação no Amazon ECS para o CloudWatch Application Signals. Você instala e configura o agente do CloudWatch e o AWS Distro para OpenTelemetry.

Em clusters do Amazon ECS, o Application Signals não descobre automaticamente os nomes dos seus serviços ou dos clusters em que eles são executados. Você deve especificar esses nomes durante a configuração personalizada, e os nomes especificados serão exibidos nos painéis do Application Signals.

> **Important**
>
> Somente o modo de rede `awsvpc` é compatível.

Etapa 1: habilitar o Application Signals em sua conta

Caso ainda não tenha habilitado o Application Signals nessa conta, você deve conceder as permissões necessárias para o Application Signals descobrir seus serviços. Para fazê-lo, siga as etapas apresentadas a seguir. Você precisa fazer isso somente uma vez para a conta.

Como habilitar o Application Signals para suas aplicações

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Serviços.
3. Escolha Começar a descobrir os serviços.
4. Marque a caixa de seleção e escolha Começar a descobrir serviços.

   A conclusão dessa etapa pela primeira vez em sua conta cria a função vinculada ao serviço AWSServiceRoleForCloudWatchApplicationSignals. Essa função concede as seguintes permissões ao Application Signals:

   - `xray:GetServiceGraph`
   - `logs:StartQuery`
   - `logs:GetQueryResults`
   - `cloudwatch:GetMetricData`
   - `cloudwatch:ListMetrics`
   - `tag:GetResources`

   Para obter mais informações sobre essa função, consulte Permissões de perfis vinculados ao serviço para o CloudWatch Application Signals.

Etapa 2: criar perfis do IAM

É necessário criar dois perfis do IAM. Se você já criou esses perfis, talvez seja necessário adicionar permissões a eles.

- Função de tarefa do ECS: os contêineres usam essa função para serem executados. As permissões devem ser as necessárias para as aplicações, acrescidas de CloudWatchAgentServerPolicy e AWSXRayWriteOnlyAccess.
- Função de execução de tarefa do ECS: o Amazon ECS usa essa função para iniciar e executar os contêineres. Se você já criou essa função, anexe as políticas AmazonSSMReadOnlyAccess, AmazonECSTaskExecutionRolePolicy e CloudWatchAgentServerPolicy a ela.

  Se precisar armazenar mais dados sigilosos para uso do Amazon ECS, consulte Especificar dados sigilosos para obter mais informações.

Para obter mais informações sobre a criação de funções do IAM, consulte Criar funções do IAM.

Etapa 3: preparar a configuração do agente do CloudWatch

Primeiro, prepare a configuração do agente com o Application Signals habilitado. Para fazer isso, crie um arquivo local denominado `/tmp/ecs-cwagent.json`.

```
{
  "traces": {
    "traces_collected": {
      "app_signals": {}
    }
  },
  "logs": {
    "metrics_collected": {
      "app_signals": {}
    }
  }
}
```

Em seguida, faça o upload dessa configuração no SSM Parameter Store. Para fazer isso, insira o comando a seguir. No arquivo, substitua *$REGION* pelo nome real da sua região.

```
aws ssm put-parameter \
--name "ecs-cwagent" \
--type "String" \
--value "`cat /tmp/ecs-cwagent.json`" \
--region "$REGION"
```

Etapa 4: instrumentalizar a aplicação com o agente do CloudWatch

A próxima etapa corresponde à instrumentação da sua aplicação para o CloudWatch Application Signals.

Java

Como instrumentalizar a aplicação no Amazon ECS com o agente do CloudWatch

1. Primeiro, especifique uma associação de montagem. O volume será usado para compartilhar arquivos entre contêineres nas próximas etapas. Você usará essa associação de montagem posteriormente nesse procedimento.

```
"volumes": [
  {
    "name": "opentelemetry-auto-instrumentation"
  }
]
```

2. Adicione uma definição de arquivo associado para o agente do CloudWatch. Para fazer isso, anexe um novo contêiner, denominado `ecs-cwagent`, à definição de tarefa da aplicação. Substitua *$REGION* pelo nome real da sua região. Realize a substituição pelo caminho para a imagem de contêiner mais recente do CloudWatch no Amazon Elastic Container Registry. Para obter mais informações, consulte cloudwatch-agent no Amazon ECR.

```
{
  "name": "ecs-cwagent",
  "image": "$IMAGE",
  "essential": true,
  "secrets": [
    {
      "name": "CW_CONFIG_CONTENT",
      "valueFrom": "ecs-cwagent"
    }
  ],
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/ecs-cwagent",
      "awslogs-region": "$REGION",
      "awslogs-stream-prefix": "ecs"
```

```
    }
  }
}
```

3. Anexe um novo contêiner `init` à definição de tarefa da aplicação. Substitua *`$IMAGE`* pela imagem mais recente do repositório de imagens do AWS Distro para OpenTelemetry do Amazon ECR.

```
{
  "name": "init",
  "image": "$IMAGE",
  "essential": false,
  "command": [
    "cp",
    "/javaagent.jar",
    "/otel-auto-instrumentation/javaagent.jar"
  ],
  "mountPoints": [
    {
      "sourceVolume": "opentelemetry-auto-instrumentation",
      "containerPath": "/otel-auto-instrumentation",
      "readOnly": false
    }
  ]
}
```

4. Adicione as variáveis de ambiente apresentadas a seguir ao contêiner da aplicação. Para obter mais informações, consulte

Variável de ambiente	Configuração para habilitação do Application Signals
`OTEL_RESOURCE_ATTRIBUTES`	Substitua `$SVC_NAME` pelo nome da aplicação. Essa variável será exibida como o nome da aplicação nos painéis do Application Signals. Substitua `$HOST_ENV` pelo ambiente de host no qual a aplicação está em execução. Essa variável será exibida como

Variável de ambiente	Configuração para habilitação do Application Signals
	o ambiente Hospedado em da aplicação nos painéis do Application Signals.
`OTEL_AWS_APP_SIGNALS_ENABLED`	Defina como `true` para habilitar o SpanMetricsProcessor do Application Signals.
`OTEL_METRICS_EXPORTER`	Defina como `none` para desabilitar outros exportadores de métricas.
`OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`	Defina como `http://127.0.0.1:4315` para enviar métricas para o arquivo associado do CloudWatch.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Defina como `http://127.0.0.1:4315` para enviar rastreamentos para o arquivo associado do CloudWatch.
`OTEL_TRACES_SAMPLER`	Defina o X-Ray como o gerador de amostras de rastreamentos.
`OTEL_PROPAGATORS`	Adicione o X-Ray como um dos propagadores.
`JAVA_TOOL_OPTIONS`	Injete o agente em Java do AWS Distro para OpenTelemetry.

5. Monte o volume `opentelemetry-auto-instrumentation` definido na etapa 1 deste procedimento.

   Para uma aplicação em Java, use o seguinte:

```
{
  "name": "app",
   ...
  "environment": [
    {
      "name": "OTEL_RESOURCE_ATTRIBUTES",
```

```
      "value": "aws.hostedin.environment=$HOST_ENV,service.name=$SVC_NAME"
    },
    {
      "name": "OTEL_AWS_APP_SIGNALS_ENABLED",
      "value": "true"
    },
    {
      "name": "OTEL_METRICS_EXPORTER",
      "value": "none"
    },
    {
      "name": "JAVA_TOOL_OPTIONS",
      "value": " -javaagent:/otel-auto-instrumentation/javaagent.jar"
    },
    {
      "name": "OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT",
      "value": "http://127.0.0.1:4315"
    },
    {
      "name": "OTEL_TRACES_SAMPLER",
      "value": "xray"
    },
    {
      "name": "OTEL_EXPORTER_OTLP_TRACES_ENDPOINT",
      "value": "http://127.0.0.1:4315"
    },
    {
      "name": "OTEL_PROPAGATORS",
      "value": "tracecontext,baggage,b3,xray"
    }
  ],
  "mountPoints": [
    {
      "sourceVolume": "opentelemetry-auto-instrumentation",
      "containerPath": "/otel-auto-instrumentation",
      "readOnly": false
    }
  ]
}
```

Python

Antes de habilitar o Application Signals para suas aplicações em Python, esteja ciente das considerações apresentadas a seguir.

- Em algumas aplicações em contêineres, uma variável de ambiente `PYTHONPATH` ausente pode, às vezes, causar falhas na inicialização da aplicação. Para resolver isso, certifique-se de definir a variável de ambiente `PYTHONPATH` para o local do diretório de trabalho da sua aplicação. Isso ocorre devido a um problema conhecido com a instrumentação automática do OpenTelemetry. Para obter mais informações sobre esse problema, consulte Python autoinstrumentation setting of PYTHONPATH is not compliant.
- Para aplicações em Django, existem configurações adicionais necessárias, descritas na documentação do OpenTelemetry em Python.
  - Use o sinalizador `--noreload` para evitar o recarregamento automático.
  - Defina a variável de ambiente `DJANGO_SETTINGS_MODULE` para o local do arquivo `settings.py` da sua aplicação em Django. Isso garante que o OpenTelemetry possa acessar e se integrar adequadamente às suas configurações do Django.

Como instrumentalizar a aplicação em Python no Amazon ECS com o agente do CloudWatch

1. Primeiro, especifique uma associação de montagem. O volume será usado para compartilhar arquivos entre contêineres nas próximas etapas. Você usará essa associação de montagem posteriormente nesse procedimento.

```
"volumes": [
  {
    "name": "opentelemetry-auto-instrumentation-python"
  }
]
```

2. Adicione uma definição de arquivo associado para o agente do CloudWatch. Para fazer isso, anexe um novo contêiner, denominado `ecs-cwagent`, à definição de tarefa da aplicação. Substitua *`$REGION`* pelo nome real da sua região. Realize a substituição pelo caminho para a imagem de contêiner mais recente do CloudWatch no Amazon Elastic Container Registry. Para obter mais informações, consulte cloudwatch-agent no Amazon ECR.

```
{
  "name": "ecs-cwagent",
```

```
  "image": "$IMAGE",
  "essential": true,
  "secrets": [
    {
      "name": "CW_CONFIG_CONTENT",
      "valueFrom": "ecs-cwagent"
    }
  ],
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/ecs-cwagent",
      "awslogs-region": "$REGION",
      "awslogs-stream-prefix": "ecs"
    }
  }
}
```

3. Anexe um novo contêiner `init` à definição de tarefa da aplicação. Substitua *$IMAGE* pela imagem mais recente do repositório de imagens do AWS Distro para OpenTelemetry do Amazon ECR.

```
{
    "name": "init",
    "image": "$IMAGE",
    "essential": false,
    "command": [
        "cp",
        "-a",
        "/autoinstrumentation/.",
        "/otel-auto-instrumentation-python"
    ],
    "mountPoints": [
        {
            "sourceVolume": "opentelemetry-auto-instrumentation-python",
            "containerPath": "/otel-auto-instrumentation-python",
            "readOnly": false
        }
    ]
}
```

4. Adicione as variáveis de ambiente apresentadas a seguir ao contêiner da aplicação. Para obter mais informações, consulte

Variável de ambiente	Configuração para habilitação do Application Signals
`OTEL_RESOURCE_ATTRIBUTES`	Substitua `$SVC_NAME` pelo nome da aplicação. Essa variável será exibida como o nome da aplicação nos painéis do Application Signals. Substitua `$HOST_ENV` pelo ambiente de host no qual a aplicação está em execução. Essa variável será exibida como o ambiente Hospedado em da aplicação nos painéis do Application Signals.
`OTEL_AWS_APP_SIGNALS_ENABLED`	Defina como `true` para habilitar o SpanMetricsProcessor do Application Signals.
`OTEL_METRICS_EXPORTER`	Defina como `none` para desabilitar outros exportadores de métricas.
`OTEL_EXPORTER_OTLP_PROTOCOL`	Defina como `http/protobuf` para enviar métricas e rastreamentos ao CloudWatch usando HTTP.
`OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`	Defina como `http://127.0.0.1:4316/v1/metrics` para enviar métricas para o arquivo associado do CloudWatch.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Defina como `http://127.0.0.1:4316/v1/traces` para enviar rastreamentos para o arquivo associado do CloudWatch.

Variável de ambiente	Configuração para habilitação do Application Signals
OTEL_TRACES_SAMPLER	Defina o X-Ray como o gerador de amostras de rastreamentos.
OTEL_PROPAGATORS	Adicione o X-Ray como um dos propagadores.
OTEL_PYTHON_DISTRO	Defina como aws_distro para usar a instrumentação do ADOT em Python.
OTEL_PYTHON_CONFIGURATOR	Defina como aws_configuration para usar a configuração do ADOT em Python.
PYTHONPATH	Substitua $APP_PATH pelo local do diretório de trabalho da aplicação no contêiner. Isso é necessário para que o interpretador em Python localize os módulos da aplicação.
DJANGO_SETTINGS_MODULE	Obrigatória somente para aplicações em Django. Defina-a como o local do arquivo settings.py da sua aplicação em Django. Substitua $PATH_TO_SETTINGS .

5. Monte o volume opentelemetry-auto-instrumentation-python definido na etapa 1 deste procedimento.

   Para uma aplicação em Python, use o seguinte:

```
{
    "name": "app",
    ...
    "environment": [
        {
            "name": "PYTHONPATH",
            "value": "/otel-auto-instrumentation-python/opentelemetry/
instrumentation/auto_instrumentation:$APP_PATH:/otel-auto-instrumentation-
python"
```

```
        },
        {
            "name": "OTEL_EXPORTER_OTLP_PROTOCOL",
            "value": "http/protobuf"
        },
        {
            "name": "OTEL_TRACES_SAMPLER",
            "value": "xray"
        },
        {
            "name": "OTEL_TRACES_SAMPLER_ARG",
            "value": "endpoint=http://localhost:2000"
        },
        {
            "name": "OTEL_LOGS_EXPORTER",
            "value": "none"
        },
        {
            "name": "OTEL_PYTHON_DISTRO",
            "value": "aws_distro"
        },
        {
            "name": "OTEL_PYTHON_CONFIGURATOR",
            "value": "aws_configurator"
        },
        {
            "name": "OTEL_EXPORTER_OTLP_TRACES_ENDPOINT",
            "value": "http://localhost:4316/v1/traces"
        },
        {
            "name": "OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT",
            "value": "http://localhost:4316/v1/metrics"
        },
        {
            "name": "OTEL_METRICS_EXPORTER",
            "value": "none"
        },
        {
            "name": "OTEL_AWS_APP_SIGNALS_ENABLED",
            "value": "true"
        },
        {
            "name": "OTEL_RESOURCE_ATTRIBUTES",
            "value": "aws.hostedin.environment=$HOST_ENV,service.name=$SVC_NAM"
```

```
        },
        {
            "name": "DJANGO_SETTINGS_MODULE",
            "value": "$PATH_TO_SETTINGS.settings"
        }
    ],
    "mountPoints": [
        {
            "sourceVolume": "opentelemetry-auto-instrumentation-python",
            "containerPath": "/otel-auto-instrumentation-python",
            "readOnly": false
        }
    ]
}
```

Etapa 5: implantar a aplicação

Crie uma nova revisão da sua definição de tarefa e implante-a no cluster da aplicação. Você deverá visualizar três contêineres na tarefa recém-criada:

- `init`
- `ecs-cwagent`
- `app`

Usar uma configuração personalizada para habilitar o Application Signals no Amazon EC2 e em outras plataformas

⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Para as aplicações em execução no Amazon EC2 e em outras arquiteturas diferentes do Amazon EKS, você instala e configura o agente do CloudWatch e o AWS Distro para OpenTelemetry. Nessas arquiteturas habilitadas com uma configuração personalizada do Application Signals, o Application Signals não descobre automaticamente os nomes dos seus serviços ou dos hosts ou clusters em que eles são executados. Você deve especificar esses nomes durante a configuração personalizada, e os nomes especificados serão exibidos nos painéis do Application Signals.

As etapas apresentadas a seguir foram testadas em instâncias do Amazon EC2, mas também devem funcionar em outras arquiteturas compatíveis com o AWS Distro para OpenTelemetry.

Requisitos

- Para obter suporte para o Application Signals, é necessário usar a versão mais recente do agente do CloudWatch e do agente do AWS Distro para OpenTelemetry.
- Você deve ter a AWS CLI instalada na instância. Recomendamos a versão 2 da AWS CLI, mas a versão 1 também deve funcionar. Para obter mais informações sobre como instalar a AWS CLI, consulte Instalar ou atualizar para a versão mais recente da AWS CLI.

> **Important**
>
> Se você já estiver usando o OpenTelemetry com uma aplicação para a qual pretende habilitar o Application Signals, consulte Considerações sobre a compatibilidade com o OpenTelemetry antes de habilitar o Application Signals.

Etapa 1: habilitar o Application Signals em sua conta

Caso ainda não tenha habilitado o Application Signals nessa conta, você deve conceder as permissões necessárias para o Application Signals descobrir seus serviços. Para fazê-lo, siga as etapas apresentadas a seguir. Você precisa fazer isso somente uma vez para a conta.

Como habilitar o Application Signals para suas aplicações

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Serviços.
3. Escolha Começar a descobrir os serviços.
4. Marque a caixa de seleção e escolha Começar a descobrir serviços.

   A conclusão dessa etapa pela primeira vez em sua conta cria a função vinculada ao serviço AWSServiceRoleForCloudWatchApplicationSignals. Essa função concede as seguintes permissões ao Application Signals:

   - `xray:GetServiceGraph`
   - `logs:StartQuery`
   - `logs:GetQueryResults`

- `cloudwatch:GetMetricData`
- `cloudwatch:ListMetrics`
- `tag:GetResources`

Para obter mais informações sobre essa função, consulte Permissões de perfis vinculados ao serviço para o CloudWatch Application Signals.

Etapa 2: fazer download e iniciar o agente do CloudWatch

Como instalar o agente do CloudWatch como parte da habilitação do Application Signals em uma instância do Amazon EC2

1. Faça download da versão mais recente do agente do CloudWatch para a instância. Se a instância já tiver o agente do CloudWatch instalado, talvez seja necessário atualizá-lo. Somente as versões do agente liberadas em 30 de novembro de 2023, ou após essa data, oferecem suporte ao CloudWatch Application Signals.

   Para obter informações sobre como fazer download do agente do CloudWatch, consulte Baixar o pacote do atendente do CloudWatch.
2. Antes de iniciar o agente do CloudWatch, configure-o para habilitar o Application Signals. O exemplo apresentado a seguir corresponde a uma configuração do agente do CloudWatch que habilita o Application Signals para as métricas e os rastreamentos em um host do EC2.

   É possível criar esse arquivo ao inserir o seguinte comando:

   ```
   vim amazon-cloudwatch-agent.json
   ```

   Adicione o apresentado a seguir como o conteúdo desse arquivo.

   ```
   {
     "traces": {
       "traces_collected": {
         "app_signals": {}
       }
     },
     "logs": {
       "metrics_collected": {
         "app_signals": {}
   ```

```
    }
  }
}
```

3. Anexe as políticas do IAM CloudWatchAgentServerPolicy e AWSXrayWriteOnlyAccess ao perfil do IAM da sua instância do Amazon EC2.

   a. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.

   b. Escolha Perfis e localize o perfil usado pela sua instância do Amazon EC2. Em seguida, escolha o nome desse perfil.

   c. Na guia Permissões, escolha Adicionar permissões e, em seguida, Anexar políticas.

   d. Localize CloudWatchAgentServerPolicy. Use a caixa de pesquisa, se necessário. Em seguida, marque a caixa de seleção dessa política e escolha Adicionar permissões.

   e. Localize AWSXrayWriteOnlyAccess. Use a caixa de pesquisa, se necessário. Em seguida, marque a caixa de seleção dessa política e escolha Adicionar permissões.

4. Inicie o agente do CloudWatch ao inserir os comandos apresentados a seguir. Substitua *agent-config-file-path* pelo caminho para o arquivo de configuração do agente do CloudWatch, como `./amazon-cloudwatch-agent.json`. Você deve incluir o prefixo `file:`, conforme mostrado.

```
export CONFIG_FILE_PATH=./amazon-cloudwatch-agent.json
```

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \
-a fetch-config \
-m ec2 -s -c file:$CONFIG_FILE_PATH
```

Etapa 3: instrumentalizar a aplicação e iniciá-la

A próxima etapa corresponde à instrumentação da sua aplicação para o CloudWatch Application Signals.

Java

Como instrumentalizar as aplicações em Java como parte da habilitação do Application Signals em uma instância do Amazon EC2

1. Faça download da versão mais recente do agente de instrumentação automática em Java do AWS Distro para OpenTelemetry. É possível fazer download da versão mais recente ao usar este link. É possível visualizar informações sobre todas as versões liberadas em aws-otel-java-instrumentation Releases.
2. Para otimizar os benefícios do Application Signals, use as variáveis de ambiente para fornecer informações adicionais antes de iniciar a aplicação. Essas informações serão exibidas nos painéis do Application Signals.

   a. Para a variável `OTEL_RESOURCE_ATTRIBUTES`, especifique as seguintes informações como pares de chave/valor:

      - `aws.hostedin.environment` define o ambiente em que a aplicação é executada. Essa variável será exibida como o ambiente Hospedado em da aplicação nos painéis do Application Signals. Essa chave de atributo é usada somente pelo Application Signals e é convertida em anotações de rastreamentos do X-Ray e em dimensões de métricas do CloudWatch. Se você não fornecer um valor para essa chave, o padrão `Generic` será usado.
      - `service.name` define o nome do serviço. Isso será exibido como o nome do serviço para a aplicação nos painéis do Application Signals. Se você não fornecer um valor para essa chave, o padrão `unknown_service` será usado.

   b. Para a variável `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`, especifique o URL do endpoint de base para o qual os rastreamentos serão exportados. O agente do CloudWatch mostra 4315 como a porta OLTP. No Amazon EC2, como as aplicações se comunicam com o agente do CloudWatch local, é necessário definir esse valor como `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4315`.

   c. Para a variável `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`, especifique o URL do endpoint de base para o qual as métricas serão exportadas. O agente do CloudWatch mostra 4315 como a porta OLTP. No Amazon EC2, como as aplicações se comunicam com o agente do CloudWatch local, é necessário definir esse valor como `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4315`.

d. Para a variável `JAVA_TOOL_OPTIONS`, especifique o caminho no qual o agente de instrumentação automática em Java do AWS Distro para OpenTelemetry está armazenado.

```
export JAVA_TOOL_OPTIONS=' -javaagent:$ADOT_AGENT_PATH'
```

Por exemplo:

```
export ADOT_AGENT_PATH=./aws-opentelemetry-agent.jar
```

e. Para a variável `OTEL_METRICS_EXPORTER`, recomendamos definir o valor como `none`. Isso desabilita outros exportadores de métricas para que somente o exportador do Application Signals seja usado.

f. Para a variável `OTEL_AWS_APP_SIGNALS_ENABLED`, habilite o SpanMetricProcessor (SMP) ao configurar `OTEL_AWS_APP_SIGNALS_ENABLED` como `true`. Isso gera métricas do Application Signals usando os rastreamentos.

3. Inicie a aplicação com as variáveis de ambiente discutidas na etapa anterior. Veja a seguir um exemplo de um script inicial.

```
JAVA_TOOL_OPTIONS=' -javaagent:$ADOT_AGENT_PATH' \
OTEL_METRICS_EXPORTER=none \
OTEL_AWS_APP_SIGNALS_ENABLED=true \
OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4315 \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4315 \
OTEL_RESOURCE_ATTRIBUTES=aws.hostedin.environment=$YOUR_HOST_ENV,service.name=
$YOUR_SVC_NAME \
java -jar $MY_JAVA_APP.jar
```

Python

Como instrumentalizar as aplicações em Python como parte da habilitação do Application Signals em uma instância do Amazon EC2

1. Faça download da versão mais recente do agente de instrumentação automática do AWS Distro para OpenTelemetry em Python. Instale-o executando o seguinte comando da .

```
pip install aws-opentelemetry-distro
```

É possível visualizar informações sobre todas as versões lançadas em AWS Distro for OpenTelemetry Python instrumentation.

2. Para otimizar os benefícios do Application Signals, use as variáveis de ambiente para fornecer informações adicionais antes de iniciar a aplicação. Essas informações serão exibidas nos painéis do Application Signals.

   a. Para a variável `OTEL_RESOURCE_ATTRIBUTES`, especifique as seguintes informações como pares de chave/valor:

      - `aws.hostedin.environment` define o ambiente em que a aplicação é executada. Essa variável será exibida como o ambiente Hospedado em da aplicação nos painéis do Application Signals. Essa chave de atributo é usada somente pelo Application Signals e é convertida em anotações de rastreamentos do X-Ray e em dimensões de métricas do CloudWatch. Se você não fornecer um valor para essa chave, o padrão `Generic` será usado.
      - `service.name` define o nome do serviço. Isso será exibido como o nome do serviço para a aplicação nos painéis do Application Signals. Se você não fornecer um valor para essa chave, o padrão `unknown_service` será usado.

   b. Para a variável `OTEL_EXPORTER_OTLP_PROTOCOL`, especifique `http/protobuf` para exportar dados de telemetria por HTTP para os endpoints do agente do CloudWatch listados nas etapas a seguir.

   c. Para a variável `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`, especifique o URL do endpoint de base para o qual os rastreamentos serão exportados. O agente do CloudWatch expõe 4316 como a porta OLTP por HTTP. No Amazon EC2, como as aplicações se comunicam com o agente do CloudWatch local, é necessário definir esse valor como `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4316/v1/traces`.

   d. Para a variável `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`, especifique o URL do endpoint de base para o qual as métricas serão exportadas. O agente do CloudWatch expõe 4316 como a porta OLTP por HTTP. No Amazon EC2, como as aplicações se comunicam com o agente do CloudWatch local, é necessário definir esse valor como `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4316/v1/metrics`.

e. Para a variável `OTEL_METRICS_EXPORTER`, recomendamos definir o valor como `none`. Isso desabilita outros exportadores de métricas para que somente o exportador do Application Signals seja usado.

f. Para a variável `OTEL_AWS_APP_SIGNALS_ENABLED`, habilite o SpanMetricProcessor ao definir `OTEL_AWS_APP_SIGNALS_ENABLED` como `true`. Isso gera métricas do Application Signals usando os rastreamentos.

3. Inicie a aplicação com as variáveis de ambiente discutidas na etapa anterior. Veja a seguir um exemplo de um script inicial.

   - Substitua `$HOST_ENV` pelo ambiente de host no qual a aplicação está em execução. Essa variável será exibida como o ambiente Hospedado em para a aplicação nos painéis do Application Signals.
   - Substitua `$SVC_NAME` pelo nome da aplicação. Essa variável será exibida como o nome da aplicação nos painéis do Application Signals.
   - Substitua `$PYTHON_APP` pelo local e pelo nome da aplicação.

```
OTEL_METRICS_EXPORTER=none \
OTEL_LOGS_EXPORTER=none \
OTEL_AWS_APP_SIGNALS_ENABLED=true \
OTEL_PYTHON_DISTRO=aws_distro \
OTEL_PYTHON_CONFIGURATOR=aws_configurator \
OTEL_EXPORTER_OTLP_PROTOCOL=http/protobuf \
OTEL_TRACES_SAMPLER=xray \
OTEL_TRACES_SAMPLER_ARG="endpoint=http://localhost:2000" \
OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4316/v1/metrics \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4316/v1/traces \
OTEL_RESOURCE_ATTRIBUTES=aws.hostedin.environment=$HOST_ENV,service.name=
$SVC_NAME \
opentelemetry-instrument python $PYTHON_APP.py
```

Antes de habilitar o Application Signals para suas aplicações em Python, esteja ciente das considerações apresentadas a seguir.

- Em algumas aplicações em contêineres, uma variável de ambiente `PYTHONPATH` ausente pode, às vezes, causar falhas na inicialização da aplicação. Para resolver isso, certifique-se de definir a variável de ambiente `PYTHONPATH` para o local do diretório de trabalho da sua aplicação. Isso ocorre devido a um problema conhecido com a instrumentação

automática do OpenTelemetry. Para obter mais informações sobre esse problema, consulte Python autoinstrumentation setting of PYTHONPATH is not compliant.

- Para aplicações em Django, existem configurações adicionais necessárias, descritas na documentação do OpenTelemetry em Python.
  - Use o sinalizador `--noreload` para evitar o recarregamento automático.
  - Defina a variável de ambiente `DJANGO_SETTINGS_MODULE` para o local do arquivo `settings.py` da sua aplicação em Django. Isso garante que o OpenTelemetry possa acessar e se integrar adequadamente às suas configurações do Django.

Solução de problemas relacionados à instalação do Application Signals

⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Esta seção contém dicas de solução de problemas para o CloudWatch Application Signals.

Tópicos

A aplicação não inicia após a habilitação do Application Signals

Se a aplicação em um cluster do Amazon EKS não iniciar após a habilitação do Application Signals no cluster, verifique o seguinte:

- Verifique se a aplicação foi instrumentada por outra solução de monitoramento. O Application Signals não é compatível com a coexistência com outras soluções de instrumentação.
- Confirme se a aplicação atende aos requisitos de compatibilidade para o uso do Application Signals. Para obter mais informações, consulte Sistemas compatíveis para o Application Signals .

- Se não foi possível usar a aplicação para realizar a extração dos artefatos do Application Signals, como o agente em Java ou em Python do AWS Distro para OpenTelemetery e as imagens do agente do CloudWatch, pode ser um problema de rede.

Para mitigar o problema, remova a anotação `instrumentation.opentelemetry.io/inject-java: "true"` ou `instrumentation.opentelemetry.io/inject-python: "true"` do manifesto de implantação da aplicação e implante-a novamente. Em seguida, verifique se a aplicação está funcionando.

A aplicação em Python não é iniciada após a habilitação do Application Signals

É um problema conhecido na instrumentação automática do OpenTelemetry que uma variável de ambiente `PYTHONPATH` ausente, às vezes, pode causar falha na inicialização da aplicação. Para resolvê-lo, certifique-se de definir a variável de ambiente `PYTHONPATH` para o local do diretório de trabalho da sua aplicação. Para obter mais informações sobre esse problema, consulte Python autoinstrumentation setting of PYTHONPATH is not compliant with Python's module resolution behavior, breaking Django applications.

Para aplicações em Django, existem configurações adicionais necessárias, descritas na documentação do OpenTelemetry em Python.

- Use o sinalizador `--noreload` para evitar o recarregamento automático.
- Defina a variável de ambiente `DJANGO_SETTINGS_MODULE` para o local do arquivo `settings.py` da sua aplicação em Django. Isso garante que o OpenTelemetry possa acessar e se integrar adequadamente às suas configurações do Django.

Os dados de telemetria estão ausentes no CloudWatch e no X-Ray

Se as métricas ou os rastreamentos estiverem ausentes nos painéis do Application Signals, as informações apresentadas a seguir podem ser as causas. Investigue essas causas somente se você tiver aguardado 15 minutos para que o Application Signals coletasse e exibisse dados desde a última atualização.

- Certifique-se de que a biblioteca e a estrutura que você está usando sejam compatíveis com o agente em Java do ADOT. Para obter mais informações, consulte Libraries / Frameworks.
- Certifique-se de que o agente do CloudWatch esteja em execução. Primeiro, verifique o status dos pods do agente do CloudWatch e certifique-se de que todos estejam com o status `Running`.

```
kubectl -n amazon-cloudwatch get pods.
```

Adicione o conteúdo apresentado a seguir ao arquivo de configuração do agente do CloudWatch para habilitar os logs de depuração e, em seguida, reinicie o agente.

```
"agent": {
>>>>>>> streams
  "region": "${REGION}",
  "debug": true
},
```

Em seguida, verifique se ocorreram erros nos pods do agente do CloudWatch.

- Verifique se há problemas de configuração com o agente do CloudWatch. Confirme se o conteúdo apresentado a seguir ainda está no arquivo de configuração do agente do CloudWatch e se o agente foi reiniciado desde que houve essa adição.

```
"agent": {
  "region": "${REGION}",
  "debug": true
},
```

Em seguida, verifique os logs de depuração do OpenTelemetry em busca de mensagens de erro, como `ERROR io.opentelemetry.exporter.internal.grpc.OkHttpGrpcExporter - Failed to export ...`. É possível que essas mensagens indiquem o problema.

Se isso não resolver o problema, realize o despejo e verifique as variáveis de ambiente com nomes que começam com `OTEL_` ao descrever o pod com o comando `kubectl describe pod`.

- Para habilitar que o OpenTelemetry em Python depure o registro em log, defina a variável de ambiente `OTEL_PYTHON_LOG_LEVEL` para `debug` e implante a aplicação novamente.
- Verifique se há permissões incorretas ou insuficientes para a exportação de dados do agente do CloudWatch. Se você visualizar mensagens de `Access Denied` nos logs do agente do CloudWatch, esse pode ser o problema. É possível que as permissões aplicadas quando você instalou o agente do CloudWatch tenham sido alteradas ou revogadas posteriormente.
- Verifique se há um problema relacionado ao AWS Distro para OpenTelemetry (ADOT) ao gerar dados de telemetria.

Certifique-se de que as anotações de instrumentação `instrumentation.opentelemetry.io/inject-java` e `sidecar.opentelemetry.io/inject-java` estejam aplicadas à implantação da aplicação e que o valor seja `true`. Sem essas anotações, os pods da aplicação não serão instrumentados, mesmo que o complemento do ADOT esteja instalado corretamente.

Em seguida, verifique se o contêiner `Init` está aplicado na aplicação e se o estado `Ready` é `True`. Se o contêiner `init` não estiver pronto, consulte o status para saber o motivo.

Se o problema persistir, realize o procedimento a seguir para habilitar o registro em log de depuração no SDK para Java do OpenTelemetry. Em seguida, procure por mensagens que comecem com `ERROR io.telemetry`.

Para ativar o registro em log de depuração, defina a variável de ambiente `OTEL_JAVAAGENT_DEBUG` como verdadeira e reimplante a aplicação.

- O exportador de métricas ou de extensões pode estar descartando dados. Para descobrir se isso está ocorrendo, verifique o log da aplicação em busca de mensagens que incluam `Failed to export...`.
- O agente do CloudWatch pode estar com controle de utilização ao enviar métricas ou extensões para o Application Signals. Verifique se há mensagens que indicam um controle de utilização nos logs do agente do CloudWatch.

As métricas de dependência têm valores desconhecidos

Se você visualizar UnknownOperation, UnknownRemoteService ou UnknownRemoteOperation para um nome de dependência ou de operação nos painéis do Application Signals, verifique se a ocorrência de pontos de dados para o serviço remoto desconhecido e para a operação remota desconhecida estão coincidindo com as implantações. Este é um problema conhecido no Application Signals, que está planejado para ser corrigido em uma versão futura.

Tratamento de um ConfigurationConflict durante o gerenciamento do complemento Amazon CloudWatch Observability do EKS

Ao instalar ou atualizar o complemento Amazon CloudWatch Observability do EKS, se você perceber uma falha causada por um `Health Issue` do tipo `ConfigurationConflict` com uma descrição que começa com `Conflicts found when trying to apply. Will not continue due to resolve conflicts mode`, é provável que você já tenha o agente do CloudWatch e os componentes associados, como o ServiceAccount, o ClusterRole e o ClusterRoleBinding instalados

no cluster. Quando o complemento tentar instalar o agente do CloudWatch e os componentes associados, se ele detectar quaisquer alterações no conteúdo, por padrão, apresentará falhas na instalação ou na atualização para evitar a substituição do estado dos recursos no cluster.

Se você estiver tentando realizar a integração do complemento Amazon CloudWatch Observability do EKS e obter essa falha, recomendamos excluir uma configuração existente do agente do CloudWatch instalada anteriormente no cluster e, em seguida, instalar o complemento do EKS. Certifique-se de fazer backup de quaisquer personalizações que você possa ter executado na configuração original do agente do CloudWatch, como uma configuração do agente personalizada, e fornecê-las ao complemento Amazon CloudWatch Observability do EKS na próxima instalação ou atualização. Se você realizou a instalação do agente do CloudWatch para a integração com o Container Insights, consulte Exclusão do agente do CloudWatch e do Fluent Bit para o Container Insights para obter mais informações.

Como alternativa, o complemento oferece suporte a uma opção de configuração de resolução de conflitos que tem a funcionalidade de especificar `OVERWRITE`. É possível usar essa opção para prosseguir com a instalação ou a atualização do complemento ao substituir os conflitos no cluster. Se você estiver usando o console do Amazon EKS, encontrará o Método de resolução de conflitos ao escolher as Definições de configuração opcionais na criação ou na atualização do complemento. Caso esteja usando a AWS CLI, você poderá fornecer o comando `--resolve-conflicts OVERWRITE` para criar ou atualizar o complemento.

Configuração do Application Signals

> ⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Esta seção contém informações sobre como configurar o CloudWatch Application Signals.

Taxa de amostragem de rastreamentos

Por padrão, quando você habilita a amostragem centralizada do X-Ray para o Application Signals, a habilitação ocorre usando as configurações de taxa de amostragem padrão de `reservoir=1/s` e `fixed_rate=5%`. As variáveis de ambiente para o agente do SDK para AWS Distro para OpenTelemetry (ADOT) são definidas conforme apresentado a seguir.

Variável de ambiente	Valor	Observação
`OTEL_TRACES_SAMPLER`	`xray`	
`OTEL_TRACES_SAMPLER_ARG`	`endpoint=http://cloudwatch-agent.amazon-cloudwatch:2000`	Endpoint do agente do CloudWatch

Para obter informações sobre como alterar a configuração de amostragem, consulte o seguinte:

- Para alterar a amostragem do X-Ray, consulte Customizing sampling rules.
- Para alterar a amostragem do ADOT, consulte Configuring the OpenTelemetry Collector for X-Ray remote sampling.

Caso deseje desabilitar a amostragem centralizada do X-Ray e usar a amostragem local, defina os valores a seguir para o agente em Java do SDK para ADOT, conforme apresentado abaixo. O exemplo apresentado a seguir define a taxa de amostragem em 5%.

Variável de ambiente	Valor
`OTEL_TRACES_SAMPLER`	`parentbased_traceidratio`
`OTEL_TRACES_SAMPLER_ARG`	`0.05`

Para obter informações sobre configurações de amostragem mais avançadas, consulte OTEL_TRACES_SAMPLER.

Gerenciamento de operações de alta cardinalidade

O Application Signals inclui configurações no agente do CloudWatch que podem ser usadas para gerenciar a cardinalidade das operações e administrar a exportação de métricas para otimizar custos. Por padrão, a função de limitação de métricas torna-se ativa quando o número de operações distintas para um serviço ao longo do tempo excede o limite padrão de 500. É possível ajustar o comportamento ao adaptar as definições de configuração.

Como determinar se a limitação de métricas está ativada

É possível usar os métodos apresentados a seguir para descobrir se a limitação de métricas padrão está ocorrendo. Se a limitação estiver ativada, considere otimizar o controle de cardinalidade ao seguir as etapas da próxima seção.

- No console do CloudWatch, escolha Application Signals e, em seguida, selecione Serviços. Se você vir uma dimensão Operation chamada AllOtherOperations ou uma dimensão RemoteOperation chamada AllOtherRemoteOperations, a limitação de métricas estará ocorrendo.
- Se alguma métrica coletada pelo Application Signals tiver o valor `AllOtherOperations` para a dimensão `Operation`, a limitação de métricas estará ocorrendo.
- Se alguma métrica coletada pelo Application Signals tiver o valor `AllOtherRemoteOperations` para a dimensão `RemoteOperation`, a limitação de métricas estará ocorrendo.

Como otimizar o controle de cardinalidade

Para otimizar o controle de cardinalidade, é possível fazer o seguinte:

- Criar regras personalizadas para agregar operações.
- Configurar a política de limitação de métricas.

Criar regras personalizadas para agregar operações

Às vezes, as operações de alta cardinalidade podem ser causadas por valores exclusivos inadequados que foram extraídos do contexto. Por exemplo, o envio de solicitações HTTP/S que incluem IDs de usuário ou IDs de sessão no caminho pode resultar em centenas de operações diferentes. Para resolver esses problemas, recomendamos configurar o agente do CloudWatch com regras de personalização para gravar novamente essas operações.

Nos casos em que há um aumento na geração de inúmeras métricas diferentes por meio de chamadas `RemoteOperation` individuais, como `PUT /api/customer/owners/123`, `PUT /api/customer/owners/456` e solicitações semelhantes, recomendamos consolidar essas operações em uma única `RemoteOperation`. Uma abordagem possível é padronizar todas as chamadas `RemoteOperation` que começam com `PUT /api/customer/owners/` para um formato uniforme, especificamente `PUT /api/customer/owners/{ownerId}`. Isso é ilustrado no exemplo a seguir. Para obter informações sobre outras regras de personalização, consulte Habilitar o CloudWatch Application Signals.

```
{
   "logs":{
      "metrics_collected":{
         "app_signals":{
            "rules":[
               {
                  "selectors":[
                     {
                        "dimension":"RemoteOperation",
                        "match":"PUT /api/customer/owners/*"
                     }
                  ],
                  "replacements":[
                     {
                        "target_dimension":"RemoteOperation",
                        "value":"PUT /api/customer/owners/{ownerId}"
                     }
                  ],
                  "action":"replace"
               }
            ]
         }
      }
   }
}
```

Em outros casos, as métricas de alta cardinalidade podem ter sido agregadas a `AllOtherRemoteOperations` e pode não estar claro quais métricas específicas estão incluídas. O agente do CloudWatch é capaz de registrar em log as operações descartadas. Para identificar as operações descartadas, use a configuração apresentada no exemplo a seguir para ativar o registro em log até que o problema ocorra novamente. Em seguida, inspecione os logs do agente do CloudWatch (acessíveis por `stdout` do contêiner ou por arquivos de log do EC2) e pesquise a palavra-chave `drop metric data`.

```
{
  "agent": {
    "config": {
      "agent": {
        "debug": true
      },
      "traces": {
        "traces_collected": {
```

```
          "app_signals": {
          }
        }
      },
      "logs": {
        "metrics_collected": {
          "app_signals": {
            "limiter": {
              "log_dropped_metrics": true
            }
          }
        }
      }
    }
  }
}
```

Criar a política de limitação de métricas

Se a configuração de limitação de métricas padrão não abordar a cardinalidade para o seu serviço, será possível personalizar a configuração do limitador de métricas. Para fazê-lo, adicione uma seção `limiter` na seção `logs/metrics_collected/app_signals` no arquivo de configuração do agente do CloudWatch.

O exemplo apresentado a seguir reduz o limite de limitação de métricas de 500 métricas distintas para 100.

```
{
  "logs": {
    "metrics_collected": {
      "app_signals": {
        "limiter": {
          "drop_threshold": 100
        }
      }
    }
  }
}
```

Objetivos de nível de serviço (SLOs)

⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Você pode usar o Application Signals para criar objetivos de nível de serviço para os serviços das suas operações de negócios críticas. Ao criar SLOs nesses serviços, você poderá rastreá-los no painel do SLO, obtendo uma visualização à primeira vista das suas operações mais importantes.

Além de criar uma visualização rápida que seus operadores podem usar para ver o status atual das operações críticas, você pode usar os SLOs para monitorar a performance de longo prazo dos seus serviços para garantir que eles estejam atendendo às suas expectativas. Se você tem acordos de serviço com clientes, os SLOs são uma excelente ferramenta para garantir que eles sejam cumpridos.

A avaliação da integridade dos serviços com SLOs começa com a definição de objetivos claros e mensuráveis com base nas principais métricas de performance: indicadores de nível de serviço (SLIs). Um SLO rastreia a performance do SLI em comparação com o limite e com a meta que você definiu e informa o ponto em que a performance da aplicação se encontra com relação ao limite.

O Application Signals ajuda você a definir SLOs nas principais métricas de performance. O Application Signals coleta automaticamente métricas de `Latency` e `Availability` para cada serviço e operação que ele descobre, e essas métricas muitas vezes são ideais para uso como SLIs. Com o assistente de criação de SLO, você pode usar essas métricas para seus SLOs. Em seguida, você pode rastrear o status de todos os seus SLOs com os painéis do Application Signals.

Você pode definir SLOs em operações específicas que seu serviço chama ou usa. Você pode usar qualquer métrica ou expressão métrica do CloudWatch como SLI, além de usar as métricas `Latency` e `Availability`.

Criar SLOs é muito importante para obter o máximo benefício do CloudWatch Application Signals. Depois de criar SLOs, você pode visualizar o status deles no console do Application Signals para ver rapidamente quais desses serviços e operações essenciais estão apresentando boa performance e quais não estão íntegros. Ter SLOs para rastrear oferece os seguintes benefícios principais:

- É mais fácil para seus operadores de serviços ver a integridade operacional atual dos serviços essenciais medida em relação ao SLI. Em seguida, eles podem rapidamente fazer uma triagem e identificar serviços e operações não íntegros.
- Você pode rastrear a performance do seu serviço em relação a metas de negócios mensuráveis por longos períodos.

Ao escolher no que definir SLOs, você está priorizando o que é importante para você. Os painéis do Application Signals apresentam automaticamente informações sobre o que você priorizou.

Ao criar um SLO, você também pode optar por criar alarmes do CloudWatch ao mesmo tempo para monitorar os SLOs. Você pode definir alarmes que monitorem violações do limite e também os níveis de aviso. Esses alarmes podem avisar automaticamente se as métricas de SLO estão ultrapassando o limite que você definiu ou se estão se aproximando de um limite de aviso. Por exemplo, um SLO próximo do limite de aviso pode informar que sua equipe talvez precise diminuir a rotatividade da aplicação para garantir que as metas de performance de longo prazo sejam cumpridas.

Tópicos

Conceitos de SLO

Um SLO inclui os seguintes componentes:

- Um indicador de nível de serviço (SLI), que é uma métrica essencial de performance que você especifica. Ele representa o nível de performance desejado para sua aplicação. O Application Signals coleta automaticamente as métricas essenciais de `Latency` e `Availability` para os serviços e operações que ele descobre, e essas métricas muitas vezes são ideais para se definir SLOs.

  Você escolhe o limite a ser usado para o SLI. Por exemplo, 200 ms para latência.

- Uma meta ou meta de realização, que é a porcentagem de tempo em que é esperado que o SLI atinja o limite em cada intervalo de tempo. Os intervalos de tempo podem ser de algumas horas ou até de um ano.

  Os intervalos podem ser intervalos do calendário ou intervalos contínuos.

  - Os intervalos do calendário estão alinhados com o calendário, como um SLO que é rastreado por mês. O CloudWatch ajusta automaticamente a integridade, o orçamento e os números de realizações com base no número de dias em um mês. Os intervalos do calendário são mais adequados para metas de negócios que são avaliadas de acordo com o calendário.
  - Os intervalos contínuos são calculados em uma base contínua. Os intervalos contínuos são mais adequados para rastrear a experiência recente do usuário na aplicação.
- O período é um intervalo de tempo mais curto, e muitos períodos formam um intervalo. A performance da aplicação é comparada ao SLI durante cada período dentro do intervalo. Para cada período, é determinado se a aplicação atingiu ou não a performance necessária.

Por exemplo, uma meta de 99% com um intervalo do calendário de um dia e um período de um minuto significa que a aplicação deve cumprir ou atingir o limite de sucesso durante 99% dos períodos de um minuto durante o dia. Se isso acontecer, o SLO terá sido alcançado nesse dia. O dia seguinte é um novo intervalo de avaliação, e a aplicação deve cumprir ou atingir o limite de sucesso durante 99% dos períodos de um minuto durante o segundo dia para alcançar o SLO desse segundo dia.

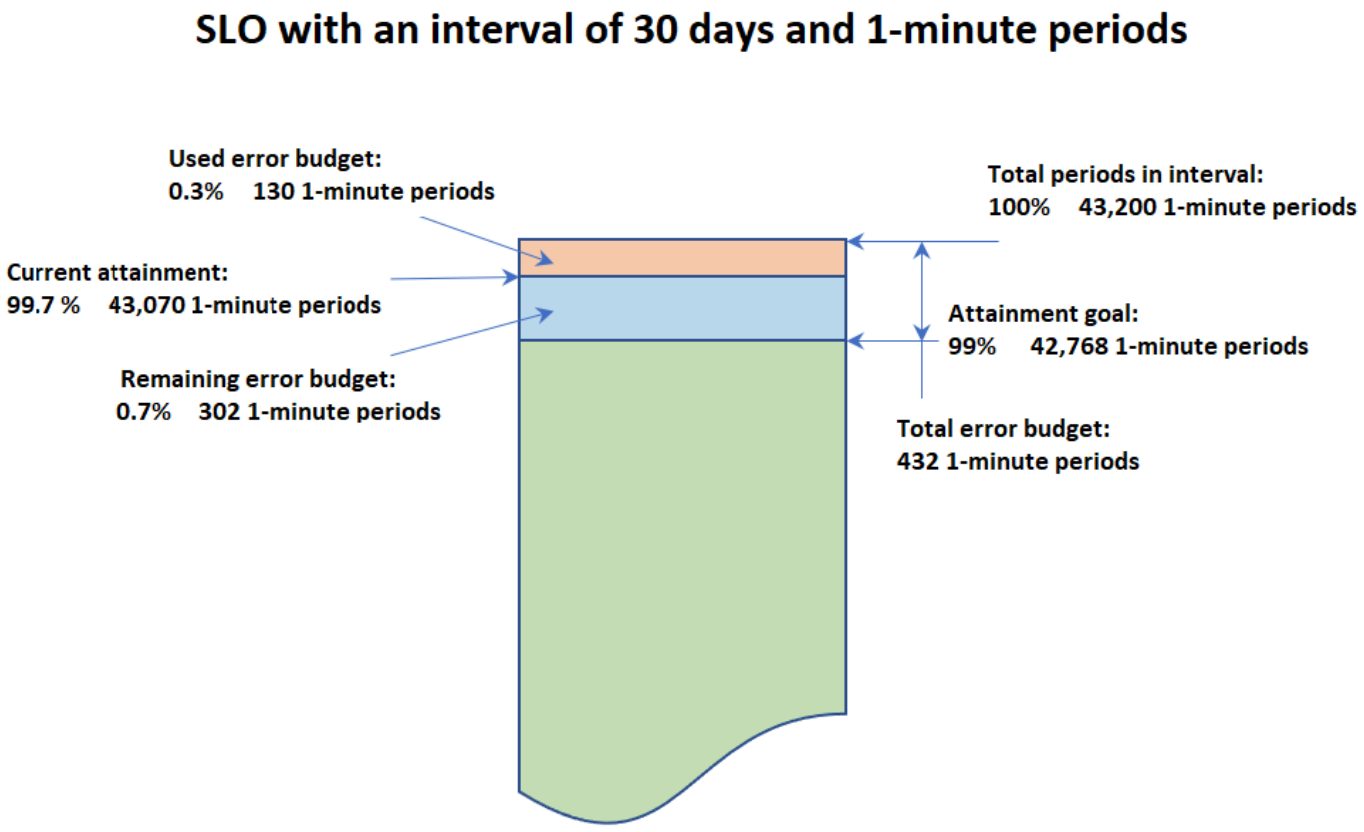
Um SLI pode ser baseado em uma das novas métricas de aplicação padrão coletadas pelo Application Signals. Como alternativa, pode ser qualquer métrica ou expressão métrica do CloudWatch. As métricas de aplicação padrão que você pode usar para um SLI são `Latency` e `Availability`. `Availability` representa o número de respostas bem-sucedidas dividido pelo total de solicitações. Essa métrica é calculada como (1 - Taxa de falha)*100, em que as respostas à falha são erros `5xx`. Respostas de sucesso são respostas sem erros `5XX`. Respostas `4XX` são tratadas como com êxito.

> **Note**
>
> Atualmente, somente cálculos baseados em períodos são compatíveis. A compatibilidade com cálculos baseados em volume ou em solicitações está planejada para versões futuras.

Calcular o orçamento de erros e a realização

Ao visualizar informações sobre um SLO, você vê o status atual de integridade e o orçamento de erros dele. O orçamento de erros é a quantidade de tempo dentro do intervalo que pode violar o limite, mas ainda permitir que o SLO seja alcançado. O orçamento total de erros é o tempo total de violação que pode ser tolerado durante todo o intervalo. O orçamento restante de erros é o tempo restante de violação que pode ser tolerado durante o intervalo atual. Isso ocorre depois que a quantidade de tempo de violação que já ocorreu foi subtraída do orçamento total de erros.

A figura a seguir ilustra os conceitos de realização e orçamento de erro para uma meta com um intervalo de 30 dias, períodos de um minuto e uma meta de realização de 99%. Trinta dias incluem 43.200 períodos de um minuto e 99% de 43.200 são 42.768. Portanto, 42.768 minutos durante o mês devem estar íntegros para que o SLO seja alcançado. Até agora, no intervalo atual, 130 dos períodos de um minuto não estão íntegros.



Determinar o sucesso em cada período

Em cada período, os dados do SLI são agregados em um único ponto de dados com base na estatística usada para o SLI. Esse ponto de dados representa toda a duração do período. Esse

único ponto de dados é comparado ao limite do SLI para determinar se o período está íntegro. Ver períodos não íntegros durante o intervalo de tempo atual no painel pode alertar seus operadores de serviços de que o serviço precisa ser submetido a uma triagem.

Se o período for determinado como não íntegro, toda a duração do período será contabilizada como falha no orçamento de erros. O rastreamento do orçamento de erros permite que você saiba se o serviço está atingindo a performance desejada por um longo período.

Criar um SLO

Recomendamos que você defina SLOs de latência e disponibilidade nas aplicações essenciais. Essas métricas coletadas pelo Application Signals se alinham às metas de negócios comuns.

Você também pode definir SLOs em qualquer métrica do CloudWatch ou em qualquer expressão de matemática de métricas que resulte em uma única série temporal.

Na primeira vez que você cria um SLO em sua conta, o CloudWatch cria o perfil vinculado ao serviço AWSServiceRoleForCloudWatchApplicationSignals em sua conta de forma automática, se ele ainda não existir. Esse perfil vinculado ao serviço permite que o CloudWatch colete dados do CloudWatch Logs, dados de rastreamento do X-Ray, dados de métricas do CloudWatch e dados de marcação de aplicações em sua conta. Para obter mais informações sobre os perfis vinculados ao serviço do CloudWatch, consulte Usar funções vinculadas ao serviço para o CloudWatch.

Como criar um SLO

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Objetivos de nível de serviço (SLOs).
3. Escolha Criar SLO.
4. Insira um nome para o SLO. Incluir o nome de um serviço ou operação, junto com palavras-chave apropriadas, como latência ou disponibilidade, ajudará você a identificar rapidamente o que o status do SLO indicará durante a triagem.
5. Em Definir indicador de nível de serviço (SLI), execute uma das seguintes ações:
   - Definir o SLO em uma das métricas padrão da aplicação, `Latency` ou `Availability`:
     a. Escolha Operação de serviço.
     b. Selecione o serviço que esse SLO monitorará.
     c. Selecione a operação que esse SLO monitorará.

Os menus suspensos Selecionar serviço e Selecionar operação são preenchidos por serviços e operações que estiveram ativos nas últimas 24 horas.

d. Escolha Disponibilidade ou Latência e, em seguida, defina o limite.

- Para definir o SLO em qualquer métrica do CloudWatch ou em uma expressão matemática de métricas do CloudWatch:

  a. Escolha Métrica do CloudWatch.

  b. Escolha Selecionar métrica do CloudWatch.

  A tela Selecionar métrica é exibida. Use as guias Procurar ou Consultar para encontrar a métrica desejada ou crie uma expressão matemática de métricas.

  Depois de selecionar a métrica desejada, escolha a guia Métricas representadas graficamente e selecione a Estatística e o Período a serem usados para o SLO. Depois, escolha Select metric (Selecionar métrica).

  Para obter mais informações sobre essas telas, consulte Criar um gráfico de uma métrica e Adicionar uma expressão matemática a um gráfico do CloudWatch.

  c. Em Definir condição, selecione um operador de comparação e um limite para o SLO usar como indicador de sucesso.

6. Caso tenha selecionado Operação de serviço na etapa 5, você poderá, opcionalmente, escolher Configurações adicionais e, em seguida, ajustar a duração do período para esse SLO.
7. Defina o intervalo e a meta de realização para o SLO. Para obter mais informações sobre intervalos, metas de realização e como eles funcionam juntos, consulte Conceitos de SLO.
8. (Opcional) Defina um ou mais alarmes do CloudWatch ou um limite de aviso para o SLO.

   a. Os alarmes do CloudWatch podem usar o Amazon SNS para avisar proativamente se uma aplicação não está íntegra com base na performance do SLI dela.

   Para criar um alarme, marque uma das caixas de seleção de alarme e insira ou crie o tópico do Amazon SNS a ser usado nas notificações quando o alarme entrar no estado `ALARM`. Para obter mais informações sobre alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch. A criação de alarmes gera cobranças. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

b. Se você definir um limite de aviso, ele aparecerá nas telas do Application Signals para ajudar você a identificar SLOs que correm o risco de não serem alcançados, mesmo que estejam íntegros no momento.

   Para definir um limite de aviso, insira o valor do limite em Limite de aviso. Quando o orçamento de erros do SLO é mais baixo do que o limite de aviso, o SLO é marcado com Aviso em várias telas do Application Signals. Os limites de aviso também aparecem nos gráficos de orçamento de erros. Você também pode criar um Alarme de aviso de SLO com base no limite de aviso.

9. Para adicionar tags a esse SLO, escolha a guia Tags e, em seguida, escolha Adicionar nova tag. As tags podem ajudar você a gerenciar, identificar, organizar, pesquisar e filtrar recursos. Para obter mais informações sobre marcação, consulte Marcação dos recursos da AWS.

> **Note**
>
> Se a aplicação à qual esse SLO está relacionado estiver registrada no AWS Service Catalog AppRegistry, você poderá usar a tag `awsApplication` para associar o SLO a essa aplicação no AppRegistry. Para obter mais informações, consulte What is AppRegistry?

10. Escolha Criar SLO. Se você também optar por criar um ou mais alarmes, o nome do botão será alterado para refletir isso.

Visualizar e fazer a triagem do status do SLO

Você pode ver rapidamente a integridade dos SLOs usando os Objetivos de nível de serviço ou as opções de Serviços no console do CloudWatch. A exibição de Serviços fornece uma visualização à primeira vista da proporção de serviços não íntegros, calculada com base nos SLOs que você definiu. Para obter mais informações sobre como usar a opção Serviços, consulte Monitorar a integridade operacional das suas aplicações com o Application Signals.

A visualização de Objetivos de nível de serviço fornece uma visão macro da sua organização. Você pode ver os SLOs alcançados e não alcançados como um todo. Isso dá a você uma visão de quantos serviços e operações têm a performance de acordo com as expectativas por longos períodos, de acordo com os SLIs que você escolheu.

Como visualizar todos os SLOs usando a visualização de Objetivos de nível de serviço

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Objetivos de nível de serviço (SLOs).

   A lista Objetivos de nível de serviço (SLO) é exibida.

   Você pode ver rapidamente o status atual dos seus SLOs na coluna Status do SLI. Para classificar os SLOs de forma que todos os não íntegros estejam no topo da lista, escolha a coluna Status do SLI até que os SLOs não íntegros estejam todos no topo.

   A tabela de SLO tem as colunas padrão a seguir. Você pode ajustar quais colunas são exibidas ao escolher o ícone de engrenagem acima da lista. Para obter mais informações sobre metas, SLI, realização e intervalos, consulte Conceitos de SLO.

   - O nome do SLO.
   - A coluna Meta exibe a porcentagem de períodos durante cada intervalo que devem atingir com êxito o limite de SLI para que a meta de SLO seja cumprida. Ela também exibe a duração do intervalo para o SLO.
   - O Status do SLI mostra se o estado operacional atual da aplicação está íntegro ou não. Se algum período durante o intervalo de tempo atualmente selecionado não estiver íntegro para o SLO, o Status do SLI exibirá Não íntegro.
   - A Realização final é o nível de realização atingido no final do intervalo de tempo selecionado. Classifique por essa coluna para ver os SLOs que correm maior risco de não serem alcançados.
   - O Delta de realização é a diferença no nível de realização entre o início e o final do intervalo de tempo selecionado. Um delta negativo significa que a métrica está tendendo para uma direção descendente. Classifique por essa coluna para ver as últimas tendências dos SLOs.
   - O Orçamento final de erros (%) é a porcentagem do tempo total no intervalo que pode ter períodos não íntegros e, ainda assim, ter o SLO alcançado com êxito. Se você defini-lo como 5% e o SLI não estiver íntegro em 5% ou menos dos períodos restantes no intervalo, o SLO ainda será alcançado com êxito.
   - O Delta do orçamento de erros é a diferença no orçamento de erros entre o início e o final do intervalo de tempo selecionado. Um delta negativo significa que a métrica está tendendo para uma direção de falha.
   - O Orçamento final de erros (tempo) é o tempo real no intervalo que pode não estar íntegro e ainda assim fazer com que o SLO seja alcançado com êxito. Por exemplo, se ele for 14

minutos, então se o SLI não estiver íntegro por menos de 14 minutos durante o intervalo restante, o SLO ainda será alcançado com êxito.

- As colunas Serviço, Operação e Tipo exibem informações sobre para qual serviço e operação esse SLO está definido.

3. Para visualizar os gráficos da realização e do orçamento de erros de um SLO, escolha o botão de opção ao lado do nome do SLO.

   Os gráficos na parte superior da página exibem a Realização do SLO e o status do Orçamento de erros. Um gráfico sobre a métrica do SLI associada a esse SLO também é exibido.

4. Para fazer uma triagem adicional de um SLO que não está atingindo a meta, escolha o nome do serviço ou o nome da operação associado a esse SLO. Você será direcionado para a página de detalhes, na qual poderá fazer uma triagem adicional. Para ter mais informações, consulte Visualize as atividades de serviço e a integridade operacional em detalhes com a página de detalhes do serviço.

5. Para alterar o intervalo de tempo dos gráficos e tabelas da página, escolha um novo intervalo de tempo próximo à parte superior da tela.

Editar um SLO existente

Siga estas etapas para editar um SLO existente. Ao editar um SLO, você pode alterar somente o limite, o intervalo, a meta de realização e as tags. Para alterar outros aspectos, como serviço, operação ou métrica, crie um novo SLO em vez de editar um existente.

Alterar parte da configuração principal de um SLO, como período ou limite, invalida todos os pontos de dados e avaliações anteriores sobre realização e integridade. Isso efetivamente exclui e recria o SLO.

> **Note**
>
> Se você editar um SLO, os alarmes associados a ele não serão atualizados automaticamente. Talvez seja necessário atualizar os alarmes para mantê-los sincronizados com o SLO.

Como editar um SLO existente

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

2. No painel de navegação, escolha Objetivos de nível de serviço (SLOs).
3. Escolha o botão de opção ao lado do SLO que você deseja editar e escolha Ações, Editar SLO.
4. Faça suas alterações e, em seguida, escolha Salvar alterações.

Excluir um SLO

Siga estas etapas para excluir um SLO atual.

> **Note**
>
> Se você excluir um SLO, os alarmes associados a ele não serão excluídos automaticamente. Você precisará excluí-los por conta própria. Para ter mais informações, consulte Gerenciar alarmes.

Como excluir um SLO

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Objetivos de nível de serviço (SLOs).
3. Escolha o botão de opção ao lado do SLO que você deseja editar e escolha Ações, Excluir SLO.
4. Selecione a opção Confirmar.

Monitorar a integridade operacional das suas aplicações com o Application Signals

> **Warning**
>
> O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

Use o Application Signals no console do CloudWatch para monitorar e solucionar problemas da integridade operacional das suas aplicações:

- Monitore seus serviços de aplicações: como parte do monitoramento operacional diário, use a página Serviços para visualizar um resumo de todos os seus serviços. Veja os serviços com a

maior taxa de falhas ou latência e veja quais serviços têm indicadores de nível de serviço (SLIs) não íntegros. Selecione um serviço para abrir a página Detalhes do serviço e veja as métricas precisas, as operações de serviço, os canários do Synthetics e as solicitações de clientes. Isso pode ajudar a solucionar e identificar a causa-raiz dos problemas operacionais.

- Inspecione a topologia da aplicação: use o Mapa de serviços para entender e monitorar a topologia da sua aplicação ao longo do tempo, incluindo os relacionamentos entre clientes, os canários do Synthetics, os serviços e as dependências. Veja instantaneamente a integridade do SLI e as principais métricas, como volume de chamadas, taxa de falhas e latência. Faça uma busca profunda para ver informações mais precisas na página Detalhes do serviço.

Explore um exemplo de cenário que demonstre como essas páginas podem ser usadas para solucionar rapidamente um problema operacional de integridade do serviço, desde a detecção inicial até a identificação da causa-raiz.

Como o Application Signals permite o monitoramento da integridade operacional

Depois de habilitar a aplicação para o Application Signals, seus serviços de aplicações, as APIs e suas dependências são automaticamente descobertos e exibidos nas páginas Serviços, Detalhes do serviço e Mapa de serviços. O Application Signals coleta informações de várias fontes para permitir a descoberta de serviços e o monitoramento da integridade operacional:

- AWS Distro para OpenTelemetry (ADOT): como parte da habilitação do Application Signals, uma biblioteca de instrumentação automática OpenTelemetry Java é configurada para emitir métricas e rastreamentos que são coletados pelo agente do CloudWatch. As métricas e os rastreamentos são usados para permitir a descoberta de serviços, operações, dependências e outras informações do serviço.
- Objetivos de nível de serviço (SLOs): depois de criar objetivos de nível para seus serviços, as páginas Serviços, Detalhes do serviço e Mapa de serviços exibem a integridade do indicador de nível de serviço (SLI). Os SLIs podem monitorar a latência, a disponibilidade e outras métricas operacionais.
- Canários do CloudWatch Synthetics: quando você configura o rastreamento do X-Ray nos canários, as chamadas dos scripts dos canários para os serviços são associadas ao seu serviço e exibidas na página Detalhes do serviço.
- Monitoramento real de usuários (RUM) do CloudWatch: quando o rastreamento do X-Ray é habilitado no seu cliente Web do CloudWatch RUM, as solicitações para seus serviços são automaticamente associadas e exibidas na página de detalhes do serviço.

- AWS Service Catalog AppRegistry: o Application Signals descobre automaticamente recursos da AWS na sua conta e permite que você os agrupe em aplicações lógicas criadas no AppRegistry. O nome da aplicação exibido na página Serviços é baseado no recurso de computação subjacente no qual seus serviços estão sendo executados.

> **Note**
>
> O Application Signals exibe seus serviços e operações com base em métricas e rastreamentos emitidos no filtro de tempo atual que você escolheu. (Por padrão, isso corresponde às últimas três horas.) Se não houver qualquer atividade no filtro de tempo atual para um serviço, uma operação, uma dependência, um canário do Synthetics ou uma página do cliente, nada será exibido.
> Atualmente, até mil serviços podem ser exibidos. A descoberta dos seus serviços e da topologia do serviço pode ser atrasada em até dez minutos. A avaliação da integridade do SLI pode ser atrasada em até 15 minutos.

Visualizar a atividade geral do serviço e a integridade operacional com a página Serviços

> **Warning**
>
> O Application Signals está na versão de pré-visualização para Amazon CloudWatch e está sujeito a alterações.

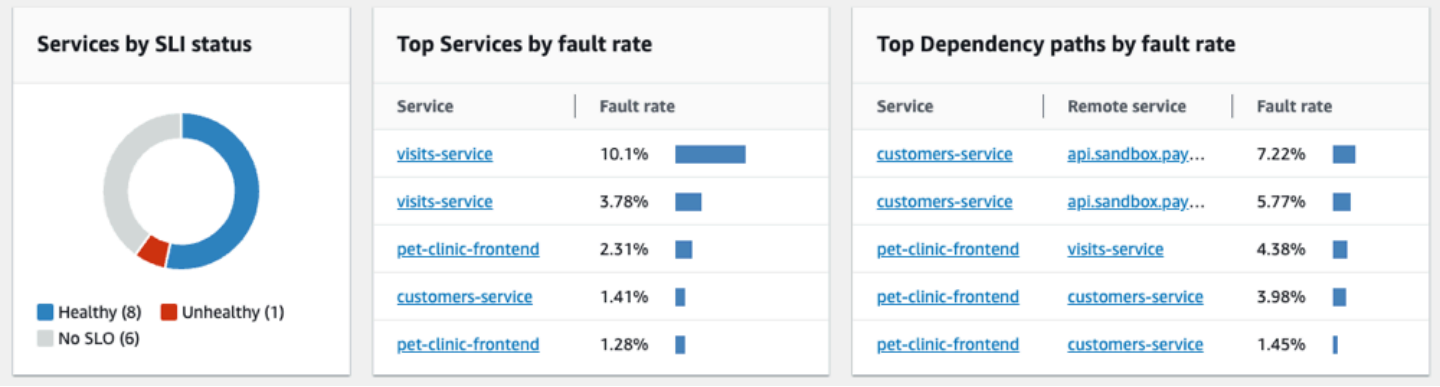
Use a página Serviços para ver uma lista dos serviços que estão habilitados para o Application Signals. Você também pode visualizar métricas operacionais e ver rapidamente quais serviços têm indicadores de nível de serviço (SLIs) não íntegros. Faça uma busca detalhada para procurar anomalias de performance quando identificar a causa-raiz dos problemas operacionais. Para visualizar essa página, abra o console do CloudWatch e escolha Serviços na seção Application Signals no painel de navegação esquerdo.

Explorar métricas de integridade operacional para seus serviços

A parte superior da página Serviços inclui um gráfico geral da integridade operacional do serviço e várias tabelas exibindo os principais serviços e dependências de serviços por taxa de falhas. O gráfico Serviços à esquerda mostra um detalhamento do número de serviços que têm indicadores de

nível de serviço (SLIs) íntegros ou não íntegros durante o filtro de tempo atual em nível de página. Os SLIs podem monitorar a latência, a disponibilidade e outras métricas operacionais.

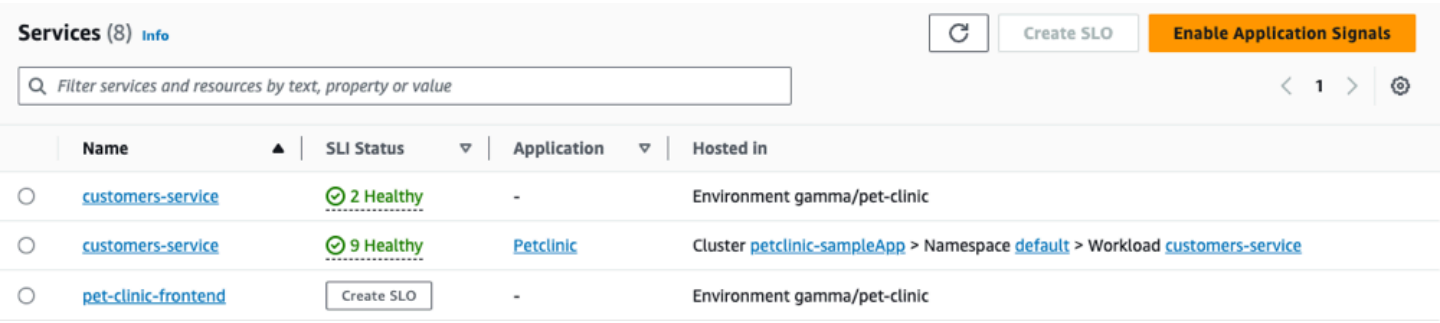
As duas tabelas ao lado do gráfico exibem uma lista dos principais serviços por taxa de falhas. Escolha qualquer nome de serviço em qualquer tabela para abrir uma página de detalhes do serviço e ver detalhes da operação do serviço. Escolha um caminho de dependência para abrir a página de detalhes e ver os detalhes da dependência do serviço. Ambas as tabelas exibem informações das últimas três horas, mesmo que um filtro de período mais longo seja escolhido no canto superior direito da página.



Monitorar a integridade operacional com a tabela Serviços

A tabela Serviços exibe uma lista dos serviços que foram habilitados para o Application Signals. Escolha Habilitar o Application Signals para abrir uma página de configuração e começar a configurar os serviços. Para obter mais informações, consulte Enable Application Signals.

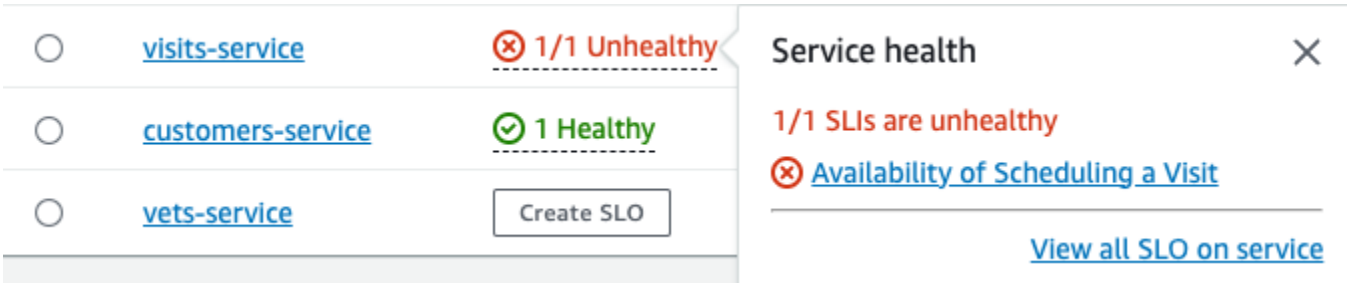
Filtre a tabela Serviços para facilitar a descoberta do que você está procurando ao escolher uma ou mais propriedades na caixa de texto do filtro. Ao escolher cada propriedade, você é guiado pelos critérios do filtro. Você verá o filtro completo abaixo da caixa de texto do filtro. Escolha Limpar filtros a qualquer momento para remover o filtro da tabela.



Escolha o nome de qualquer serviço na tabela para visualizar uma página de detalhes do serviço contendo métricas de nível de serviço, operações e detalhes adicionais. Se você tiver associado

o recurso de computação subjacente do serviço a uma aplicação no AppRegistry ou ao cartão do Applications na página inicial do AWS Management Console, escolha o nome da aplicação para exibir os detalhes da aplicação na página myApplications do console. Para serviços hospedados no Amazon EKS, escolha qualquer link na coluna Hospedado em para visualizar Cluster, Namespace ou Workload no CloudWatch Container Insights. Para serviços em execução no Amazon ECS ou no Amazon EC2, o valor de Ambiente é mostrado.

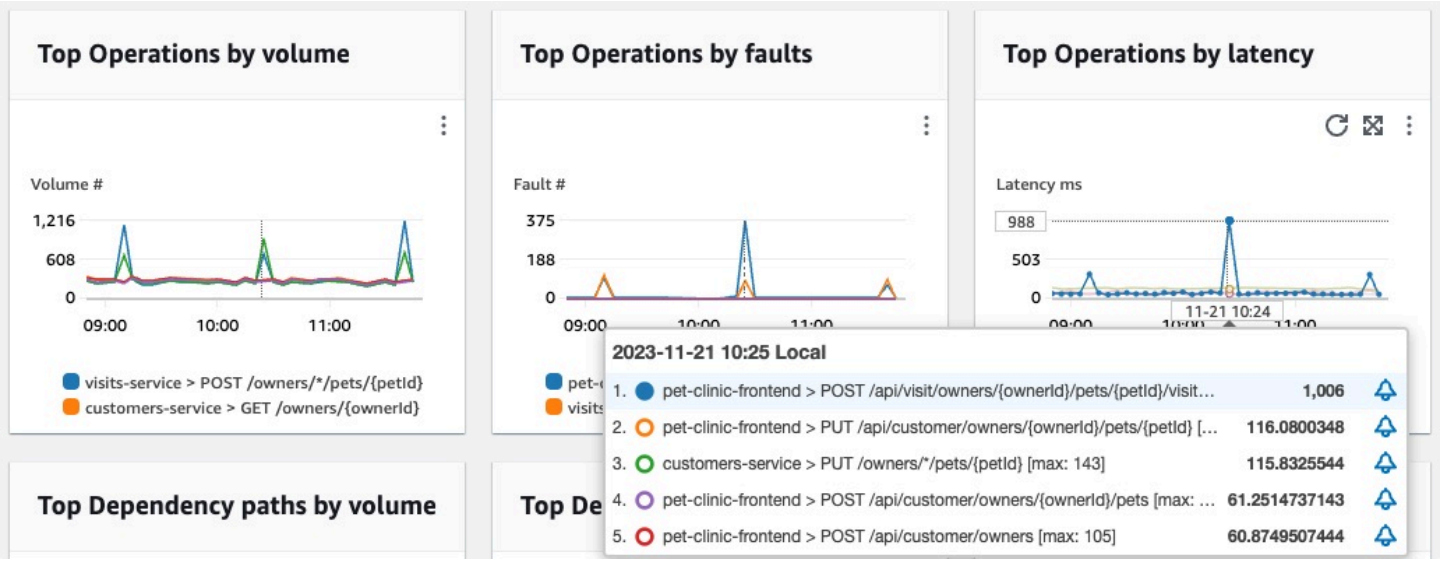
O status do indicador de nível de serviço (SLI) é exibido para cada serviço na tabela. Escolha o status do SLI de um serviço para exibir um pop-up contendo um link para quaisquer SLIs não íntegros e um link para ver todos os SLOs do serviço.



Se nenhum SLO tiver sido criado para um serviço, escolha o botão Criar SLO na coluna Status do SLI. Para criar SLOs adicionais para qualquer serviço, selecione o botão de opção ao lado do nome do serviço e escolha Criar SLO no canto superior direito da tabela. Ao criar SLOs, você pode ver rapidamente quais dos seus serviços e operações apresentam boa performance e quais não estão íntegros. Para obter mas informações, consulte service level objectives (SLOs).

Visualizar as principais métricas de operação e dependência

Abaixo da tabela Serviços, você pode visualizar as principais operações e dependências em todos os serviços por volume de chamadas, falhas e latência. Esse conjunto de gráficos fornece informações críticas sobre quais operações ou dependências podem não estar íntegras em todos os serviços. Escolha qualquer ponto em um gráfico para ver um pop-up contendo informações das séries mais detalhadas. Passe o cursor sobre as descrições das séries na parte inferior de um gráfico para ver um pop-up contendo métricas detalhadas de uma operação específica ou caminho de dependência. Selecione o botão do menu de contexto no canto superior direito de um gráfico para ver opções adicionais, incluindo a visualização de métricas ou páginas de logs do CloudWatch.



Visualize as atividades de serviço e a integridade operacional em detalhes com a página de detalhes do serviço

⚠ O Application Signals está na versão de pré-visualização para Amazon CloudWatch e está sujeito a alterações.

Ao realizar a instrumentação da aplicação, o Amazon CloudWatch Application Signals mapeia todos os serviços que a aplicação descobre. Use a página de detalhes do serviço para obter uma visão geral dos serviços, das operações, das dependências, dos canários e das solicitações de clientes para um único serviço. Para visualizar a página de detalhes do serviço, faça o seguinte:

- Abra o console do CloudWatch.
- Escolha Serviços, na seção Application Signals, no painel de navegação esquerdo.
- Escolha o nome de qualquer serviço nas tabelas de Serviços, de Principais serviços ou de dependências.

A página de detalhes do serviço está organizada nas seguintes guias:

- Visão geral: use esta guia para obter uma visão geral de um único serviço, incluindo o número de operações, as dependências, os canários do Synthetics e as páginas de clientes. A guia mostra as principais métricas de todo o seu serviço, as principais operações e as dependências. Essas

métricas incluem dados de séries temporais sobre latência, falhas e erros em todas as operações de serviço para esse serviço.

- Operações de serviço: use esta guia para obter uma lista das operações que seu serviço chama e os gráficos interativos com as principais métricas que medem a integridade de cada operação. É possível selecionar um ponto de dados em um gráfico para obter informações sobre rastreamentos, logs ou métricas associadas a esse ponto de dados.
- Dependências: use esta guia para obter uma lista das dependências que seu serviço chama e uma lista de métricas para essas dependências.
- Canários do Synthetics: use esta guia para obter uma lista de canários do Synthetics que simulam chamadas de usuários para o serviço e as principais métricas de desempenho para esses canários.
- Páginas de clientes: use esta guia para obter uma lista das páginas de clientes que chamam seu serviço e as métricas que medem a qualidade das interações dos clientes com a aplicação.

Visualizar a visão geral do serviço

Use a página de visão geral do serviço para visualizar um resumo de alto nível das métricas para todas as operações de serviço em um único local. Verifique o desempenho de todas as operações, dependências, páginas de clientes e canários do Synthetics que interagem com a aplicação. Use essas informações para ajudar na determinação do melhor local para concentrar os esforços com a finalidade de identificar problemas, solucionar erros e encontrar oportunidades para a otimização.

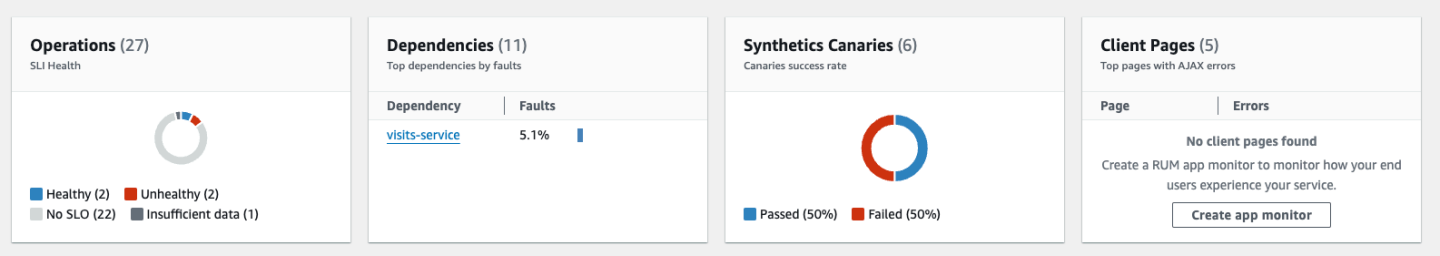
Escolha qualquer link em Detalhes do serviço para visualizar informações relacionadas a um serviço específico. Por exemplo, para serviços hospedados no Amazon EKS, a página de detalhes do serviço mostra informações relacionadas ao Cluster, ao Namespace e à Workload. Para serviços hospedados no Amazon ECS ou no Amazon EC2, a página de detalhes do serviço mostra o valor Ambiente.

Em Serviços, a guia Visão geral exibe um resumo do seguinte:

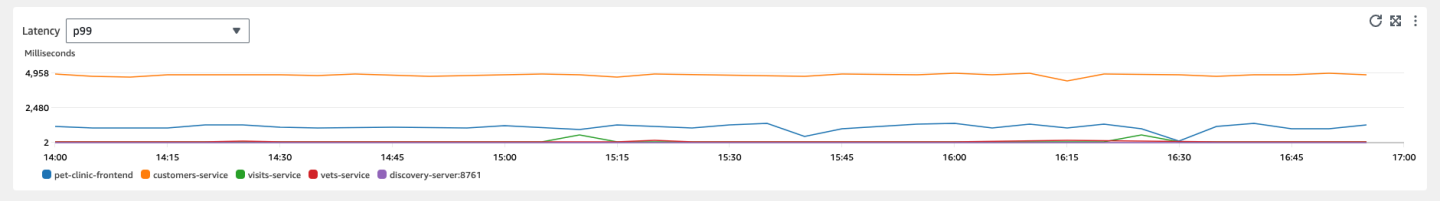
- Operações: use esta guia para obter a integridade das operações de serviço. O status da integridade é determinado por indicadores de nível de serviço (SLIs) que são definidos como parte de um objetivo de nível de serviço (SLO).
- Dependências: use esta guia para obter as principais dependências dos serviços chamados pela aplicação, listadas por taxa de falhas.
- Canários do Synthetics: use esta guia para obter o resultado de chamadas simuladas para os endpoints ou para as APIs associados ao serviço e o número de canários com falha.

- Páginas de clientes: use esta guia para obter as principais páginas chamadas por clientes que apresentam erros assíncronos de JavaScript e de XML (AJAX).

A ilustração a seguir mostra uma visão geral dos serviços:



A guia Visão geral também exibe um gráfico das dependências com maior latência entre todos os serviços. Use as métricas de latência p99, p90 e p50 para avaliar rapidamente quais dependências estão contribuindo para a latência total do serviço, da seguinte forma:

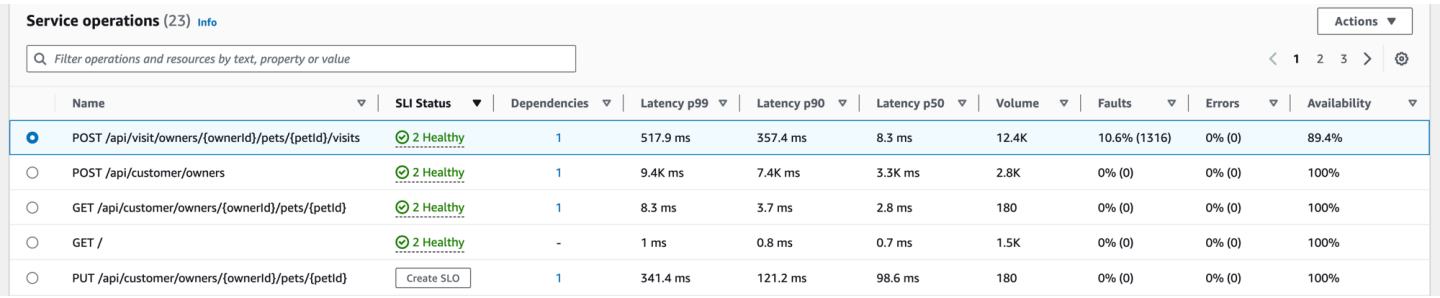


Por exemplo, o gráfico apresentado anteriormente mostra que 99% das solicitações realizadas à dependência de serviço de atendimento ao cliente foram concluídas em aproximadamente 4.950 milissegundos. As outras dependências demoraram menos tempo para serem concluídas.

Os gráficos que exibem as quatro principais operações de serviço por latência mostram o volume de solicitações, a disponibilidade, a taxa de falhas e a taxa de erros desses serviços, conforme mostrado na seguinte imagem:

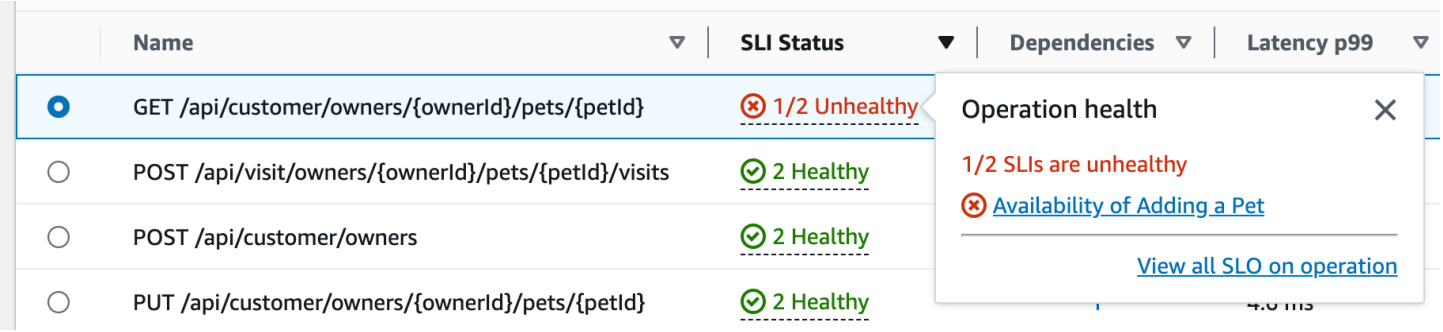
Visualizar as operações de serviço

Ao realizar a instrumentação da aplicação, o Application Signals descobre todas as operações de serviço que a aplicação chama. Use a guia Operações de serviço para visualizar uma tabela que contém as operações de serviço e um conjunto de métricas que medem o desempenho de uma operação selecionada. Essas métricas incluem o status do SLI, o número de dependências, a latência, o volume, as falhas, os erros e a disponibilidade, conforme mostrado na seguinte imagem:



Filtre a tabela para facilitar a localização de uma operação de serviço ao escolher uma ou mais propriedades na caixa de texto do filtro. Ao escolher cada propriedade, você será guiado pelos critérios do filtro e verá o filtro completo abaixo da caixa de texto do filtro. Escolha Limpar filtros a qualquer momento para remover o filtro da tabela.

Escolha o status do SLI para uma operação a fim de exibir um pop-up que contém um link para qualquer SLI não íntegro e um link para a visualização de todos os SLOs para a operação, conforme mostrado na seguinte tabela:



A tabela de operações de serviço lista o status do SLI, o número de SLIs íntegros ou não íntegros e o número total de SLOs para cada operação.

Use os SLIs para monitorar a latência, a disponibilidade e outras métricas de operações que medem a integridade operacional de um serviço. Use um SLO para verificar o desempenho e o status da integridade dos serviços e das operações.

Para criar um SLO, faça o seguinte:

- Se uma operação não tiver um SLO, escolha o botão Criar SLO na coluna Status do SLI.
- Se uma operação já tiver um SLO, faça o seguinte:
  - Selecione o botão de opção ao lado do nome da operação.
  - Escolha Criar SLO no ícone de seta para baixo Ações no canto superior direito da tabela.

Para obter mas informações, consulte service level objectives (SLOs).

A coluna Dependências mostra o número de dependências que essa operação chama. Escolha esse número para abrir a guia Dependências filtrada para a operação selecionada.

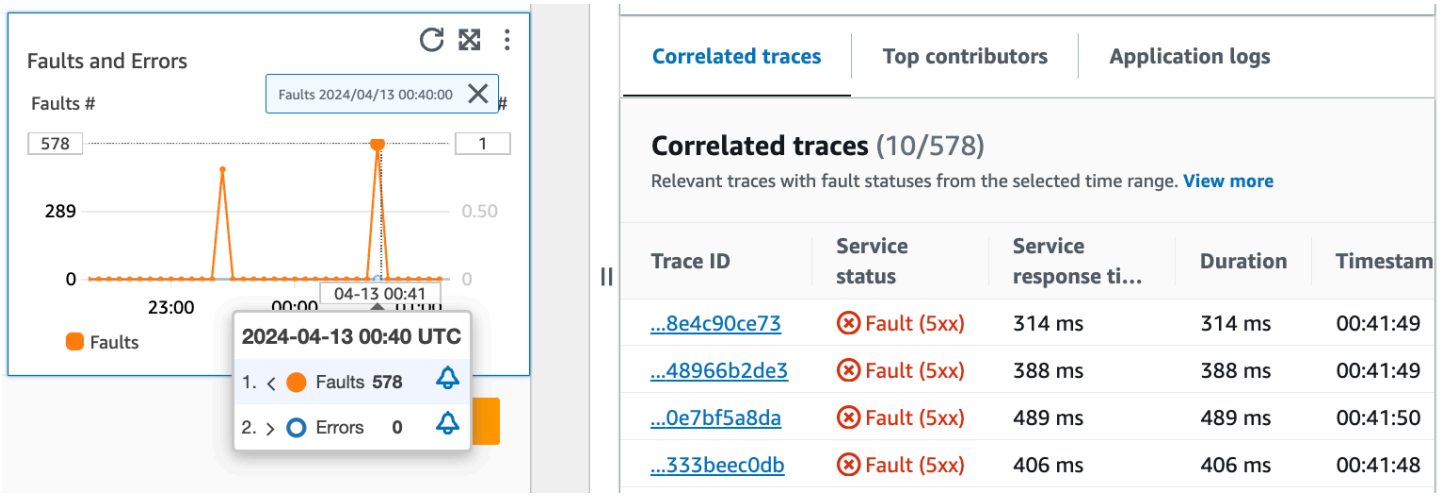
Visualizar métricas de operações de serviço, rastreamentos correlacionados e logs de aplicações

O Application Signals correlaciona as métricas de operação de serviço com os rastreamentos do AWS X-Ray, com o CloudWatch Container Insights e com os logs de aplicações. Use essas métricas para solucionar problemas de integridade operacional. Para visualizar as métricas como informações gráficas, faça o seguinte:

1. Selecione uma operação de serviço na tabela Operações de serviço para visualizar um conjunto de gráficos para a operação selecionada acima da tabela com métricas para Volume e disponibilidade, Latência e Falhas e erros.
2. Passe o cursor sobre um ponto em um gráfico para visualizar mais informações.

3. Selecione um ponto para abrir um painel de diagnóstico que mostra rastreamentos, métricas e logs de aplicações correlacionados para o ponto selecionado no gráfico.

A imagem apresentada a seguir mostra a dica de ferramenta que aparece após passar o cursor sobre um ponto no gráfico e o painel de diagnóstico que é exibido após clicar em um ponto. A dica de ferramenta contém informações sobre o ponto de dados associado no gráfico Falhas e erros. O painel contém Rastreamentos correlacionados, Principais colaboradores e Logs de aplicações associados ao ponto selecionado.



Rastreamentos correlacionados

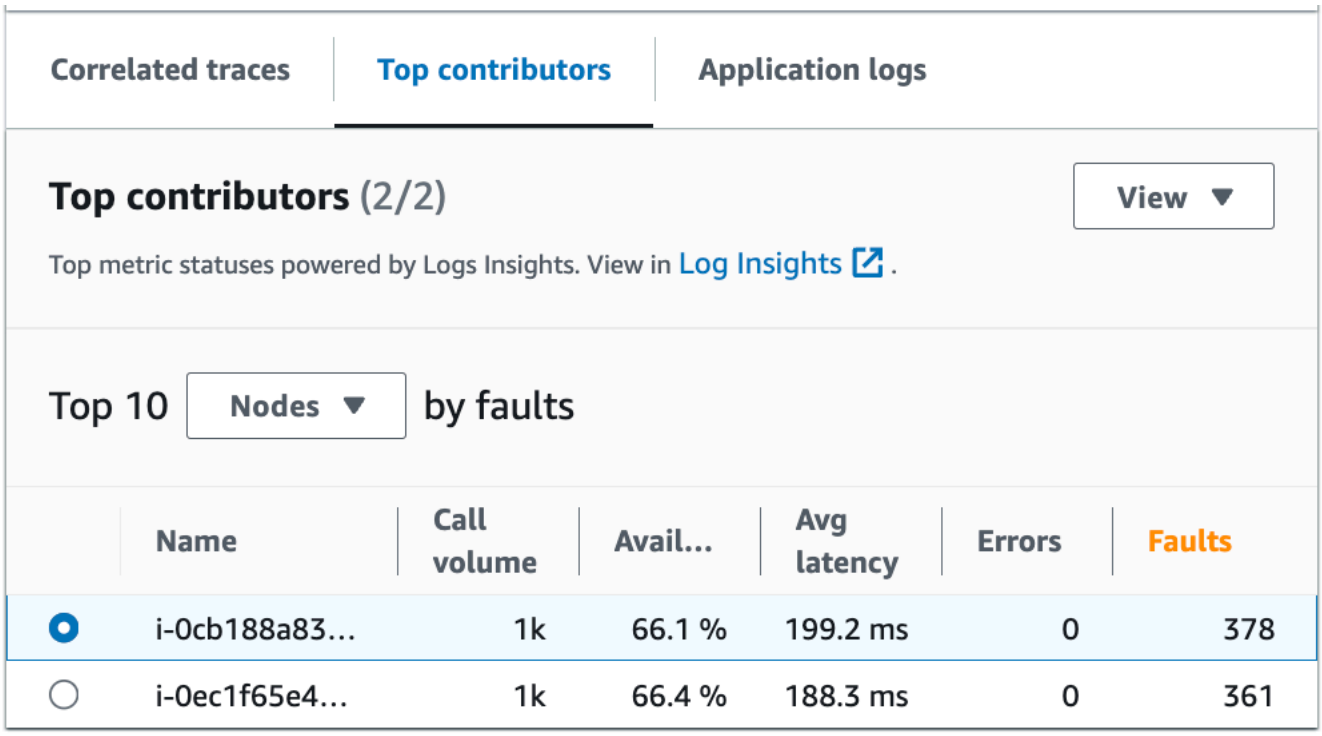
Considere os rastreamentos relacionados para compreender um problema subjacente com um rastreamento. É possível verificar se os rastreamentos correlacionados ou quaisquer nós de serviço associados a eles se comportam de maneira semelhante. Para examinar os rastreamentos correlacionados, escolha um ID de rastreamento na tabela Rastreamentos correlacionados a fim de abrir a página de detalhes do rastreamento do X-Ray para o rastreamento escolhido. A página de detalhes do rastreamento contém um mapeamento dos nós de serviço associados ao rastreamento selecionado e uma linha do tempo dos segmentos de rastreamento.

Principais responsáveis

Confira os principais colaboradores para encontrar as origens de entrada preferenciais para uma métrica. Agrupe os colaboradores por diferentes componentes para pesquisar semelhanças dentro do grupo e compreender como o comportamento de rastreamento difere entre eles.

A guia Principais colaboradores fornece métricas para Volume de chamadas, Disponibilidade, Latência média, Erros e Falhas para cada grupo. A seguinte imagem de exemplo mostra os

principais colaboradores para um conjunto de métricas de uma aplicação implantada em uma plataforma do Amazon EKS:



Os principais colaboradores contêm as seguintes métricas:

- Volume de chamadas: use o volume de chamadas para compreender o número de solicitações por intervalo de tempo para um grupo.
- Disponibilidade: use a disponibilidade para obter a porcentagem de tempo em que nenhuma falha foi detectada para um grupo.
- Latência média: use a latência para verificar o tempo médio de execução das solicitações para um grupo em um intervalo de tempo que depende de há quanto tempo as solicitações que você está investigando foram realizadas. As solicitações que foram realizadas há menos de 15 dias são avaliadas em intervalos de um minuto. As solicitações que foram realizadas entre 15 e 30 dias, inclusive, são avaliadas em intervalos de cinco minutos. Por exemplo, se você estiver investigando solicitações que causaram uma falha há 15 dias, a métrica de volume de chamadas será semelhante ao número de solicitações por intervalo de cinco minutos.
- Erros: o número de erros por grupo medido durante um intervalo de tempo.
- Falhas: o número de falhas por grupo durante um intervalo de tempo.

Principais colaboradores que usam o Amazon EKS ou o Kubernetes

Use as informações sobre os principais colaboradores de aplicações implantadas no Amazon EKS ou no Kubernetes para visualizar métricas de integridade operacional agrupadas por Nó, Pod e Hash do modelo de pod. As seguintes definições se aplicam:

- Um pod corresponde a um grupo de um ou mais contêineres do Docker que compartilham armazenamento e recursos. Um pod é a menor unidade que pode ser implantada em uma plataforma do Kubernetes. Agrupe por pods para verificar se os erros estão relacionados a limitações específicas do pod.
- Um nó corresponde a um servidor que executa pods. Agrupe por nós para verificar se os erros estão relacionados a limitações específicas do nó.
- Um hash de modelo de pod é usado para localizar uma versão específica de uma implantação. Agrupar por hash de modelo de pod para verificar se os erros estão relacionados a uma implantação específica.

Principais colaboradores que usam o Amazon EC2

Use as informações sobre os principais colaboradores de aplicações implantadas no Amazon EKS para visualizar métricas de integridade operacional agrupadas por ID da instância e por grupo do Auto Scaling. As seguintes definições se aplicam:

- Um ID de instância é um identificador exclusivo para a instância do Amazon EC2 executada pelo seu serviço. Agrupe por ID de instância para verificar se os erros estão relacionados a uma instância específica do Amazon EC2.
- Um grupo do Auto Scaling é uma coleção de instâncias do Amazon EC2 que permite diminuir ou aumentar a escala verticalmente dos recursos necessários para atender às solicitações da aplicação. Agrupe por grupo do Auto Scaling se desejar verificar se os erros têm um escopo limitado para as instâncias do grupo.

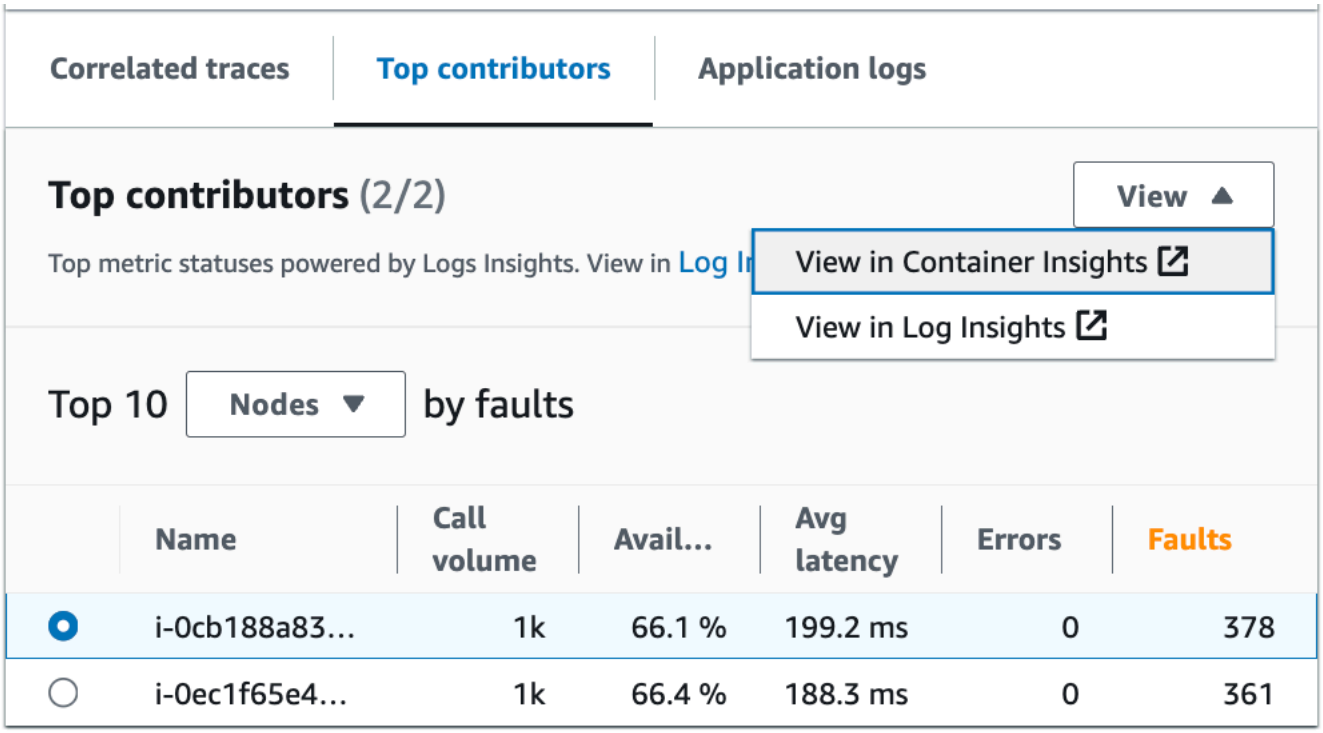
Principais colaboradores que usam uma plataforma personalizada

Use as informações sobre os principais colaboradores para aplicações implantadas usando instrumentação personalizada para visualizar as métricas de integridade operacional agrupadas por Nome do host. As seguintes definições se aplicam:

- Um nome de host identifica um dispositivo, como um endpoint ou uma instância do Amazon EC2, que está conectado a uma rede. Agrupe por nome do host para verificar se os erros estão relacionados a um dispositivo físico ou virtual específico.

Confira os principais colaboradores no Log Insights e no Container Insights

Visualize e modifique a consulta automática que gerou as métricas para os principais colaboradores no Log Insights. Visualize as métricas de desempenho de infraestrutura por grupos específicos, como pods ou nós, no Container Insights. Você pode classificar clusters, nós ou workloads por consumo de recursos e identificar anomalias com rapidez ou mitigar riscos de forma proativa antes que a experiência do usuário final seja afetada. A seguinte imagem mostra como selecionar essas opções:



No Container Insights, é possível visualizar métricas para o contêiner do Amazon EKS ou do Amazon ECS que são específicas para o agrupamento dos seus principais colaboradores. Por exemplo, se você realizou o agrupamento por pod para um contêiner do EKS com a finalidade de gerar os principais colaboradores, o Container Insights mostrará métricas e estatísticas filtradas para seu pod.

No Log Insights, é possível modificar a consulta que gerou as métricas em Principais colaboradores ao usar as seguintes etapas:

1. Selecione Visualizar no Log Insights. A página Log Insights aberta contém uma consulta gerada automaticamente e as seguintes informações:
   - O nome do grupo de clusters do log.
   - A operação que estava sendo investigada com o CloudWatch.

- O agregado da métrica de integridade operacional com a qual você interagiu no gráfico.

Os resultados do log são filtrados automaticamente para mostrar os dados dos últimos cinco minutos antes de você selecionar o ponto de dados no gráfico do serviço.

2. Para editar a consulta, substitua o texto gerado pelas suas alterações. Além disso, é possível usar o Gerador de consultas para ajudar na geração de uma nova consulta ou atualizar a consulta existente.

Logs de aplicações

Use a consulta na guia Logs de aplicações para gerar informações registradas em log para seu grupo de logs ou serviço atuais e inserir um carimbo de data/hora. Um grupo de logs é um grupo de fluxos de logs que você pode definir ao configurar a aplicação.

Use um grupo de logs para organizar os logs com características semelhantes, incluindo as seguintes:

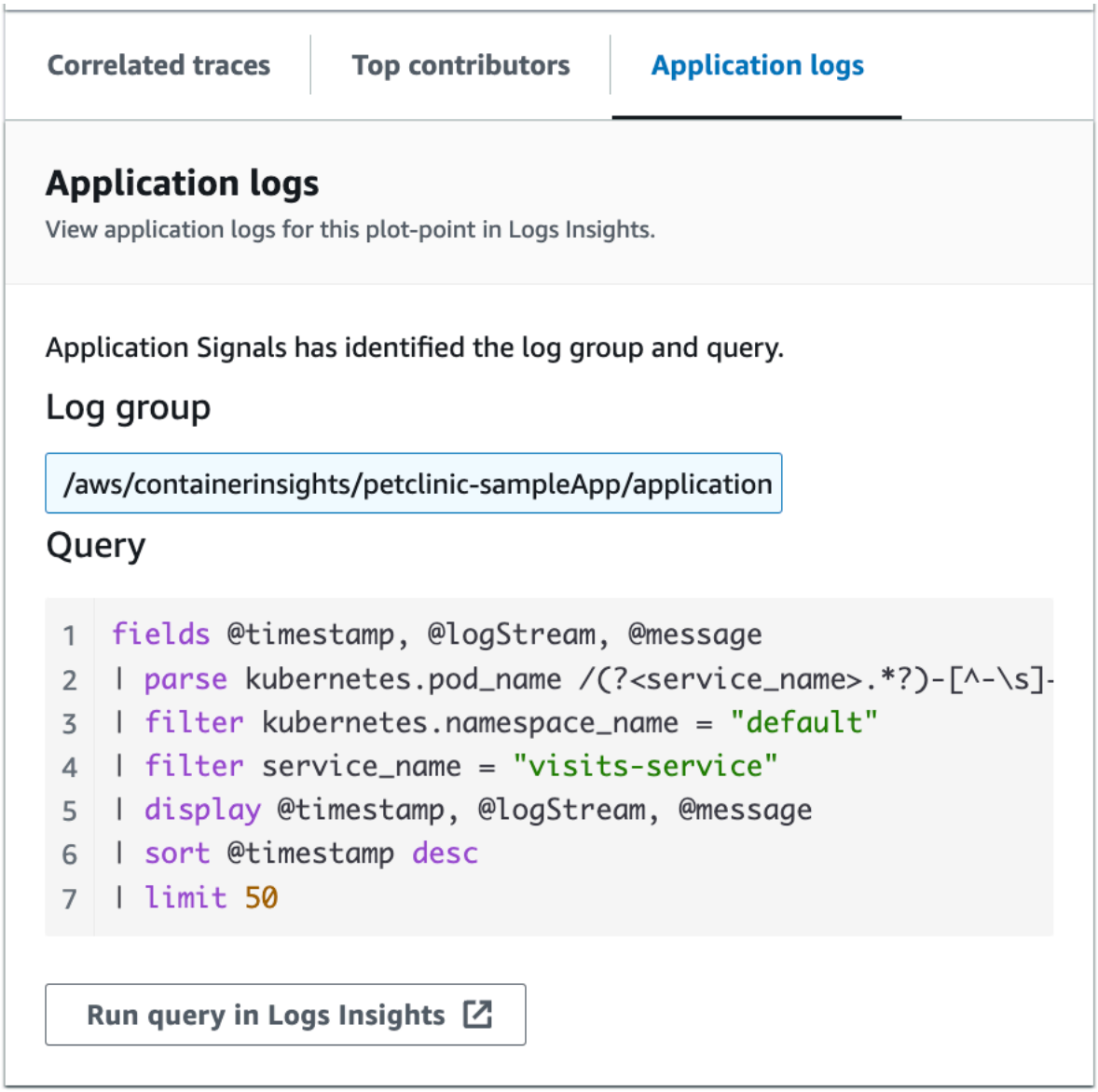
- Captura de logs de uma organização, origem ou função específica.
- Captura de logs que são acessados por um usuário específico.
- Captura de logs de um período específico.

Use esses fluxos de log para rastrear grupos ou períodos específicos. Além disso, é possível configurar regras de monitoramento, alarmes e notificações para esses grupos de logs. Para obter mais informações sobre os grupos de logs, consulte Working with log groups and log streams.

A consulta de logs de aplicações retorna os logs, os padrões de texto recorrentes e as visualizações gráficas para os grupos de logs.

Para executar a consulta, selecione Executar consulta no Logs Insights para executar a consulta gerada automaticamente ou modificá-la. Para editar a consulta, substitua o texto gerado automaticamente pelas suas alterações. Além disso, é possível usar o Gerador de consultas para ajudar na geração de uma nova consulta ou atualizar a consulta existente.

A seguinte imagem mostra a consulta de amostra que é gerada automaticamente com base no ponto selecionado no gráfico de operações de serviço:



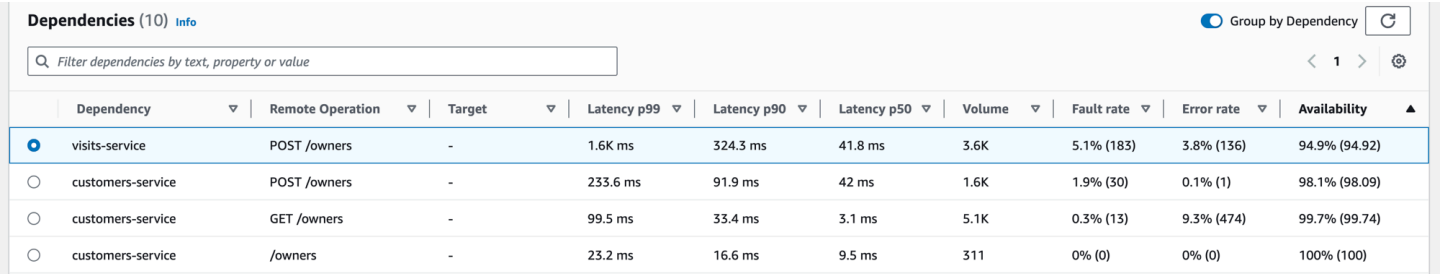
Na imagem apresentada anteriormente, o CloudWatch detectou automaticamente o grupo de logs associado ao ponto selecionado e o incluiu em uma consulta gerada.

Visualizar as dependências do serviço

Escolha a guia Dependências para exibir a tabela Dependências e um conjunto de métricas para as dependências de todas as operações de serviço ou de uma única operação. A tabela contém uma lista de dependências descobertas pelo Application Signals, incluindo métricas de latência, volume de chamadas, taxa de falhas, taxa de erros e disponibilidade.

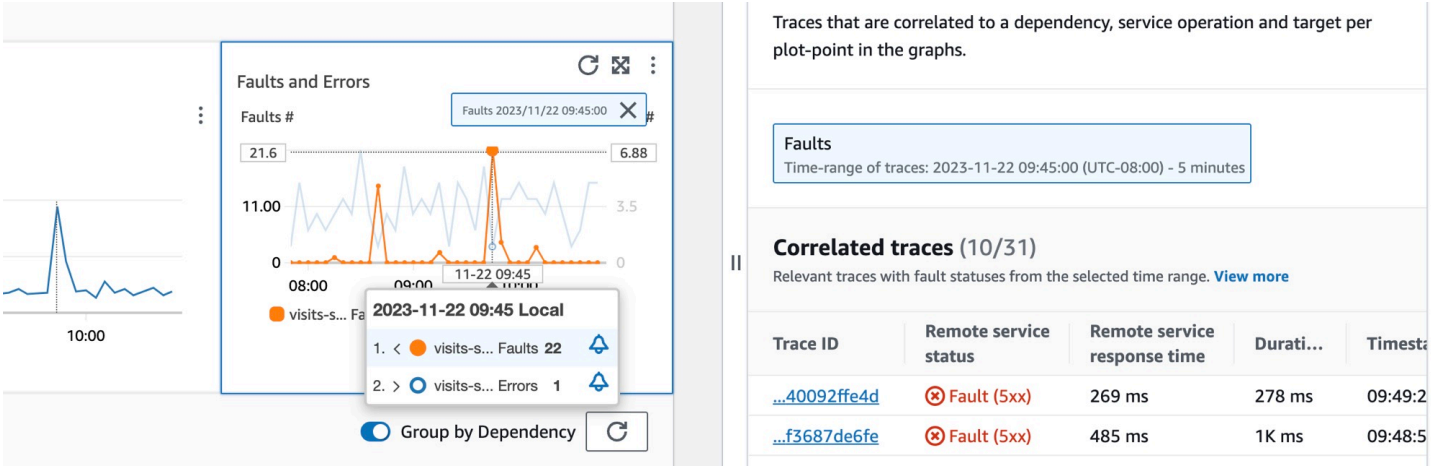
Na parte superior da página, escolha uma operação na lista com ícone de seta para baixo para visualizar as dependências ou escolha Todas para obter as dependências para todas as operações.

Filtre a tabela para facilitar a descoberta do que você está procurando ao escolher uma ou mais propriedades na caixa de texto do filtro. Ao escolher cada propriedade, você será guiado pelos critérios do filtro e verá o filtro completo abaixo da caixa de texto do filtro. Escolha Limpar filtros a qualquer momento para remover o filtro da tabela. Selecione Agrupar por dependência no canto superior direito da tabela para agrupar dependências por nome de serviço e de operação. Quando o agrupamento estiver ativado, expanda ou recolha um grupo de dependências com o ícone + ao lado do nome da dependência.



A coluna Dependência exibe o nome do serviço de dependência, enquanto a coluna Operação remota exibe o nome da operação do serviço. Ao chamar serviços da AWS, a coluna Destino exibe o recurso da AWS, como uma tabela do DynamoDB ou uma fila do Amazon SNS.

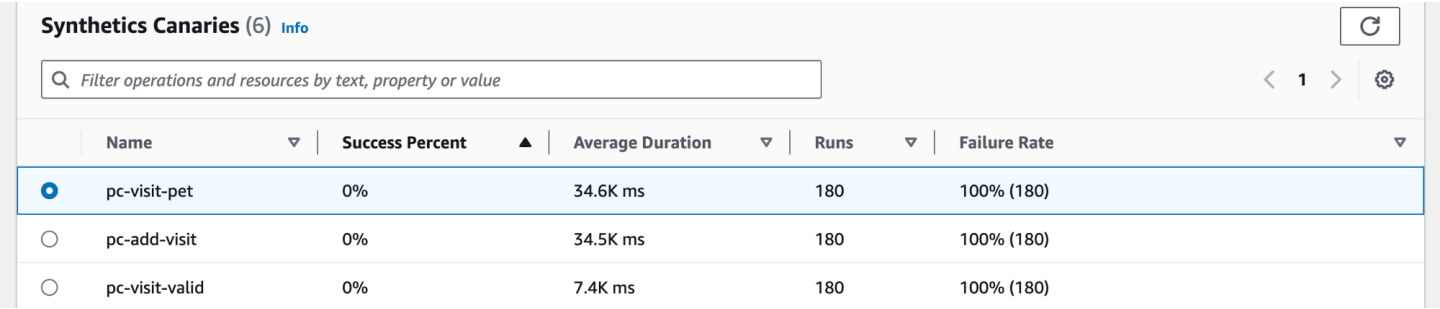
Para selecionar uma dependência, selecione a opção ao lado de uma dependência na tabela Dependências. Isso mostra um conjunto de gráficos que exibem métricas detalhadas para o volume de chamadas, a disponibilidade, as falhas e os erros. Passe o cursor sobre um ponto em um gráfico para visualizar um pop-up que contém informações adicionais. Selecione um ponto em um gráfico para abrir um painel de diagnóstico que mostra rastreamentos correlacionados para o ponto selecionado no gráfico. Escolha um ID de rastreamento na tabela Rastreamentos correlacionados para abrir a página de detalhes do Rastreamento do X-Ray para o rastreamento selecionado.



Visualizar os canários do Synthetics

Escolha a guia Canários do Synthetics para exibir a tabela Canários do Synthetics e um conjunto de métricas para cada canário na tabela. A tabela inclui métricas para porcentagem de sucesso, duração média, execuções e taxa de falhas. Somente os canários que estão habilitados para rastreamento no AWS X-Ray são exibidos.

Use a caixa de texto de filtro na tabela de canários do Synthetics para localizar o canário de seu interesse. Cada filtro criado aparece abaixo da caixa de texto de filtro. Escolha Limpar filtros a qualquer momento para remover o filtro da tabela.



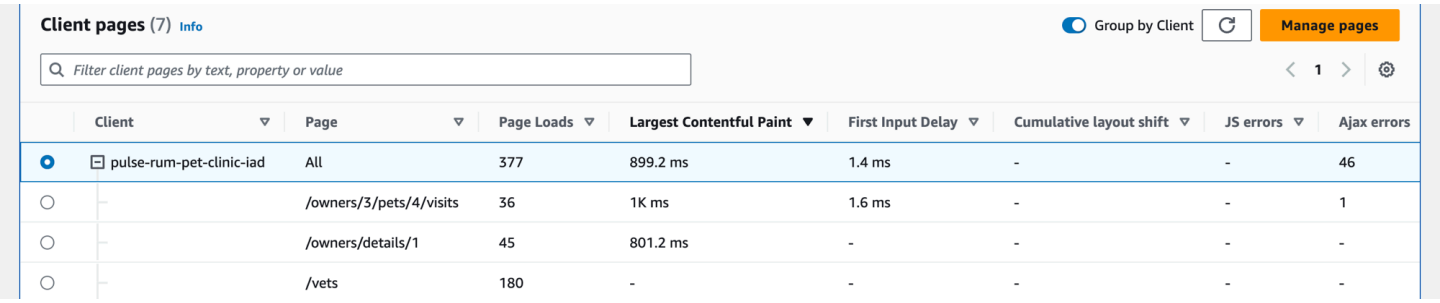
Selecione o botão de opção ao lado do nome do canário para obter um conjunto de guias que contém gráficos detalhados de métricas, incluindo a porcentagem de êxito, os erros e a duração. Passe o cursor sobre um ponto em um gráfico para visualizar um pop-up que contém informações adicionais. Selecione um ponto em um gráfico para abrir um painel de diagnóstico que mostra as execuções do canário que estão correlacionadas ao ponto selecionado. Selecione uma execução do canário e escolha o Runtime para visualizar os artefatos para a execução do canário selecionada, incluindo os logs, os arquivos em HTTP Archive (HAR), as capturas de tela e as etapas sugeridas para ajudar na solução de problemas. Escolha Saiba mais para abrir a página Canários do Cloudwatch Synthetics ao lado de Execuções do canário.

Visualizar as páginas de clientes

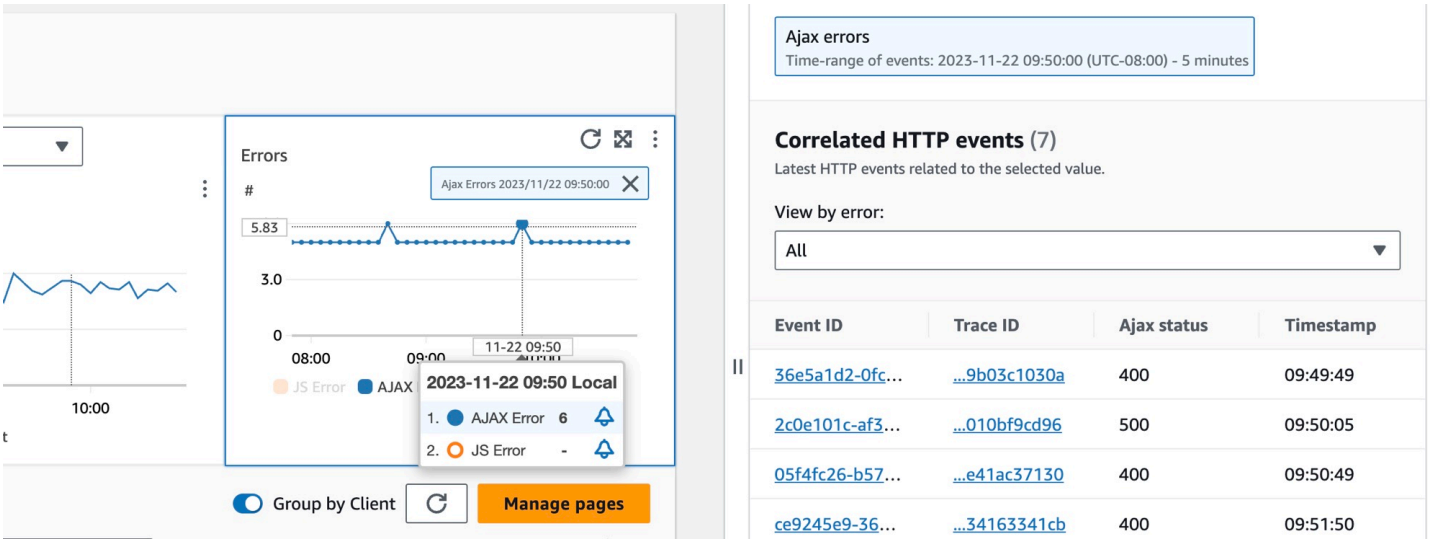
Escolha a guia Páginas de clientes para exibir uma lista de páginas da Web de clientes que chamam o serviço. Use o conjunto de métricas para a página de cliente selecionada a fim de medir a qualidade da experiência do cliente na interação com um serviço ou com uma aplicação. Essas métricas incluem carregamentos de páginas, sinais vitais da Web e erros.

Para exibir as páginas de clientes na tabela, é necessário configurar o cliente Web do CloudWatch RUM para o rastreamento do X-Ray e ativar as métricas do Application Signals para as páginas de clientes. Escolha Gerenciar páginas para selecionar quais páginas estão habilitadas para as métricas do Application Signals.

Use a caixa de texto do filtro para localizar a página de cliente ou o monitoramento de aplicações de seu interesse abaixo da caixa de texto do filtro. Escolha Limpar filtros para remover o filtro da tabela. Selecione Agrupar por cliente para agrupar páginas de clientes por cliente. Depois do agrupamento, escolha o ícone + ao lado do nome de um cliente para expandir a linha e ver todas as páginas desse cliente.



Para selecionar uma página de cliente, selecione a opção ao lado da página de cliente na tabela Páginas de clientes. Você verá um conjunto de gráficos que exibem métricas detalhadas. Passe o cursor sobre um ponto em um gráfico para visualizar um pop-up que contém informações adicionais. Selecione um ponto em um gráfico para abrir um painel de diagnóstico que mostra os eventos correlacionados de navegação de desempenho para o ponto selecionado no gráfico. Escolha um ID de evento na lista de eventos de navegação para abrir a visualização da página do CloudWatch RUM para o evento escolhido.



> **Note**
>
> Para ver erros de Asynchronous JavaScript And XML (AJAX) nas suas páginas de clientes, use a versão 1.15 ou mais recente do cliente Web do CloudWatch RUM.
> Atualmente, até cem operações, canários e páginas de clientes e até 250 dependências podem ser mostradas por serviço.

Visualização da topologia das aplicações e monitoramento da integridade operacional com o mapa de serviços do CloudWatch

⚠ O Application Signals está na versão de pré-visualização para Amazon CloudWatch e está sujeito a alterações.

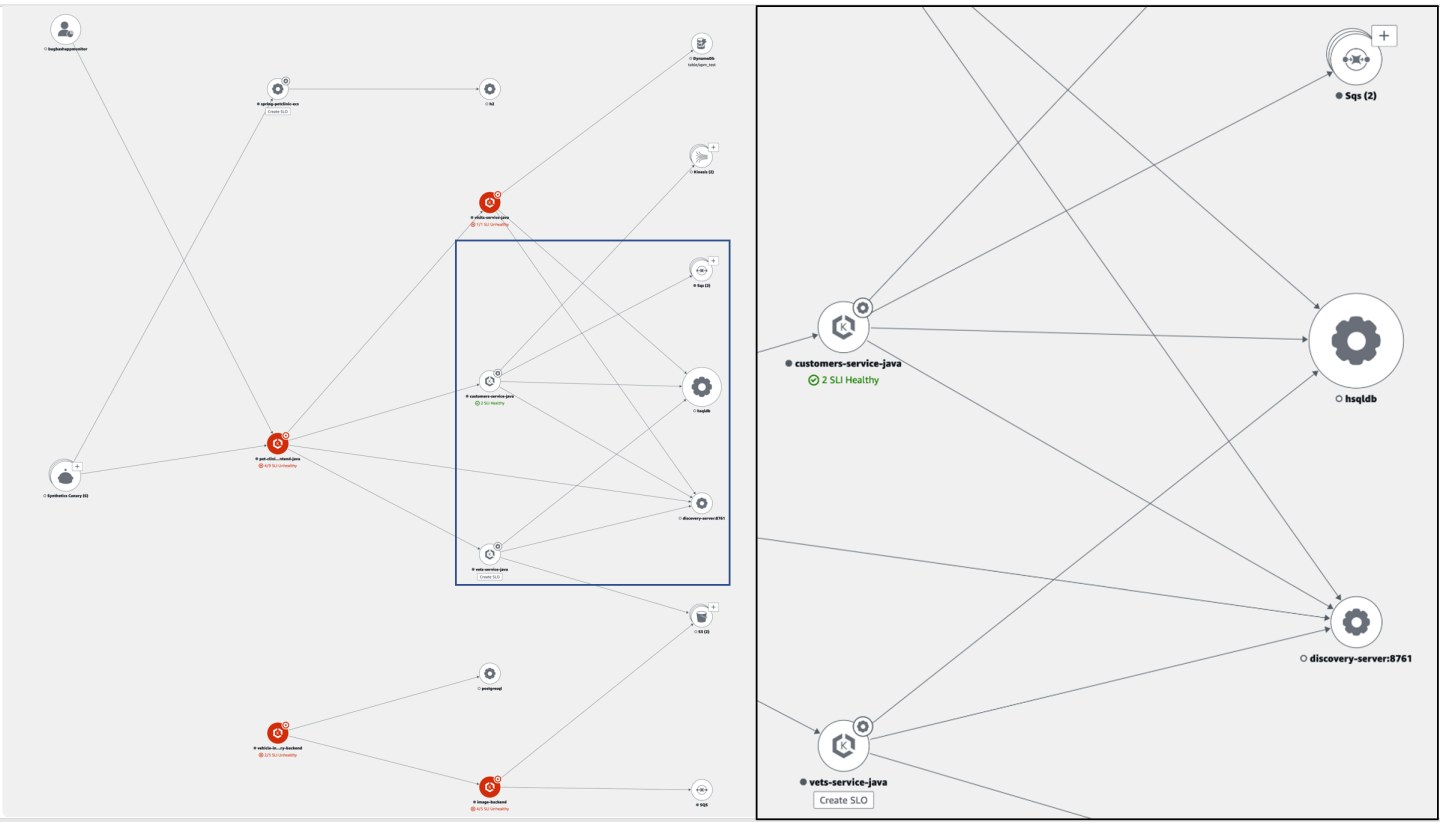
> **Note**
>
> O mapa de serviços do CloudWatch substitui o mapa do ServiceLens. Para ver um mapa da aplicação com base em rastreamentos do AWS X-Ray, abra o Mapa de rastreamento do X-Ray. Escolha Mapa de rastreamento na seção X-Ray no painel de navegação esquerdo do console do CloudWatch.

Use o mapa de serviços para visualizar a topologia de seus clientes de aplicações, canários do Synthetics, serviços e dependências, e para monitorar a integridade operacional. Para visualizar o mapa de serviço, abra o console do CloudWatch e escolha Mapa de serviços, na seção Application Signals, no painel de navegação esquerdo.

Depois de habilitar a aplicação para o Application Signals, use o mapa de serviços para facilitar o monitoramento da integridade operacional da aplicação:

- Visualize as conexões entre o cliente, o canário, o serviço e os nós de dependência para ajudar a entender a topologia da aplicação e o fluxo de execução. Isso é especialmente útil quando os operadores de serviços não são a equipe de desenvolvimento.
- Veja quais serviços estão alcançando ou não seus objetivos de nível de serviço (SLOs). Quando um serviço não está alcançando os SLOs, você pode identificar rapidamente se um serviço ou dependência downstream pode estar contribuindo para o problema ou afetando vários serviços upstream.
- Selecione um cliente individual, um canário do Synthetics, um serviço ou um nó de dependência para visualizar as métricas relacionadas. A página Detalhes do serviço apresenta informações mais detalhadas sobre as operações, as dependências, os canários do Synthetics e as páginas de clientes.
- Filtre e amplie o mapa de serviços para focar em uma parte específica da topologia da aplicação ou visualizar o mapa completo com mais facilidade. Crie um filtro escolhendo uma ou mais propriedades na caixa de texto do filtro. Ao escolher cada propriedade, você é guiado pelos critérios do filtro. Você verá o filtro completo abaixo da caixa de texto do filtro. Escolha Limpar filtros a qualquer momento para remover o filtro.

O exemplo de mapa de serviços, apresentado a seguir, mostra serviços conectados aos componentes com os quais interagem por meio de bordas. Se um SLO for definido, o mapa de serviços também mostrará o status da integridade.

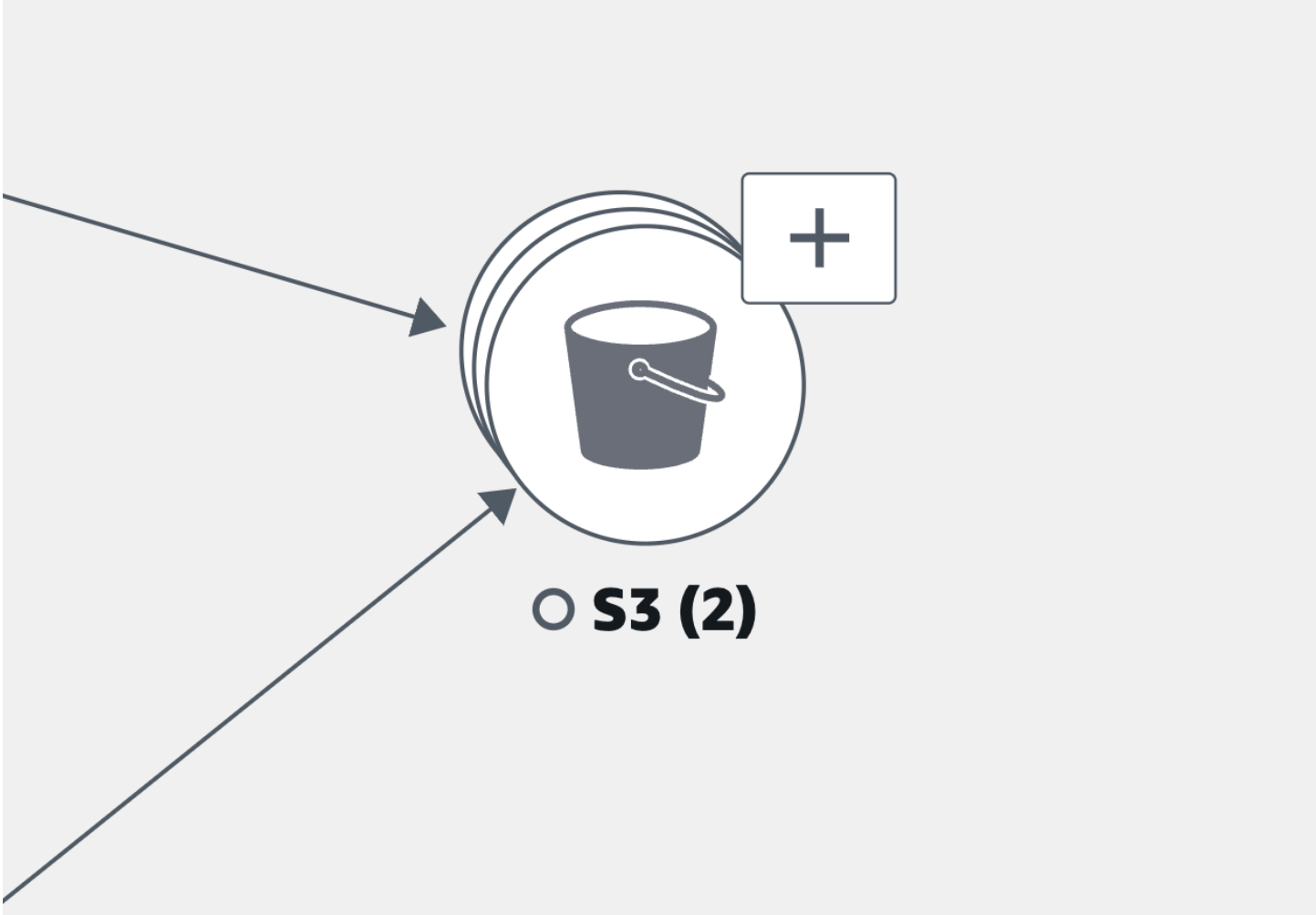


Exploração do mapa de serviços

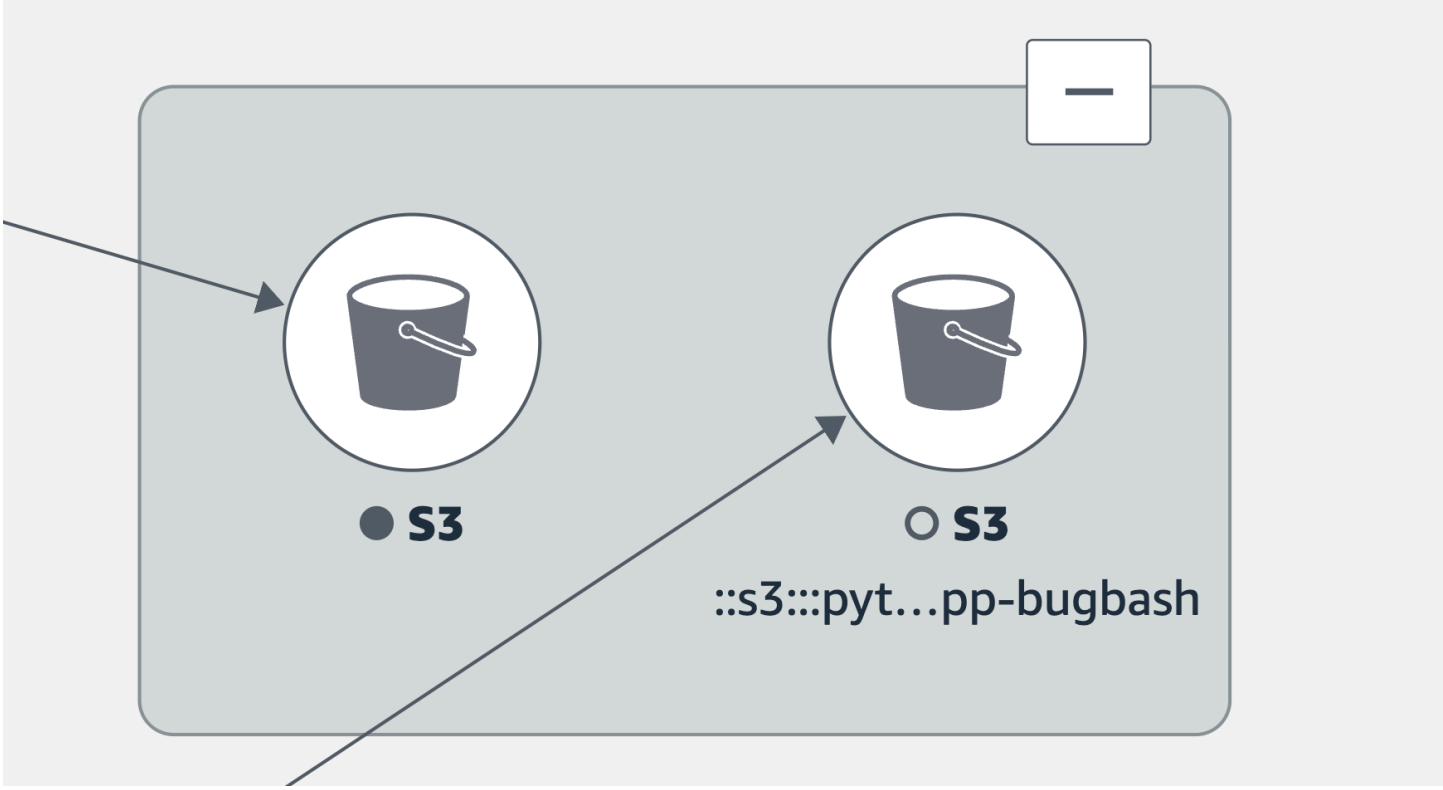
Depois de habilitar a aplicação para o Application Signals, o mapa de serviços exibe nós que representam os serviços e as dependências.

Ative o rastreamento ativo para os clientes do CloudWatch RUM e para os canários Synthetics para visualizar os nós de cliente e de canário no mapa.

Por padrão, canários, clientes do RUM e dependências de serviços da AWS que são do mesmo tipo são agrupados em um único ícone expansível no mapa de serviços. As dependências de serviços externos à AWS não são agrupadas por padrão. Por exemplo, na seguinte imagem, todos os buckets do Amazon S3 estão agrupados em um único ícone expansível:



Na imagem anterior, o rótulo entre o agrupamento do Amazon S3 e o serviço de origem exibe o número de bordas para o grupo entre parênteses sob o ícone da dependência. Selecione o ícone (+) para expandir o grupo e visualizar os elementos individuais, conforme mostrado na seguinte imagem:

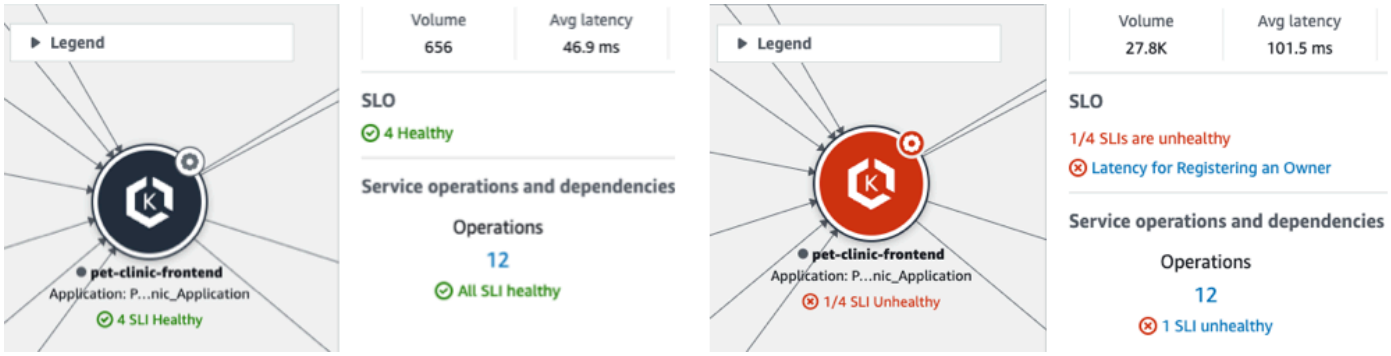


Escolha uma guia para obter informações sobre como explorar cada tipo de nó e as bordas (conexões) entre eles.

View your application services

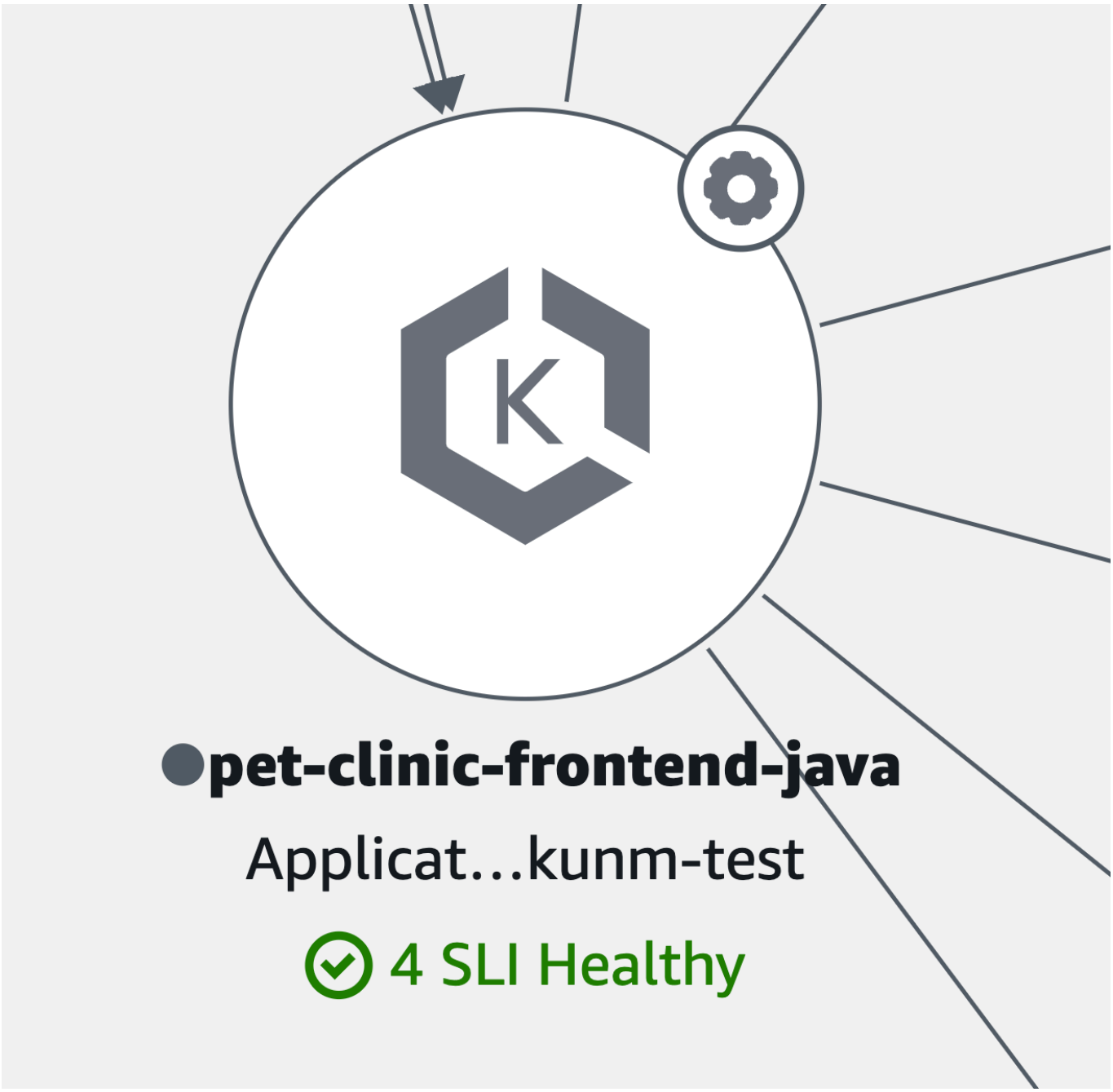
É possível visualizar os serviços de aplicações e o status dos SLOs e dos indicadores de nível de serviço (SLIs) no Mapa de serviços. Se você não criou SLOs para um serviço, escolha o botão Criar SLO abaixo do nó do serviço.

O Mapa de serviços exibe todos os serviços. Além disso, ele mostra os clientes e os canários que consomem o serviço e as dependências que os serviços chamam, conforme mostra na seguinte imagem:



Os seguintes ícones representam exemplos de serviços de aplicações no mapa de serviços:

- Amazon Elastic Kubernetes Service:



- Um contêiner do Kubernetes:

- Amazon Elastic Compute Cloud (Amazon EC2):

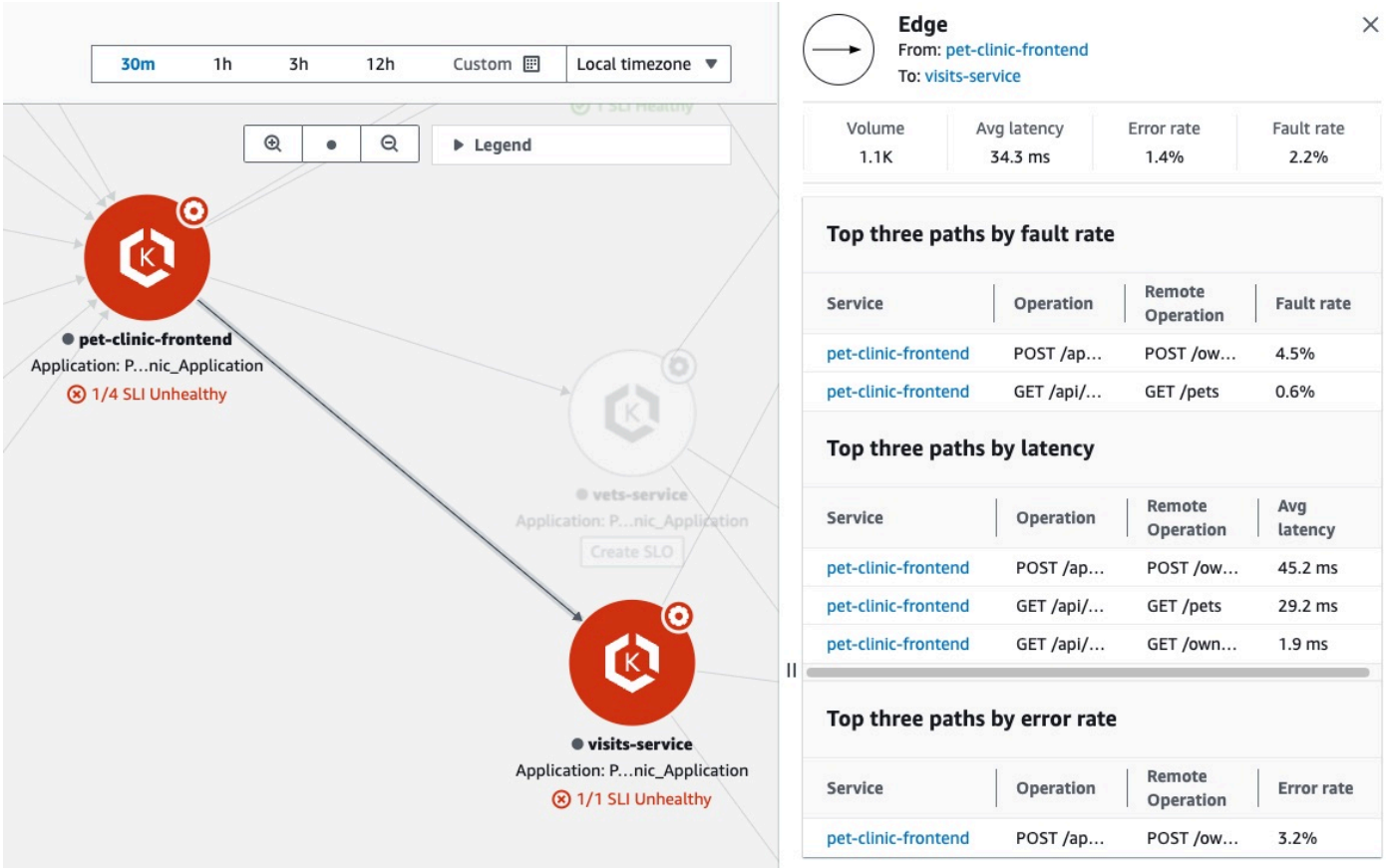
- Outros tipos de serviços de aplicações não listados anteriormente:

Ao selecionar um nó de serviço, um painel é aberto para exibir informações detalhadas do serviço:

- Métricas para volume de chamadas, latência, erro e taxa de falhas.
- O número de SLIs e de SLOs que são `healthy` ou `unhealthy`.
- A opção de visualizar mais informações sobre um SLO.
- O número de operações de serviço, dependências, canários do Synthetics e páginas de clientes.
- A opção de selecionar cada número para abrir a página Detalhes do serviço.

- O nome da aplicação, se você tiver associado o recurso de computação subjacente a uma aplicação usando o AppRegistry ou o cartão do Applications na página inicial do AWS Management Console.
  - Escolha o nome da aplicação para exibir os detalhes da aplicação na página myApplications do console.
- O `Cluster`, o `Namespace` e a `Workload` para os serviços hospedados no Amazon EKS, ou o `Environment` para os serviços hospedados no Amazon ECS ou no Amazon EC2. Para serviços hospedados no Amazon EKS, escolha qualquer link para abrir o CloudWatch Container Insights.

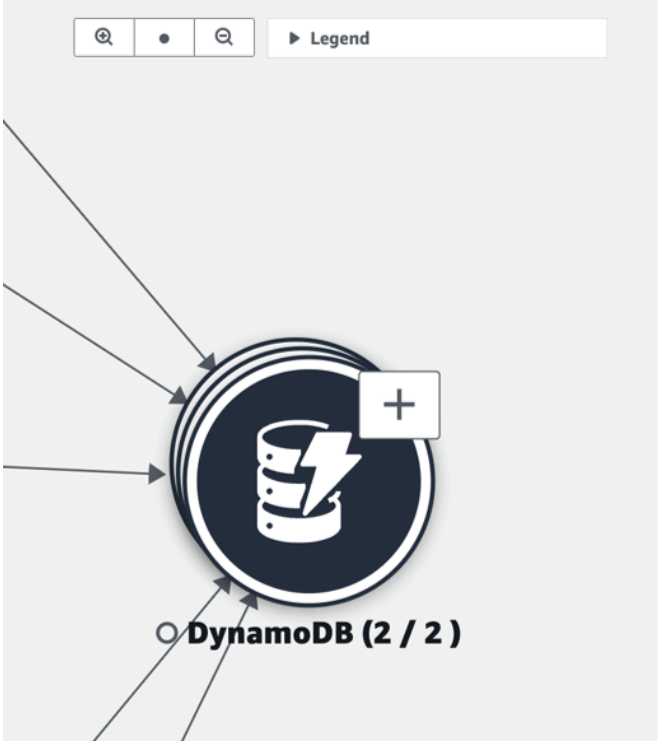
Selecione uma borda ou uma conexão entre um nó de serviço e um serviço downstream ou um nó de dependência. Isso abre um painel que contém os principais caminhos por taxa de falhas, latência e taxa de erros, conforme mostrado na imagem de exemplo apresentada a seguir. Escolha qualquer link no painel para abrir a página Detalhes do serviço e visualizar as informações detalhadas da dependência ou do serviço escolhido.



View dependencies

As dependências da aplicação são exibidas no mapa de serviços, conectadas aos serviços que as chamam.

Escolha um nó de dependência para abrir um painel que contém os principais caminhos por taxa de falhas, latência e taxa de erros. Escolha qualquer link de serviço ou de destino para abrir a página Detalhes do serviço e visualize informações detalhadas sobre o serviço ou sobre o destino de dependência escolhido, conforme mostrado na imagem de exemplo apresentada abaixo:



Volume	Avg latency	Error rate	Fault rate
-	-	-	-

Top three paths by fault rate

Service	Remote operation	Fault rate
No paths with faults		

Top three paths by latency

Service	Remote operation	Avg latency
billing-service-ec2-python	PutItem	282.8 ms
billing-service-python	PutItem	75.6 ms
visits-service-java	PutItem	64.9 ms

Top three paths by error rate

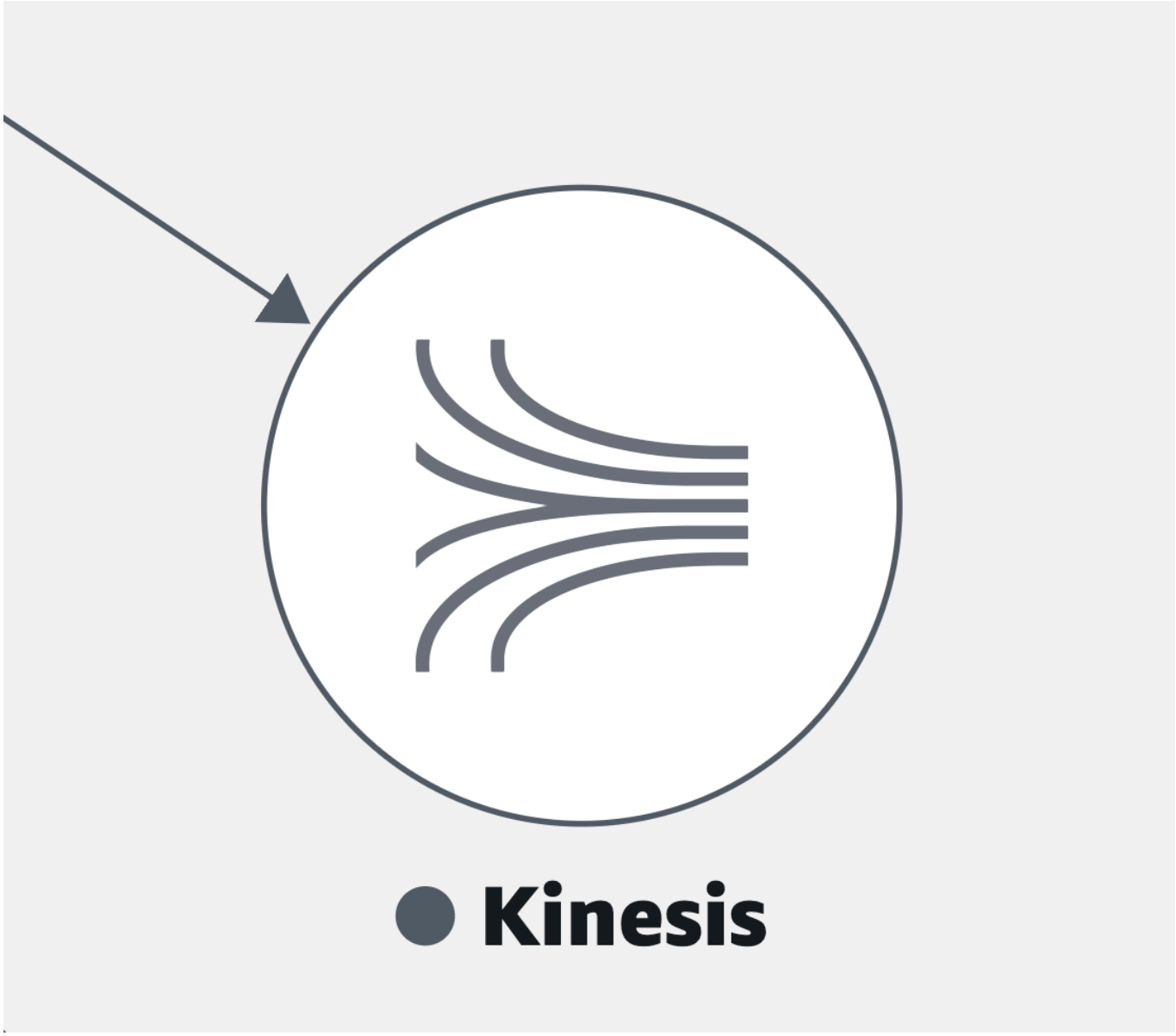
Service	Remote operation	Error rate
visits-service-java	PutItem	9.6%

As dependências de serviço são agrupadas, por padrão, em um único ícone expansível. Selecione o ícone (+), conforme mostrado na imagem anterior, para expandir o grupo e visualizar os elementos individuais.

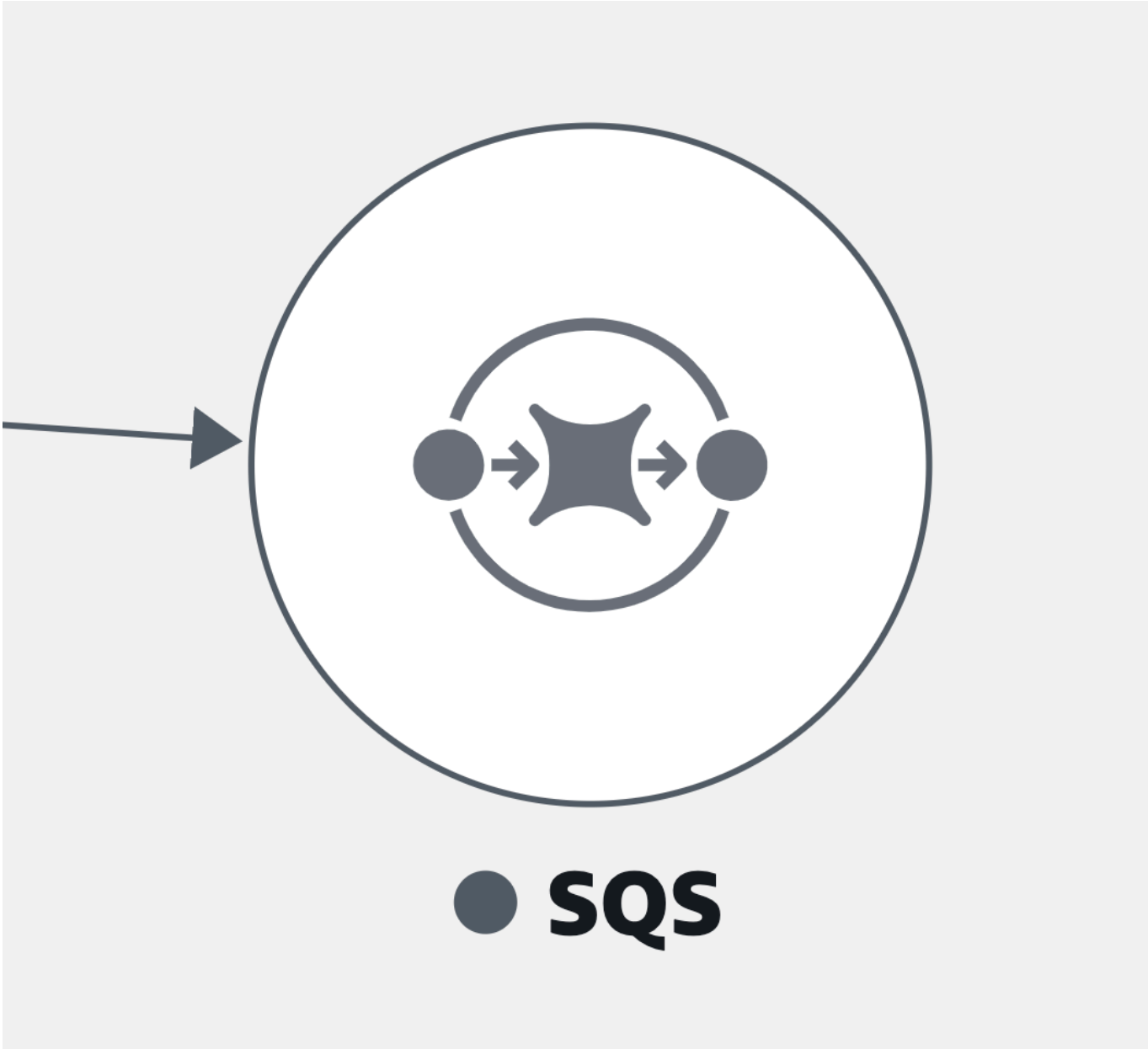
Os seguintes ícones representam exemplos de nós de dependência no mapa de serviços:

- Um bucket do Amazon S3:

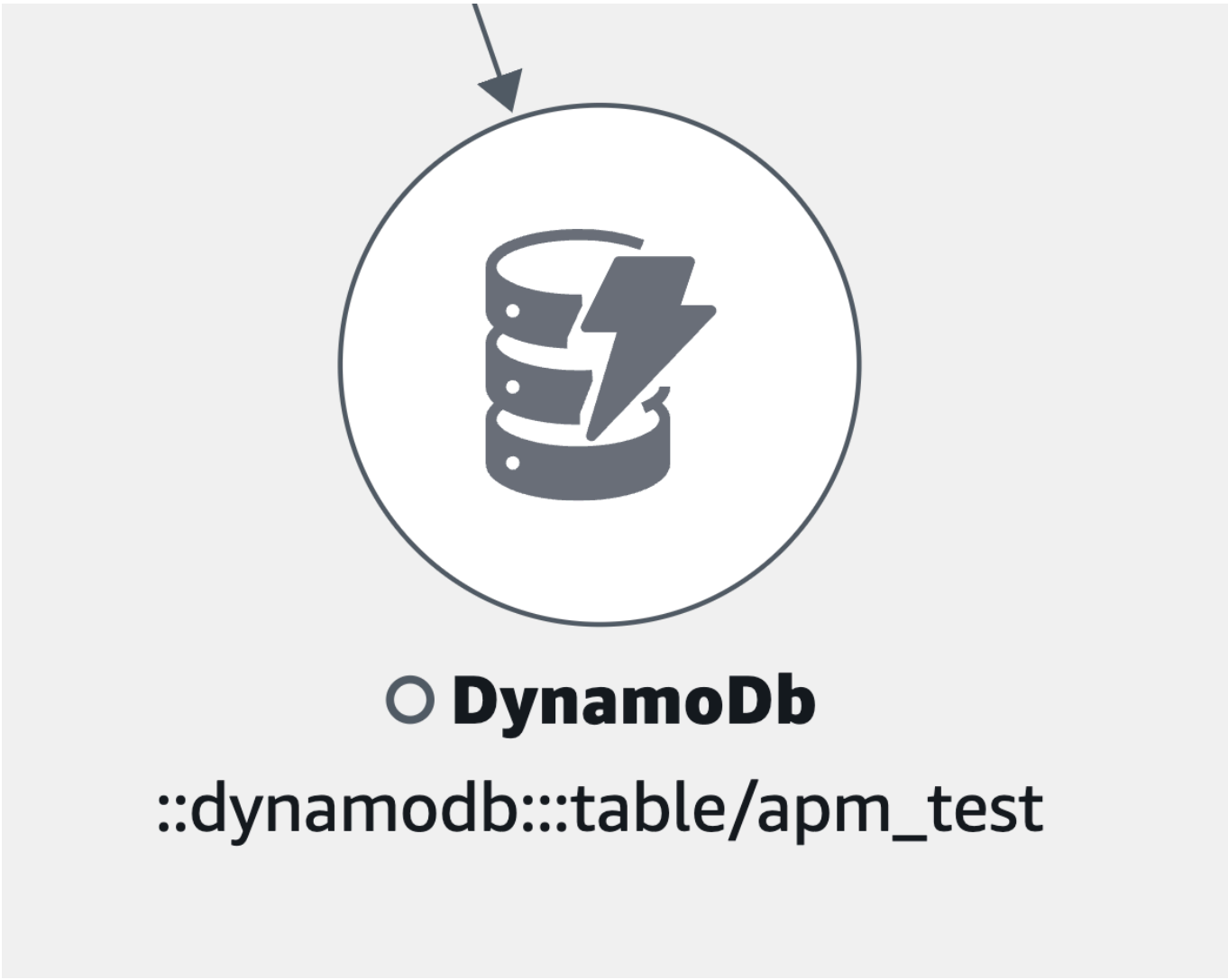
- Um fluxo do Amazon Kinesis:



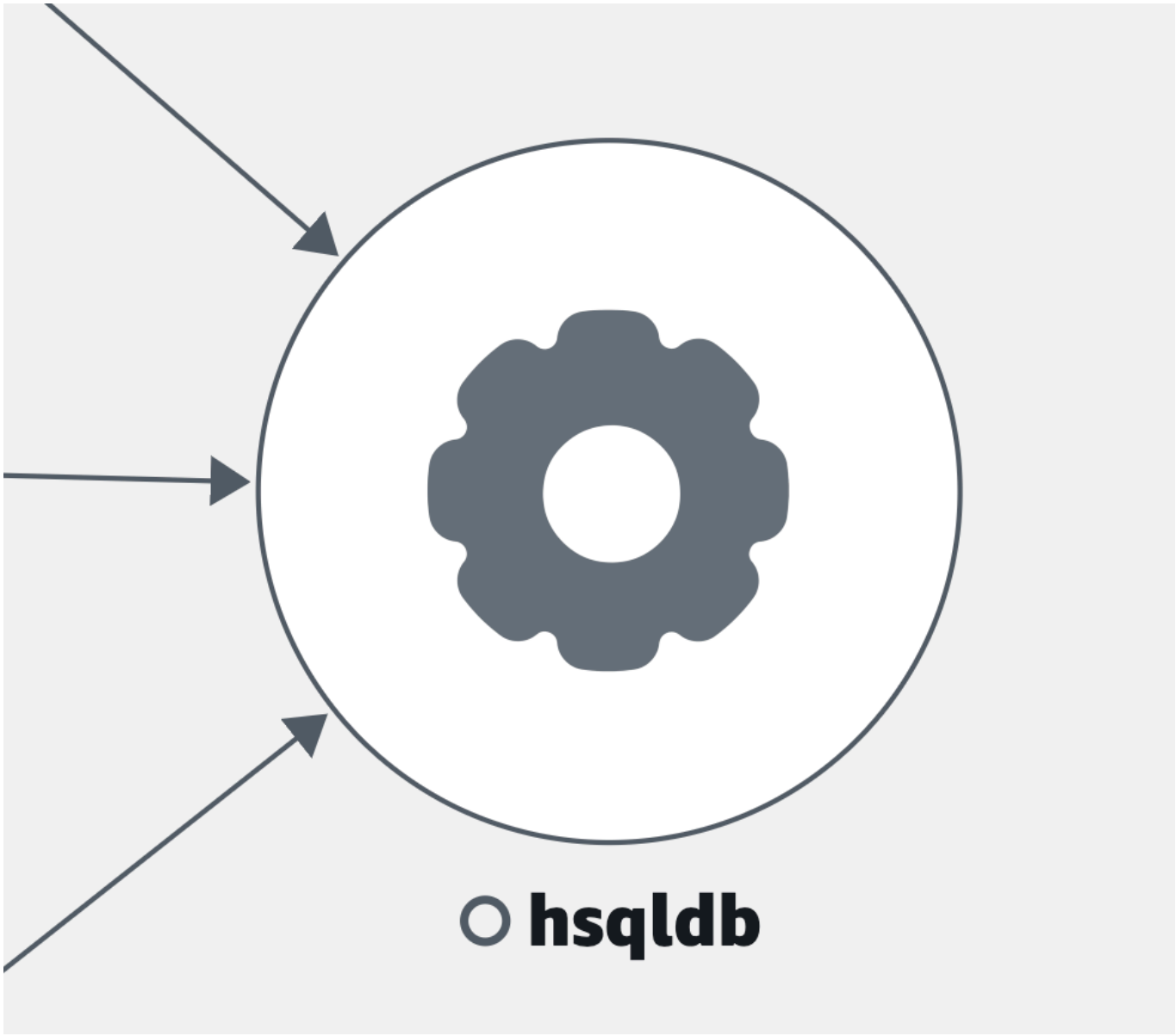
- Amazon Simple Queue Service (Amazon SQS):



- Uma tabela do Amazon DynamoDB:



- Outros tipos de dependência não listados anteriormente:



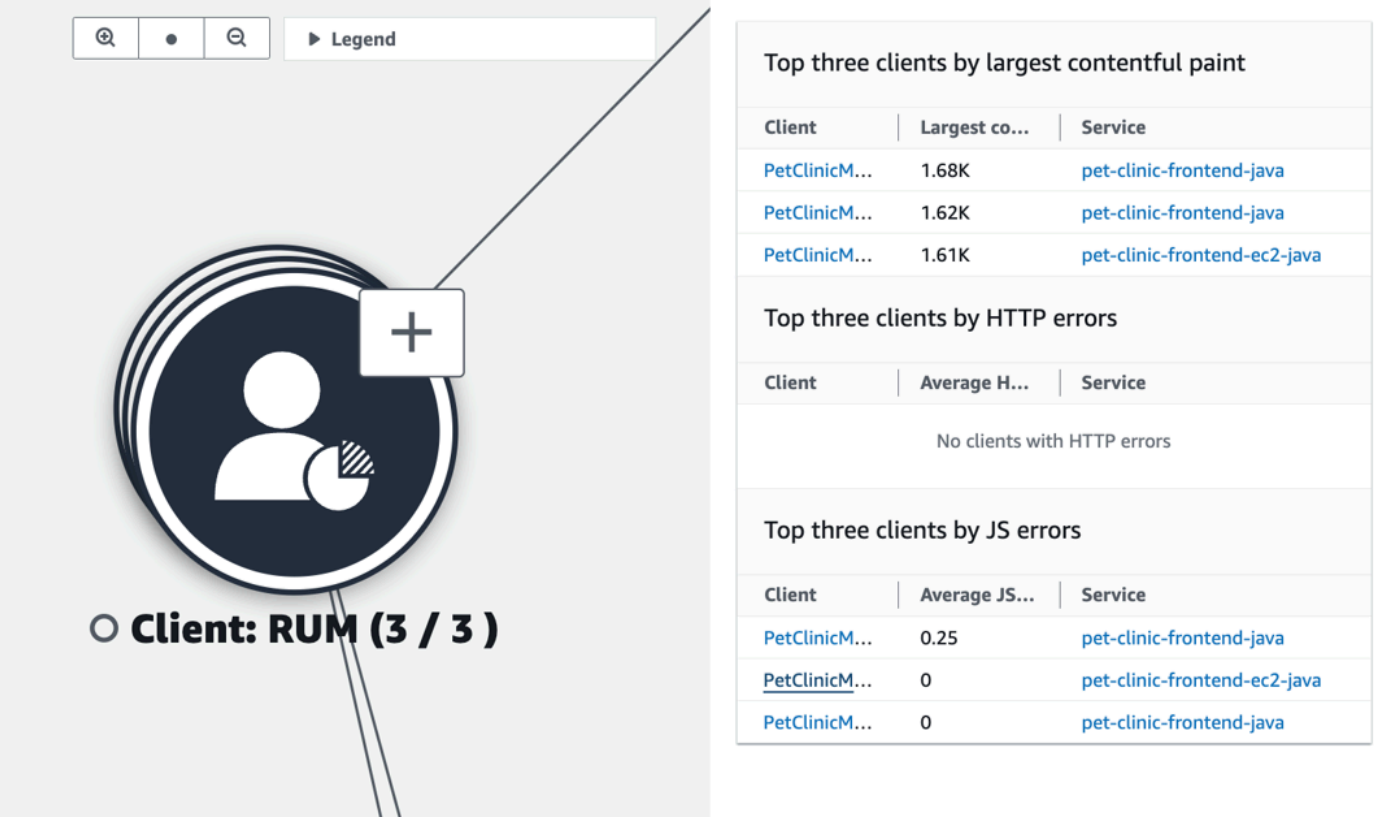
View clients

Depois de ativar o rastreamento do X-Ray para os clientes Web do CloudWatch RUM, eles serão exibidos no mapa de serviços conectados aos serviços que eles chamam.

Escolha um nó de cliente para abrir um painel que exibe as informações detalhadas do cliente:

- Métricas para carregamento de páginas, tempo médio de carregamento, erros e sinais vitais médios da Web.
- Um gráfico exibindo um detalhamento de erros.
- Um link para exibir os detalhes do cliente no CloudWatch RUM.

Os clientes do RUM são agrupados, por padrão, em um único ícone expansível. Selecione o ícone (+), conforme mostrado na imagem apresentada a seguir, para expandir o grupo e visualizar os elementos individuais.



O seguinte ícone representa um exemplo de um cliente do RUM no mapa de serviços:

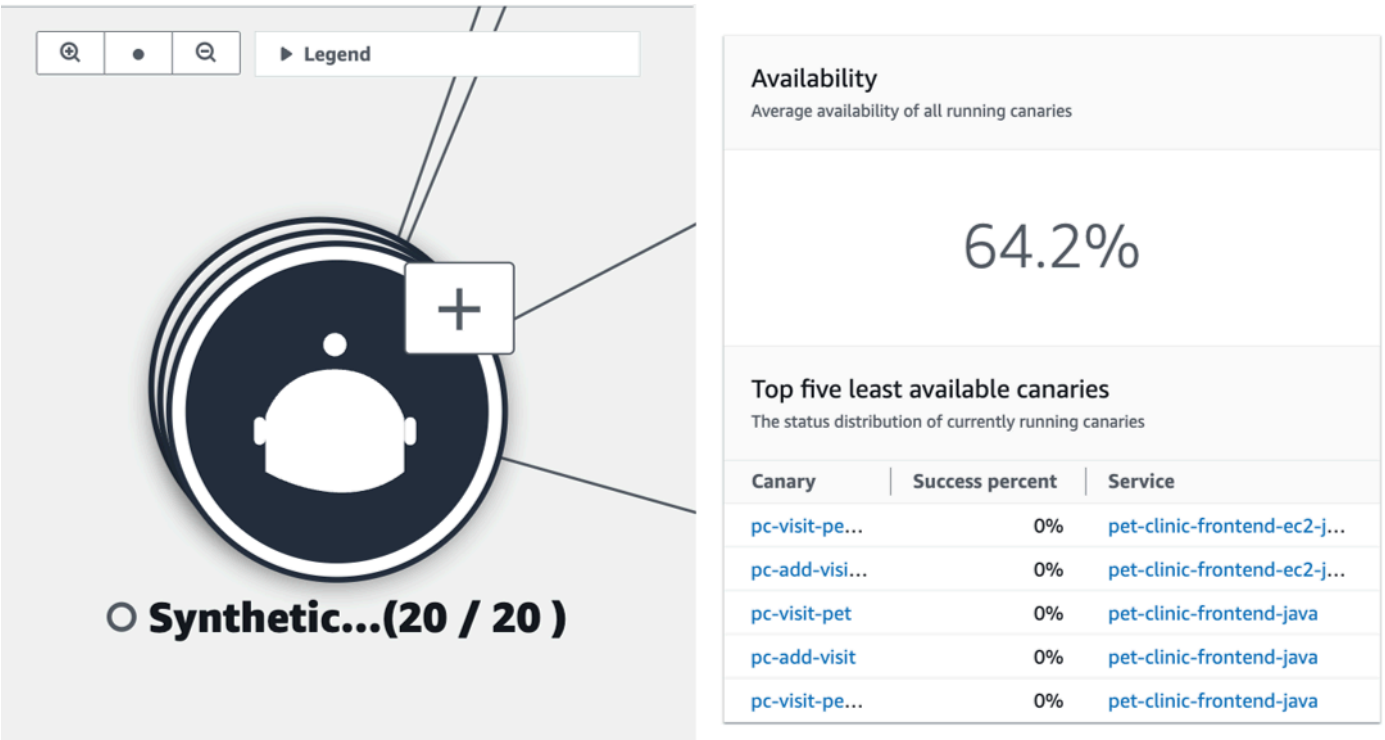
- Um cliente do RUM:

> **Note**
>
> Para ver erros de Asynchronous JavaScript And XML (AJAX) nas suas páginas de clientes, use a versão 1.15 ou mais recente do cliente Web do CloudWatch RUM.

View synthetics canaries

Depois de ativar o rastreamento do AWS X-Ray para os canários do CloudWatch Synthetics, eles serão exibidos no mapa de serviços conectados aos serviços que chamam, conforme mostrado na seguinte imagem de exemplo:

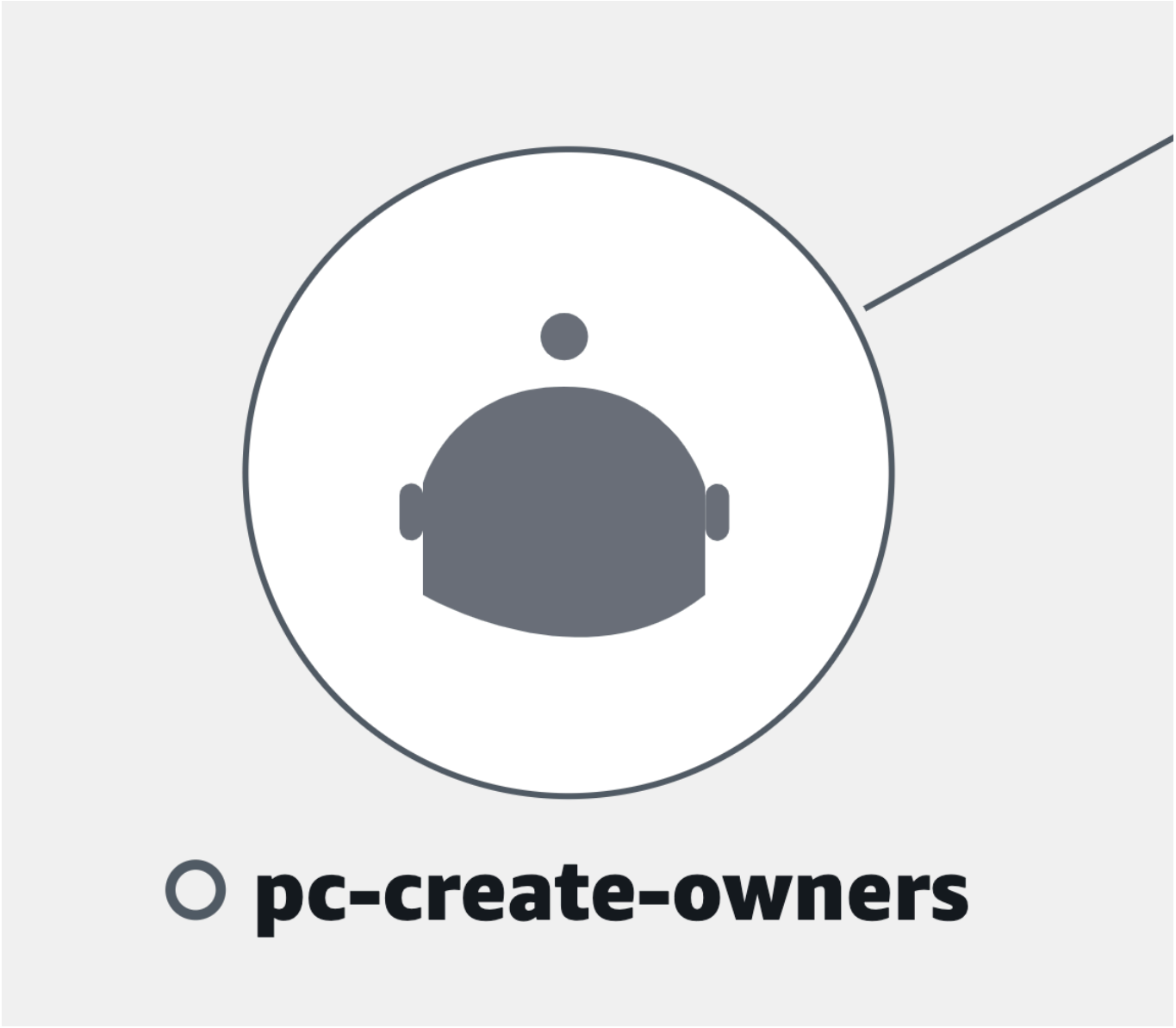
Escolha um nó de canário para abrir um painel que exibe as informações detalhadas do canário, conforme mostrado na seguinte imagem:

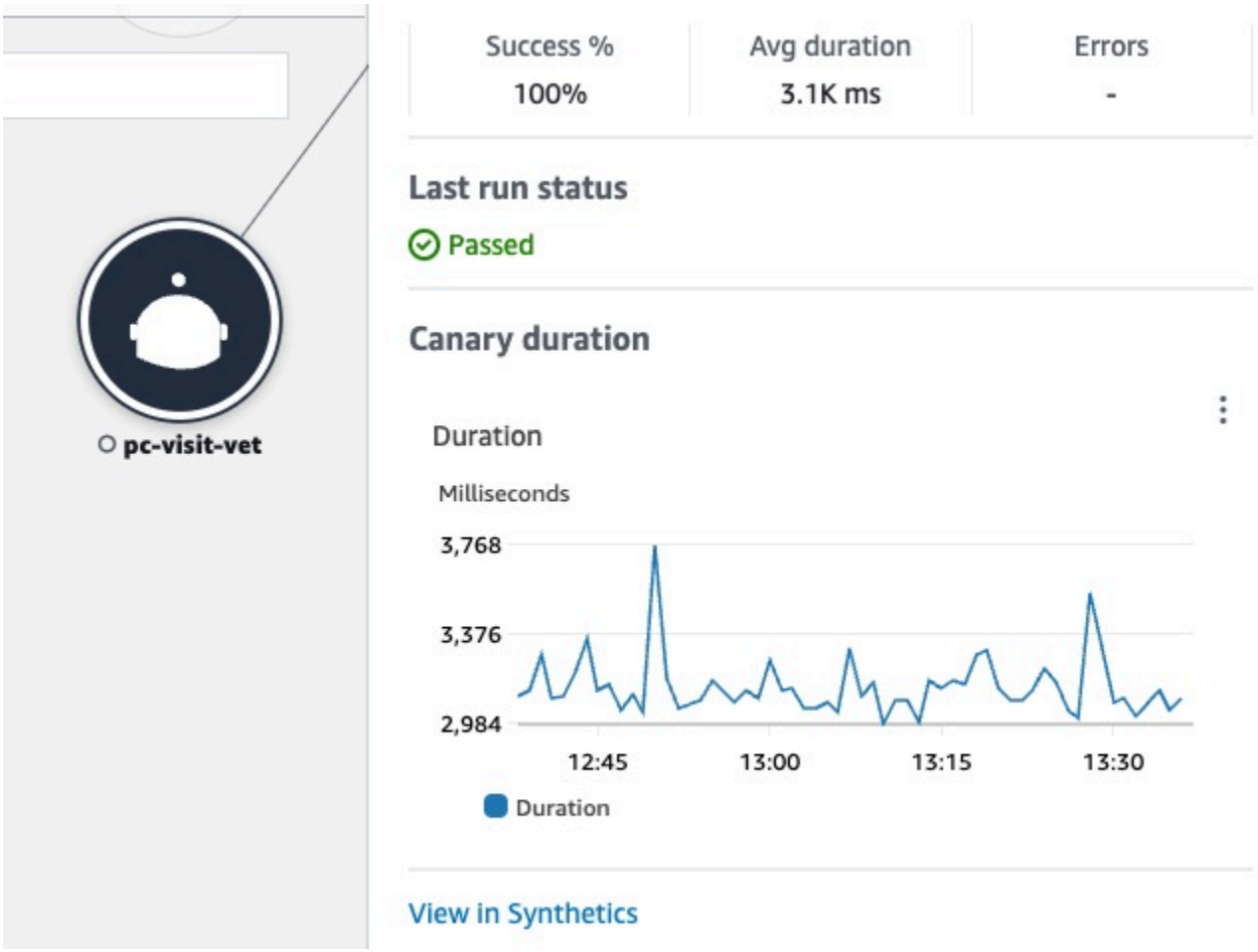


Os canários são agrupados, por padrão, em um único ícone expansível. Selecione o ícone (+), conforme mostrado na imagem anterior, para expandir o grupo e visualizar os elementos individuais.

Os seguintes ícones representam exemplos de clientes no mapa de serviços:

- Um canário do Synthetics:


pc-create-owners



No painel de nós de canário, é possível visualizar o seguinte:

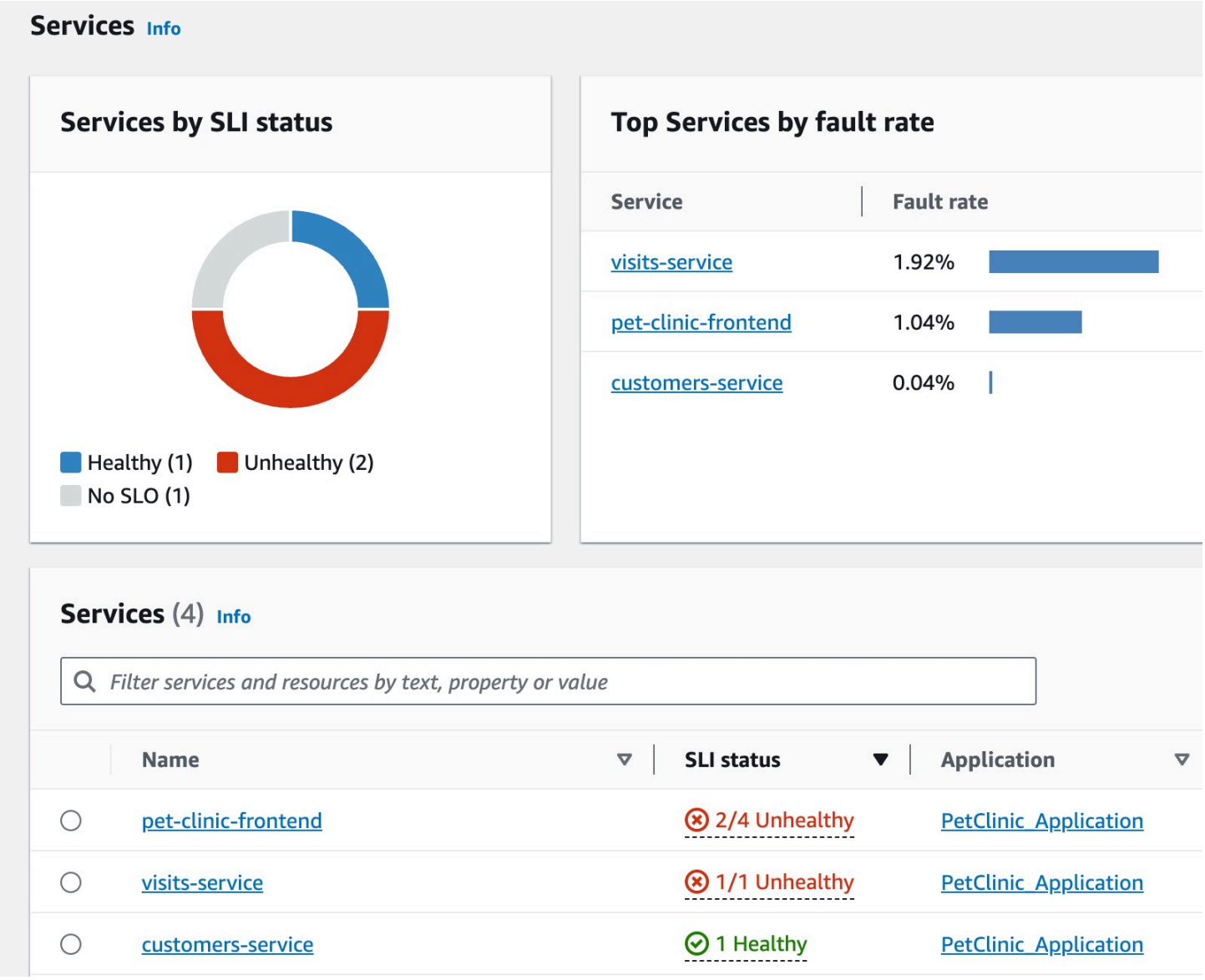
- Métricas para porcentagem de sucesso, duração média e erros.
- O status da última execução de canário.
- Um gráfico exibindo a duração da execução do canário. Passe o mouse sobre uma série de gráficos para obter um pop-up que contém mais informações.
- Um link para exibir detalhes do canário no CloudWatch Synthetics.

Exemplo: usar o Application Signals para resolver um problema de integridade operacional

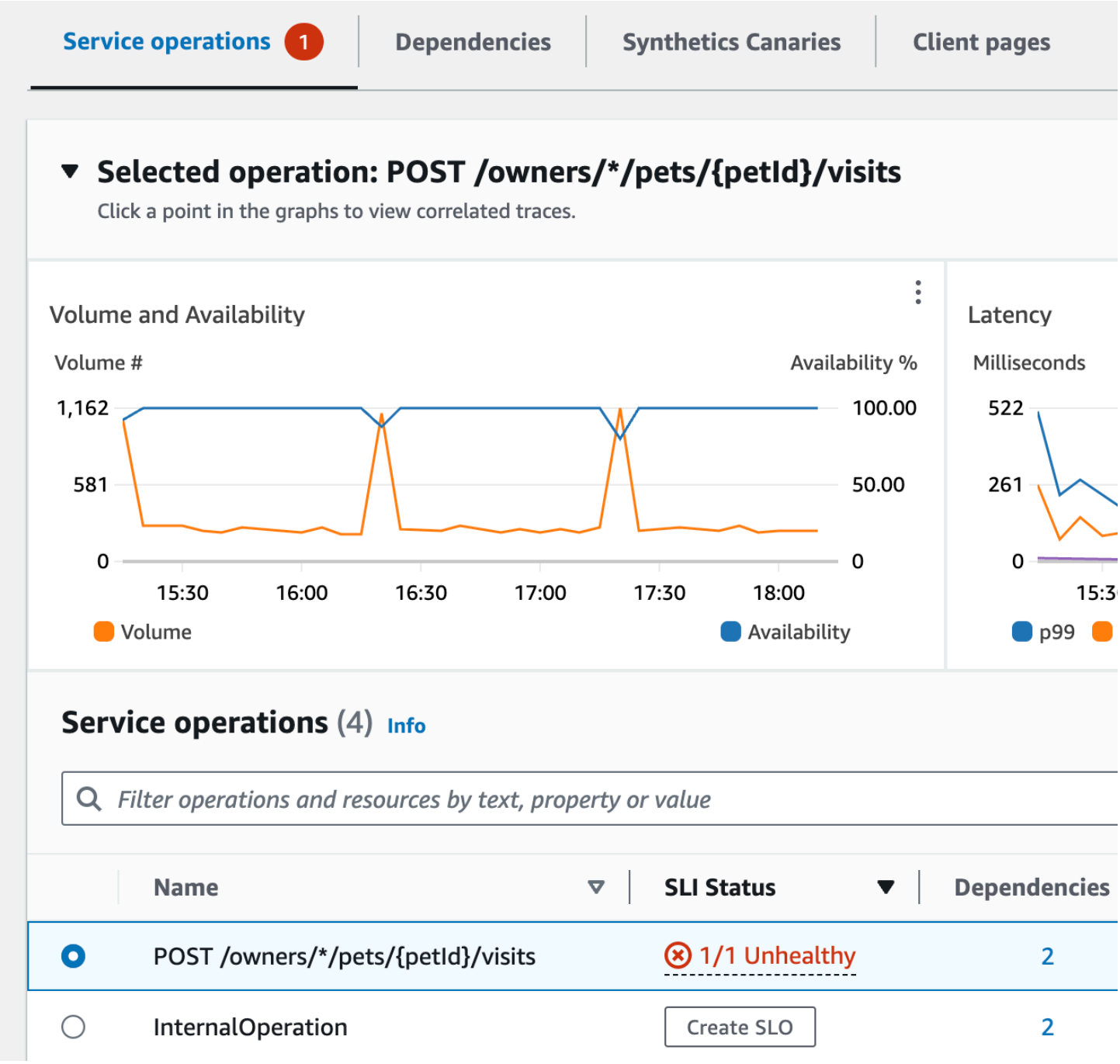
⚠ O Application Signals está na versão de pré-visualização para Amazon CloudWatch e está sujeito a alterações.

O cenário a seguir fornece um exemplo de como o Application Signals pode ser usado para monitorar serviços e identificar problemas de qualidade de serviços. Faça uma pesquisa profunda para identificar as possíveis causas raiz e tomar medidas para resolver o problema. Este exemplo se concentra em uma aplicação de uma clínica para animais de estimação, composta de vários microsserviços que chamam Serviços da AWS como o DynamoDB.

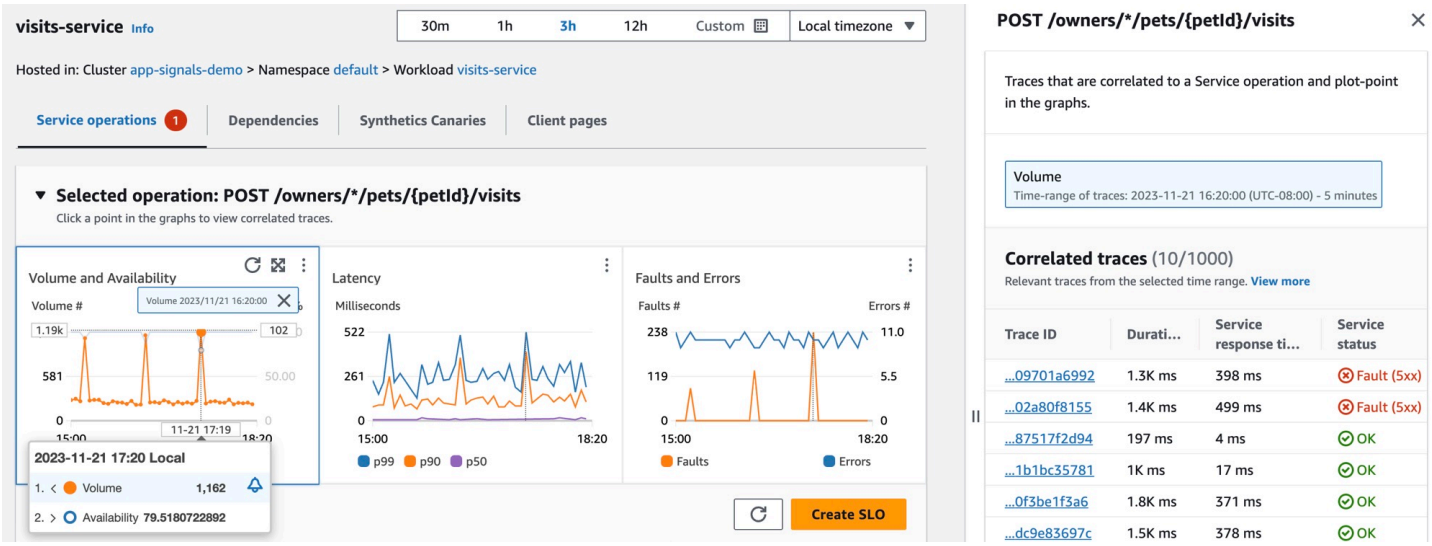
Jane faz parte de uma equipe de DevOps que supervisiona a integridade operacional de uma aplicação de uma clínica para animais de estimação. A equipe de Jane está comprometida em garantir que a aplicação seja altamente disponível e responsiva. Os membros da equipe usam objetivos de nível de serviço (SLOs) para medir a performance da aplicação em relação a esses compromissos e negócios. Ela recebe um alerta sobre vários indicadores de nível de serviço (SLIs) não íntegros. Ela abre o console do CloudWatch e navega até a página Serviços e observa vários serviços em um estado não íntegro.



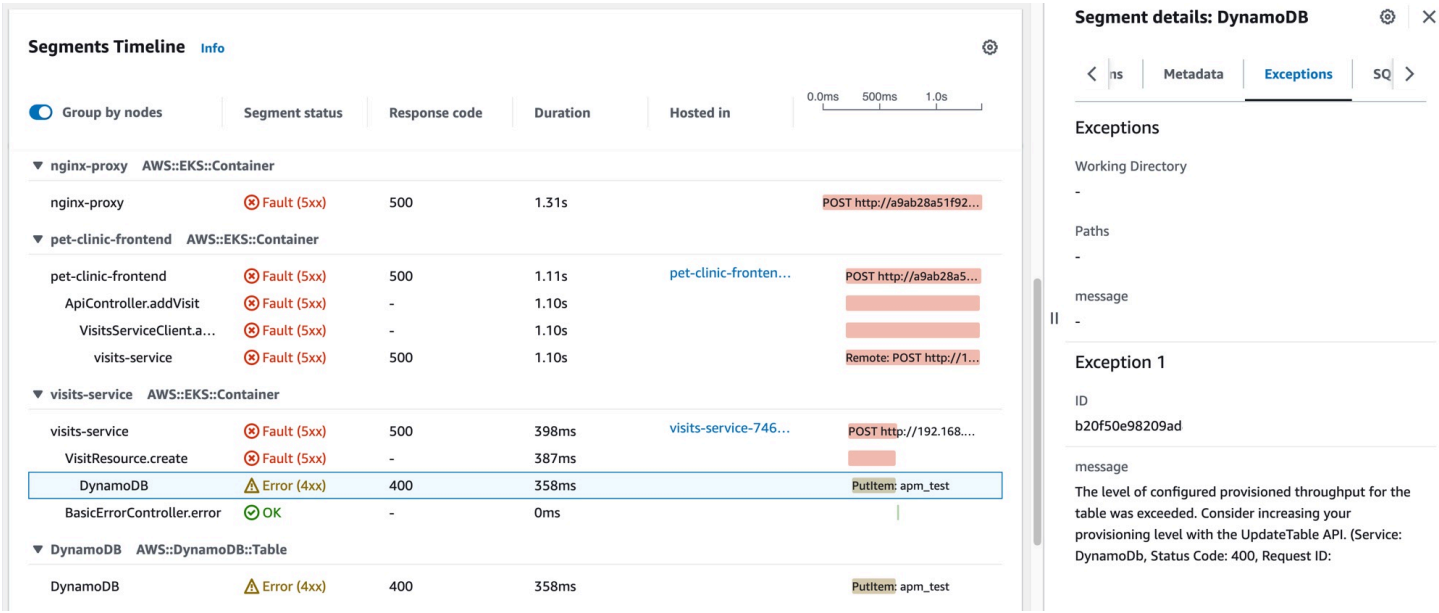
Na parte superior da página, Jane vê que `visits-service` é o melhor serviço por taxa de falhas. Ela seleciona o link no gráfico, que abre a página Detalhes do serviço para o serviço. Ela vê que há uma operação não íntegra na tabela Operações de serviço. Ela seleciona essa operação e vê no gráfico Volume e Disponibilidade que há picos periódicos no volume de chamadas que parecem estar correlacionados a quedas na disponibilidade.



Para observar mais de perto as quedas na disponibilidade do serviço, Jane seleciona um dos pontos de dados de disponibilidade no gráfico. Uma gaveta é aberta mostrando rastreamentos do X-Ray que estão correlacionados ao ponto de dados selecionado. Ela vê que há vários rastreamentos com falhas.



Jane seleciona um dos rastreamentos correlacionados com um status de falha, o que abre a página de detalhes de rastreamentos do X-Ray para o rastreamento selecionado. Jane desce até a seção Linha do tempo dos segmentos e segue o caminho de chamadas até ver que as chamadas para uma tabela do DynamoDB estão retornando erros. Ela seleciona o segmento do DynamoDB e navega até a guia Exceções na gaveta do lado direito.



Jane percebe que um recurso do DynamoDB está configurado incorretamente, resultando em erros durante picos nas solicitações dos clientes. O nível de throughput provisionado da tabela do DynamoDB é excedido periodicamente, resultando em problemas de disponibilidade do serviço e SLIs não íntegros. Com base nessas informações, a equipe consegue configurar um nível mais alto de throughput provisionado e garantir a alta disponibilidade da aplicação.

Coleta de métricas de aplicações padrão

⚠ O Application Signals está em versão de pré-visualização. Se você tiver comentários sobre esse recurso, entre em contato conosco pelo e-mail app-signals-feedback@amazon.com.

O Application Signals coleta métricas de aplicações padrão usando os serviços que descobre. Essas métricas estão relacionadas aos aspectos mais críticos da performance de um serviço, nomeadamente, a latência, as falhas e os erros. As métricas podem ajudar você a identificar problemas, monitorar tendências de performance e otimizar recursos para aprimorar a experiência geral do usuário.

A tabela apresentada a seguir lista as métricas coletadas pelo Application Signals. Essas métricas são enviadas ao CloudWatch no namespace `AppSignals`.

Métrica	Descrição
`Latency`	O atraso antes da transferência de dados começa após a solicitação ser realizada. Unidade: milissegundos
`Faults`	Uma contagem de falhas do lado do servidor HTTP 5XX e de erros de status de extensão do OpenTelemetry. Unidades: nenhuma
`Errors`	Uma contagem de erros do lado do cliente HTTP 4XX. Eles são considerados erros de solicitação que não são causados por problemas de serviço. Portanto, a métrica `Availability` , que é exibida nos painéis do Application Signals, não considera esses erros como falhas de serviço. Unidades: nenhuma

A métrica `Availability` exibida nos painéis do Application Signals é calculada como (1 - **`Faults`**/ Total)*100. As respostas com êxito são todas as respostas sem erros `5XX`. As respostas `4XX` são tratadas como com êxito quando o Application Signals calcula a `Availability`.

Dimensões coletadas e combinações de dimensões

As dimensões apresentadas a seguir são definidas para cada uma das métricas de aplicações padrão. Para obter mais informações sobre dimensões, consulte Dimensões.

Diferentes dimensões são coletadas para as métricas de serviço e para as métricas de dependência. Dentro dos serviços descobertos pelo Application Signals, quando o microsserviço A chama o microsserviço B, este está atendendo à solicitação. Nesse caso, o microsserviço A emite métricas de dependência e o microsserviço B emite métricas de serviço. Quando um cliente chama o microsserviço A, este está atendendo à solicitação e emite métricas de serviço.

Dimensões para métricas de serviço

As dimensões apresentadas a seguir são coletadas para as métricas de serviço.

Dimensão	Descrição
`Service`	O nome do serviço.
`Operation`	O nome da operação de API ou de outra atividade.
`HostedIn.EKS.Cluster`	O nome do cluster do Amazon EKS no qual os serviços estão em execução. Essa dimensão será coletada somente se os serviços estiverem em execução no Amazon EKS.
`HostedIn.K8s.Namespace`	O nome do namespace do Kubernetes no qual os serviços estão em execução. Essa dimensão será coletada somente se os serviços estiverem em execução no Amazon EKS.
`HostedIn.Environment`	O nome definido pelo usuário do ambiente no qual os serviços estão em execução.

Dimensão	Descrição
	Essa dimensão será coletada somente se os serviços estiverem em execução em um ambiente diferente do Amazon EKS.

Ao visualizar essas métricas no console do CloudWatch, é possível optar por visualizá-las com as combinações de dimensões apresentadas a seguir.

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`

Para as plataformas diferentes do Amazon EKS, também é possível visualizar as métricas de serviço com as combinações de dimensões apresentadas a seguir.

- `Service`, `Operation`, `HostedIn.Environment`
- `Service`, `HostedIn.Environment`

Dimensões para métricas de dependência

As dimensões apresentadas a seguir são coletadas para as métricas de dependência.

Dimensão	Descrição
`Service`	O nome do serviço.
`Operation`	O nome da operação de API ou de outra atividade.
`RemoteService`	O nome do serviço remoto que está sendo invocado.
`RemoteOperation`	O nome da operação de API que está sendo invocada.
`HostedIn.EKS.Cluster`	O nome do cluster do Amazon EKS no qual os serviços estão em execução. Essa dimensão será coletada somente se os serviços estiverem em execução no Amazon EKS.

Dimensão	Descrição
`HostedIn.K8s.Namespace`	O nome do namespace do Kubernetes no qual os serviços estão em execução. Essa dimensão será coletada somente se os serviços estiverem em execução no Amazon EKS.
`K8s.RemoteNamespace`	O nome do namespace do Kubernetes no qual os serviços de dependência estão em execução. Essa dimensão será coletada somente se os serviços estiverem em execução no Amazon EKS.
`RemoteTarget`	O nome do recurso invocado pelas chamadas remotas. Essa dimensão não terá valor se as chamadas remotas não forem direcionadas para recursos específicos. Essa dimensão será coletada somente se os serviços estiverem em execução no Amazon EKS.
`HostedIn.Environment`	O nome definido pelo usuário do ambiente no qual os serviços estão em execução. Essa dimensão será coletada somente se os serviços estiverem em execução em um ambiente diferente do Amazon EKS.

Ao visualizar essas métricas no console do CloudWatch, é possível optar por visualizá-las com as combinações de dimensões apresentadas a seguir.

Execução em qualquer plataforma

- `RemoteService`

Execução em clusters do Amazon EKS

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`, `RemoteTarget`

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `RemoteService`, `RemoteOperation`,
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `RemoteService`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`, `RemoteTarget`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`

Execução em plataformas diferentes dos clusters do Amazon EKS

- `Service`, `Operation`, `HostedIn.Environment`
- `Service`, `HostedIn.Environment`
- `Service`, `Operation`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`,
- `Service`, `HostedIn.Environment`, `RemoteService`
- `Service`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`,

Uso do monitoramento sintético

É possível usar o Amazon CloudWatch Synthetics para criar canaries, scripts configuráveis que são executados de acordo com uma programação, para monitorar os endpoints e as APIs. Os canaries seguem as mesmas rotas e executam as mesmas ações que um cliente, o que possibilita verificar

continuamente a experiência do cliente, mesmo quando você não tem nenhum tráfego de cliente em seus aplicativos. Ao usar canaries, é possível descobrir problemas antes que seus clientes o façam.

Canaries são scripts escritos em Node.js ou Python. Eles criam funções do Lambda em sua conta que usam Node.js ou Python como framework. Os canaries trabalham por meio de protocolos HTTP e HTTPS. Os canários usam camadas do Lambda que contêm a biblioteca CloudWatch Synthetics. A biblioteca contém a versão NodeJS do CloudWatch Synthetics para canários NodeJS e a versão Python do CloudWatch Synthetics para canários do Python. As camadas pertencem à conta de serviço do CloudWatch Synthetics. As bibliotecas nunca transmitem ou armazenam informações de clientes. Todos os dados do cliente são armazenados somente na conta do cliente.

Os canaries oferecem acesso programático a um navegador Google Chrome dedicado via Puppeteer ou Selenium Webdriver. Para obter mais informações sobre o Puppeteer, consulte Puppeteer. Para obter mais informações sobre o Selenium, consulte www.selenium.dev/

Os canaries verificam a disponibilidade e a latência dos endpoints e podem armazenar dados de tempo de carregamento e capturas de tela da interface do usuário. Eles monitoram as APIs REST, os URLs e o conteúdo do site e podem verificar se há alterações não autorizadas de phishing, injeção de código e scripts entre sites.

O CloudWatch Synthetics é integrado ao Application Signals, que pode descobrir e monitorar serviços de aplicações, clientes, canários do Synthetics e dependências de serviços. Use o Application Signals para ver uma lista ou um mapa visual dos seus serviços, visualizar métricas de integridade com base nos seus objetivos de nível de serviço (SLOs) e fazer uma busca profunda para ver rastreamentos do X-Ray correlacionados para uma solução de problemas mais detalhada. Para ver seus canários no Application Signals, ative o rastreamento ativo do X-Ray. Os canários são exibidos no Mapa de serviços conectado aos serviços e na página Detalhes do serviço dos serviços que eles chamam.

Para obter uma demonstração dos canaries em vídeo, veja o seguinte:

- Introdução ao Amazon CloudWatch Synthetics
- Demonstração do Amazon CloudWatch Synthetics
- Criar canaries usando o Amazon CloudWatch Synthetics
- Monitoramento visual com o Amazon CloudWatch Synthetics

É possível executar um canário uma vez ou em uma programação regular. Os canaries podem ser executados a cada minuto. É possível usar expressões cron e rate para agendar canaries.

Para obter informações sobre problemas de segurança a serem considerados antes de criar e executar canaries, consulte Considerações de segurança para canaries do Synthetics.

Por padrão, os canaries criam várias métricas do CloudWatch no namespace `CloudWatchSynthetics`. Essas métricas têm `CanaryName` como uma dimensão. Os canaries que usam a função `executeStep()` ou `executeHttpStep()` da biblioteca de funções também têm `StepName` como dimensão. Para obter mais informações sobre a biblioteca de funções de canaries, consulte Funções da biblioteca disponíveis para scripts o canário.

O CloudWatch Synthetics se integra bem ao mapa de rastreamento do X-Ray, que usa o CloudWatch com o AWS X-Ray para fornecer uma visualização completa dos serviços e ajudar a localizar gargalos de performance e a identificar com mais eficiência usuários afetados. Os canários criados com o CloudWatch Synthetics aparecem no mapa de rastreamento. Para obter mais informações, consulte X-Ray Trace Map.

Atualmente, o CloudWatch Synthetics está disponível em todas as regiões comerciais da AWS e regiões do GovCloud.

> **Note**
>
> Na Ásia-Pacífico (Osaka), não há suporte para AWS PrivateLink. Na região Ásia-Pacífico (Jakarta), AWS PrivateLink e X-Ray não têm suporte.

Tópicos

- Editar ou excluir um canário
- Iniciar, interromper, excluir ou atualizar o runtime de vários canários
- Monitorar eventos do canário com o Amazon EventBridge

Funções e permissões obrigatórias para canaries do CloudWatch

Tanto os usuários que criam e gerenciam canaries, quanto os próprios canaries, devem ter certas permissões.

Funções e permissões necessárias para usuários que gerenciam canaries do CloudWatch

Para visualizar detalhes do canário e os resultados de execuções do canário, é necessário estar conectado como um usuário do que tenha as políticas `CloudWatchSyntheticsFullAccess` ou `CloudWatchSyntheticsReadOnlyAccess` anexadas. Para ler todos os dados do Synthetics no console, você também precisa das políticas `AmazonS3ReadOnlyAccess` e `CloudWatchReadOnlyAccess`. Para visualizar o código-fonte usado pelos canaries, também é necessária a política `AWSLambda_ReadOnlyAccess`.

Para criar canários, é necessário estar conectado como um usuário que tenha a política `CloudWatchSyntheticsFullAccess` ou um conjunto semelhante de permissões. Para criar funções do IAM para os canaries, também é necessária a seguinte instrução de política em linha:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
```

```
}
```

> **Important**
>
> Conceder a um usuário as permissões `iam:CreateRole`, `iam:CreatePolicy` e `iam:AttachRolePolicy` concede a esse usuário acesso administrativo total à conta da AWS. Por exemplo, um usuário com essas permissões pode criar uma política com permissões totais para todos os recursos e associar essa política a qualquer função. Seja muito cuidadoso a quem você concede essas permissões.

Para obter informações sobre como anexar políticas e conceder permissões a usuários, consulte Alterar permissões para um usuário do IAM e Incorporar uma política em linha para um usuário ou para uma função.

Funções e permissões necessárias para canaries

Cada canário deve estar associado a uma função do IAM que tenha certas permissões anexadas. Ao criar um canário usando o console do CloudWatch, você pode escolher o CloudWatch Synthetics para criar uma função do IAM para o canário. Se você fizer isso, a função terá as permissões necessárias.

Para criar a função do IAM por conta própria ou criar uma função do IAM que você possa usar ao utilizar AWS CLI ou APIs para criar um canário, a função deve conter as permissões listadas nesta seção.

Todas as funções do IAM para canaries devem incluir a declaração de política de confiança a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Além disso, a função do IAM do canário precisa de uma das declarações a seguir.

O canário básico que não usa o AWS KMS nem precisa de acesso à Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/s3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/s3/bucket/that/contains/canary/results"

            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
```

```
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
               "StringEquals": {
                "cloudwatch:namespace": "CloudWatchSynthetics"
               }
            }
        }
    ]
}
```

O canário que usa o AWS KMS para criptografar artefatos do canário, mas não precisa de acesso à Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource":
 "arn:aws:kms:KMS_key_region_name:KMS_key_account_id:key/KMS_key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "s3.region_name_of_the_canary_results_S3_bucket.amazonaws.com"
                    ]
```

```
                }
            }
        }
    ]
}
```

O canário que não usa o AWS KMS, mas precisa de acesso à Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": [
                "*"
            ]
       }
    ]
}
```

O canário que usa o AWS KMS para criptografar artefatos do canário e também precisa de acesso à Amazon VPC

Se você atualizar um canal não-VPC para começar a usar uma VPC, será necessário atualizar a função do canário para incluir as permissões da interface de rede listadas na política a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
```

```
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
```

```
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": [
                "*"
            ]
       },
       {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource":
 "arn:aws:kms:KMS_key_region_name:KMS_key_account_id:key/KMS_key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "s3.region_name_of_the_canary_results_S3_bucket.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Políticas gerenciadas pela AWS para o CloudWatch Synthetics

Para adicionar permissões a usuários, grupos e perfis, é mais fácil usar políticas gerenciadas pela AWS do que gravar políticas por conta própria. É necessário tempo e experiência para criar políticas gerenciadas pelo cliente do IAM que fornecem à sua equipe apenas as permissões de que precisam. Para começar rapidamente, é possível usar nossas políticas gerenciadas pela AWS. Essas políticas abrangem casos de uso comuns e estão disponíveis na sua conta da AWS. Para

obter mais informações sobre políticas gerenciadas pela AWS, consulte Políticas gerenciadas pela AWS políticas gerenciadas pela AWS no Guia do usuário do IAM.

Os serviços da AWS mantêm e atualizam políticas gerenciadas pela AWS. Não é possível alterar as permissões em políticas gerenciadas pela AWS. Os serviços ocasionalmente alteram as permissões em uma política gerenciada pela AWS. Esse tipo de atualização afeta todas as identidades (usuários, grupos e perfis) em que a política está anexada.

Atualização do CloudWatch Synthetics para políticas gerenciadas pela AWS

Veja detalhes sobre atualizações em políticas gerenciadas pela AWS para o CloudWatch Synthetics desde que esse serviço começou a monitorar essas alterações. Para obter alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed RSS na página Document history (Histórico de documentos) do CloudWatch.

Alteração	Descrição	Data
Ações redundantes removidas de CloudWatchSyntheticsFullAccess	O CloudWatch Synthetics removeu as ações `s3:PutBucketEncryption` e `lambda:GetLayerVersionByArn` da política CloudWatchSyntheticsFullAccess porque essas ações eram redundantes com outras permissões da política. As ações removidas não forneciam permissões e não há nenhuma alteração de rede nas permissões concedidas pela política.	12 de março de 2021
O CloudWatch Synthetics começou a monitorar alterações	O CloudWatch Synthetics começou a rastrear alterações para as políticas gerenciadas pela AWS.	10 de março de 2021

CloudWatchSyntheticsFullAccess

Veja a seguir o conteúdo da política `CloudWatchSyntheticsFullAccess`:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "synthetics:*"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:CreateBucket",
            "s3:PutEncryptionConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::cw-syn-results-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles",
            "s3:ListAllMyBuckets",
            "s3:GetBucketLocation",
            "xray:GetTraceSummaries",
            "xray:BatchGetTraces",
            "apigateway:GET"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource":"arn:aws:s3:::cw-syn-*"
      },
      {
         "Effect":"Allow",
```

```
            "Action":[
               "s3:GetObjectVersion"
            ],
            "Resource":"arn:aws:s3:::aws-synthetics-library-*"
         },
         {
            "Effect":"Allow",
            "Action":[
               "iam:PassRole"
            ],
            "Resource":[
               "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ],
            "Condition":{
               "StringEquals":{
                  "iam:PassedToService":[
                     "lambda.amazonaws.com",
                     "synthetics.amazonaws.com"
                  ]
               }
            }
         },
         {
            "Effect":"Allow",
            "Action":[
               "iam:GetRole"
            ],
            "Resource":[
               "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ]
         },
         {
            "Effect":"Allow",
            "Action":[
               "cloudwatch:GetMetricData",
               "cloudwatch:GetMetricStatistics"
            ],
            "Resource":"*"
         },
         {
            "Effect":"Allow",
            "Action":[
               "cloudwatch:PutMetricAlarm",
               "cloudwatch:DeleteAlarms"
```

```
         ],
         "Resource":[
            "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "cloudwatch:DescribeAlarms"
         ],
         "Resource":[
            "arn:aws:cloudwatch:*:*:alarm:*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "lambda:CreateFunction",
            "lambda:AddPermission",
            "lambda:PublishVersion",
            "lambda:UpdateFunctionConfiguration",
            "lambda:GetFunctionConfiguration"
         ],
         "Resource":[
            "arn:aws:lambda:*:*:function:cwsyn-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "lambda:GetLayerVersion",
            "lambda:PublishLayerVersion"
         ],
         "Resource":[
            "arn:aws:lambda:*:*:layer:cwsyn-*",
            "arn:aws:lambda:*:*:layer:Synthetics:*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
```

```
         ],
         "Resource":[
            "*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sns:ListTopics"
         ],
         "Resource":[
            "*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sns:CreateTopic",
            "sns:Subscribe",
            "sns:ListSubscriptionsByTopic"
         ],
         "Resource":[
            "arn:*:sns:*:*:Synthetics-*"
         ]
      }
   ]
}
```

CloudWatchSyntheticsReadOnlyAccess

Veja a seguir o conteúdo da política `CloudWatchSyntheticsReadOnlyAccess`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

Limitar um usuário a visualizar canaries específicos

Você pode limitar a capacidade de um usuário visualizar informações sobre canaries, para que só possa ver informações sobre os canaries especificados. Para fazer isso, use uma política do IAM com uma instrução `Condition`, semelhante à que se segue, e anexe essa política a um usuário ou a um perfil do IAM.

O exemplo a seguir limita o usuário a visualizar apenas informações sobre `name-of-allowed-canary-1` e `name-of-allowed-canary-2`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "synthetics:DescribeCanaries",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "synthetics:Names": [
                        "name-of-allowed-canary-1",
                        "name-of-allowed-canary-2"
                    ]
                }
            }
        }
    ]
}
```

O CloudWatch Synthetics é compatível com a listagem de até cinco itens na matriz `synthetics:Names`.

Você também pode criar uma política que use um * como curinga nos nomes do canário que devem ser permitidos, conforme mostrado no exemplo a seguir:

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "synthetics:DescribeCanaries",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringLike": {
                    "synthetics:Names": [
                        "my-team-canary-*"
                    ]
                }
            }
        }
    ]
}
```

Um usuário que fez login com uma dessas políticas anexadas não pode usar o console do CloudWatch para exibir informações sobre canaries. Os usuários só podem visualizar informações sobre os canaries autorizados pela política e apenas usando a API DescribeCanaries ou o comando describe-canaries da AWS CLI.

Criar um canário

> **Important**
>
> Use canaries do Synthetics para monitorar somente endpoints e APIs nos quais você tem propriedade ou permissões. Dependendo das configurações de frequência do canário, esses endpoints podem sofrer aumento do tráfego.

Ao usar o console do CloudWatch para criar um canário, é possível usar um esquema fornecido pelo CloudWatch para criar o canário ou escrever seu próprio script. Para ter mais informações, consulte Usar esquemas de canaries.

Também é possível criar um canay usando o AWS CloudFormation, se você estiver usando seu próprio script para o canário. Para obter mais informações, consulte AWS::Synthetics::Canary no Manual do usuário doAWS CloudFormation.

Se você estiver escrevendo seu próprio script, poderá usar várias funções que o CloudWatch Synthetics incorporou a uma biblioteca. Para ter mais informações, consulte Versões do runtime do Synthetics.

> **Note**
>
> Ao criar um canário, uma das camadas criadas é a Synthetics prefixada com `Synthetics`. Essa camada pertence à conta de serviço Synthetics e contém o código de runtime.

Como criar um canário

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals, Canários do Synthetics.
3. Selecione Create Canary (Criar canário).
4. Escolha uma das seguintes opções:
   - Para basear o canário no script de um esquema, escolha Usar um blueprint (Usar um esquema) e selecione o tipo de canário que deseja criar. Para obter mais informações sobre o que faz cada tipo de esquema, consulte Usar esquemas de canaries.
   - Para fazer upload de seu próprio script Node.js a fim de criar um canário personalizado, selecione Upload a script (Fazer upload de um script).

     Depois, é possível arrastar o script para a área Script ou selecionar Browse files (Pesquisar arquivos) para navegar até o script no sistema de arquivos.
   - Para importar o script de um bucket do S3, selecione Import from S3 (Importar do S3). Em Source location (Local de origem), digite o caminho completo do canário ou selecione Browse S3 (Pesquisar no S3).

     É necessário ter as permissões `s3:GetObject` e `s3:GetObjectVersion` para o bucket do S3 usado. O bucket deve estar na mesma região da AWS onde o canário está sendo criado.
5. Em Name (Nome), insira um nome para o canário. O nome é usado em várias páginas, portanto, recomendamos que você forneça a ele um nome descritivo para distingui-lo de outros canaries.
6. Em Application or endpoint URL (URL do aplicativo ou endpoint), digite o URL que você deseja que o canário teste. Esse URL deve incluir o protocolo (como https://).

   Se você quiser que o canário teste um endpoint em uma VPC, insira as informações sobre a VPC posteriormente neste procedimento.
7. Se estiver usando seu próprio script para o canário, em Lambda handler (Manipulador do Lambda), insira o ponto de entrada onde deseja que o canário se inicie. Se você usar um

runtime anterior a `syn-nodejs-puppeteer-3.4` ou `syn-python-selenium-1.1`, a string inserida deverá terminar com `.handler`. Se você usar `syn-nodejs-puppeteer-3.4`, `syn-python-selenium-1.1` ou um runtime posterior, essa restrição não se aplicará.

8. Se você estiver usando variáveis de ambiente em seu script, escolha Environment variables (Variáveis de ambiente) e especifique um valor para cada variável de ambiente definida no script. Para ter mais informações, consulte Variáveis de ambiente.
9. Em Schedule (Programar), escolha se deseja executar esse canário uma vez, continuamente usando uma expressão rate ou usando uma expressão cron.

   - Ao usar o console do CloudWatch para criar um canário que é executado continuamente, você pode escolher uma taxa em qualquer lugar entre uma vez por minuto e uma vez por hora.

   - Para obter mais informações sobre como escrever uma expressão cron para programação do canário, consulte Agendamento de execuções do canário usando cron.
10. (Opcional) Para definir um valor de tempo limite para o canário, escolha Additional configuration (Configuração adicional) e, depois, especifique o valor do tempo limite. Especifique um tempo limite de pelo menos 15 segundos para permitir que o Lambda seja iniciado a frio e que a instrumentação do canário seja inicializada.
11. Em Data retention (Retenção de dados), especifique por quanto tempo as informações sobre as execuções bem-sucedidas e com falha de canaries devem ser mantidas. O intervalo é de 1 a 455 dias.

    Essa configuração afeta apenas os dados que são armazenados e exibidos no console pelo CloudWatch Synthetics. Ela não tem efeito sobre os dados armazenados nos seus buckets do Amazon S3, nem em logs ou métricas que são publicados pelo canário.
12. Em Data Storage (Armazenamento de dados), selecione o bucket do S3 a ser usado para armazenar os dados das execuções do canário. O nome do bucket não pode conter um ponto (.). Se deixar essa opção em branco, um bucket padrão do S3 será usado ou criado.

    Se você estiver usando o runtime `syn-nodejs-puppeteer-3.0` ou posterior, ao inserir a URL do bucket na caixa de texto, poderá especificar um bucket na região atual ou em uma região diferente. Caso esteja usando uma versão de runtime anterior, o bucket deverá estar na região atual.
13. (Opcional) Por padrão, os canaries armazenam seus artefatos no Amazon S3 e os artefatos são criptografados em repouso usando uma chave do AWS KMS gerenciada pela AWS. É possível usar uma opção de criptografia diferente escolhendo Additional configuration (Configuração

adicional) na seção Data Storage (Armazenamento de dados). Em seguida, escolha o tipo de chave a ser usada para a criptografia. Para ter mais informações, consulte Criptografar artefatos do canário.

14. Em Access permissions (Permissões de acesso), escolha se deseja criar uma função do IAM para executar o canário ou usar uma existente.

    Se você usar o CloudWatch Synthetics para criar a função, ele incluirá automaticamente todas as permissões necessárias. Se desejar criar a função por conta própria, consulte Funções e permissões necessárias para canaries para obter informações sobre as permissões necessárias.

    Se o console do CloudWatch for usado para criar uma função para um canário ao criá-lo, não será possível reutilizar a função para outros canários, porque essas funções são específicas de apenas um canário. Se você tiver criado manualmente uma função que funcione para vários canaries, poderá usá-la.

    Para usar uma função existente, é necessário ter a permissão `iam:PassRole` para transmitir essa função para o Synthetics e o Lambda. Também é necessário ter a permissão `iam:GetRole`.

15. (Opcional) Em Alarms (Alarmes), escolha se deseja que sejam criados alarmes padrão do CloudWatch para esse canário. Se optar por criar alarmes, eles serão criados com esta convenção de nomes: `Synthetics-Alarm-`*`canaryName`*`-`*`index`*

    `index` é um número que representa cada alarme diferente criado para este canário. O primeiro alarme tem um índice de 1, o segundo alarme tem um índice de 2 e assim por diante.

16. (Opcional) Para que esse canário teste um endpoint em uma VPC, escolha VPC settings (Configurações da VPC) e faça o seguinte:

    a. Selecione a VPC que hospeda o endpoint.

    b. Selecione uma ou mais sub-redes na sua VPC. É necessário selecionar uma sub-rede privada, porque a instância do Lambda não pode ser configurada para ser executada em uma sub-rede pública quando um endereço IP não pode ser atribuído à instância do Lambda durante a execução. Para obter mais informações, consulte Configurar uma função Lambda para acessar recursos em uma VPC.

    c. Selecione um ou mais grupos de segurança na sua VPC.

Se o endpoint estiver em uma VPC, habilite seu canário para enviar informações ao CloudWatch e ao Amazon S3. Para ter mais informações, consulte Execução de um canário em uma VPC.

17. (Opcional) Em Tags, adicione um ou mais pares de chave/valor como tags para esse canário. As tags podem ajudar a identificar e organizar seus recursos da AWS e acompanhar seus custos da AWS. Para ter mais informações, consulte Etiquetar recursos do Amazon CloudWatch.
18. (Opcional) Em Active tracing (Rastreamento ativo), escolha se deseja ativar o rastreamento ativo do X-Ray para esse canário. Essa opção só estará disponível se o canário usar a versão do runtime syn-nodejs-2.0 ou posterior. Para ter mais informações, consulte Canaries e rastreamento do X-Ray.

Recursos que são criados para canaries

Ao criar um canário, os seguintes recursos são criados:

- Uma função do IAM com o nome `CloudWatchSyntheticsRole-`*`canary-name`*`-`*`uuid`* (se você usar o console do CloudWatch para criar o canário e especificar que deve ser criada uma função para ele)
- Uma política do IAM com o nome `CloudWatchSyntheticsPolicy-`*`canary-name`*`-`*`uuid`*.
- Um bucket do S3 com o nome `cw-syn-results-`*`accountID`*`-`*`region`*.
- Alarmes com o nome `Synthetics-Alarm-`*`MyCanaryName`*, se você desejar que alarmes sejam criados para o canário.
- Camadas e funções do Lambda, caso você use um esquema para criar o canário. Esses recursos têm o prefixo `cwsyn-`*`MyCanaryName`*.
- Grupos de logs do CloudWatch Logs com o nome `/aws/lambda/cwsyn-`*`MyCanaryName`*`-`*`randomId`*.

Usar esquemas de canaries

Esta seção fornece detalhes sobre cada um dos esquemas de canaries e as tarefas para as quais cada um é mais adequado. Os esquemas são fornecidos pelos seguintes tipos de canaries:

- Monitor de pulsação
- Canário da API
- Verificador de links quebrados

- Monitoramento visual
- Gravador do canário
- Fluxo de trabalho da GUI

Ao usar um esquema para criar um canário, conforme você preenche os campos no console do CloudWatch, a área Script editor (Editor de scripts) da página exibe o canário que você está criando como um script Node.js. Também é possível editar o canário nessa área para personalizá-lo ainda mais.

Monitorar pulsação

Os scripts de pulsação carregam a URL especificada e armazenam uma captura de tela da página e um arquivo HTTP (arquivo HAR). Eles também armazenam logs de URLs acessados.

É possível usar os arquivos HAR para visualizar dados de performance detalhados sobre as páginas da web. Você pode analisar a lista de solicitações da web e detectar problemas de performance, como tempo de carregamento de um item.

Se o canário usar a versão de runtime `syn-nodejs-puppeteer-3.1` ou posterior, você poderá usar o esquema de monitoramento de pulsação para monitorar várias URLs e ver o status, a duração, as capturas de telas associadas e o motivo da falha de cada URL no resumo de etapas do relatório de execução do canário.

Canário da API

Canaries de API podem testar as funções básicas de leitura e gravação de uma API REST. REST significa representational state transfer (transferência de estado representacional) e é um conjunto de regras que os desenvolvedores seguem ao criar uma API. Uma dessas regras determina que um link para um URL específico deve retornar uma parte dos dados.

O canaries podem trabalhar com qualquer APIs e testar todos os tipos de funcionalidade. Cada canário pode fazer várias chamadas de API.

Em canários que usam a versão de runtime `syn-nodejs-2.2` ou posterior, o esquema do canário da API é compatível com canários de várias etapas que monitoram suas APIs como etapas HTTP. É possível testar várias APIs em um único canário. Cada etapa é uma solicitação separada que pode acessar uma URL diferente, usar cabeçalhos diferentes e usar regras diferentes para definir se os cabeçalhos e os corpos das respostas serão capturados. Não capturarando cabeçalhos e corpo de resposta, você pode impedir que dados sigilosos sejam registrados.

Cada solicitação de um canário da API consiste nas seguintes informações:

- O endpoint, que é o URL solicitado.
- O método, que é o tipo da solicitação enviada para o servidor. APIs REST oferecem suporte a operações GET (leitura), POST (gravação), PUT (atualização), PATCH (atualização) e DELETE (exclusão).
- Os cabeçalhos, que fornecem informações para o cliente e o servidor. Eles são usados para autenticação e para fornecer informações sobre o conteúdo do corpo. Para obter uma lista de cabeçalhos válidos, consulte Cabeçalhos HTTP.
- Os dados (ou o corpo) que contêm informações a serem enviadas para o servidor. Isso é usado somente para solicitações POST, PUT, PATCH ou DELETE.

O esquema do canário de API é compatível com os métodos GET e POST. Ao usar esse esquema, é necessário especificar cabeçalhos. Por exemplo, você pode especificar **`Authorization`** como uma Key (Chave) e especificar os dados de autorização necessários como o Value (Valor) para essa chave.

Se você estiver testando uma solicitação POST, especifique também o conteúdo a ser publicado no campo Data (Dados).

Integração com o API Gateway

O esquema de API é integrado ao Amazon API Gateway. Isso permite selecionar uma API do API Gateway e um estágio da mesma conta e região da AWS como o canário ou carregar um modelo do Swagger do API Gateway para monitoramento de API entre contas e regiões. Então é possível escolher os detalhes restantes no console para criar o canário, em vez de inseri-los do zero. Para obter mais informações sobre o API Gateway, consulte O que é o Amazon API Gateway?

Usar uma API privada

Você pode criar um canário que use uma API privada no Amazon API Gateway. Para obter mais informações, consulte Como criar uma API privada no Amazon API Gateway?

Verificador de link quebrado

O verificador de links quebrados coleta todos os links dentro da URL que você está testando usando `document.getElementsByTagName('a')`. Ele testa apenas até o número de links especificado, e a URL em si é considerada o primeiro link. Por exemplo, se você quiser verificar todos os links em uma página que contenha cinco links, deverá especificar para o canário seguir seis links.

Os canários que verificam links quebrados criados usando o runtime `syn-nodejs-2.0-beta` ou posterior oferecem suporte aos seguintes recursos adicionais:

- Fornece um relatório contendo os links verificados, o código de status, o motivo da falha (se houver) e as capturas de tela da página de origem e de destino.
- Ao visualizar os resultados do canário, é possível filtrar para ver apenas os links quebrados e corrigir o link de acordo com o motivo da falha.
- Essa versão obtém capturas de tela da página de origem anotada para cada link e destaca a âncora onde o link foi encontrado. Os componentes ocultos não são anotados.
- É possível configurar essa versão para obter capturas de tela de páginas de origem e de destino, apenas páginas de origem ou apenas páginas de destino.
- Essa versão corrige um problema na versão anterior em que o script do canário é interrompido após o primeiro link quebrado mesmo quando mais links são extraídos da primeira página.

Para atualizar um canário existente com `syn-1.0` para usar o novo runtime, é necessário excluir e recriar o canário. Atualizar um canário existente para o novo runtime não disponibiliza esses recursos.

Um canário do verificador de links quebrados detecta os seguintes tipos de erros de link:

- 404 Page Not Found (404 Página não encontrada)
- Invalid Host Name (Nome de host inválido)
- Bad URL (URL incorreto). Por exemplo, a URL não contém um colchete, tem barras extras ou usa um protocolo incorreto.
- Invalid HTTP response code (Código de resposta HTTP inválido).
- O servidor host gera respostas vazias sem conteúdo e sem código de resposta.
- As solicitações HTTP constantemente atingem o tempo limite durante a execução do canário.
- O host elimina conexões consistentemente porque ele está configurado incorretamente ou está muito ocupado.

Esquema de monitoramento visual

O esquema de monitoramento visual inclui código para comparar capturas de tela feitas durante uma execução do canário com capturas de tela feitas durante uma execução do canário de linha de base. Se a discrepância entre as duas capturas de tela estiver além de uma porcentagem limite, o canário

falhará. O monitoramento visual é compatível com canaries que executam syn-puppeteer-node-3.2 e posterior. Atualmente não é compatível com canaries que executam Python e Selenium.

O esquema de monitoramento visual inclui a seguinte linha de código no script do canário do esquema padrão, que permite o monitoramento visual.

```
syntheticsConfiguration.withVisualCompareWithBaseRun(true);
```

A primeira vez que o canário é executado corretamente após essa linha ser adicionada ao script, ele usa as capturas de tela obtidas durante a execução como linha de base para comparação. Após a primeira execução do canário, é possível usar o console do CloudWatch para editar o canário para fazer qualquer um destes procedimentos:

- Defina a próxima execução do canário como a nova linha de base.
- Estabeleça limites na captura de tela de linha de base atual para designar as áreas da captura de tela que deverão ser ignoradas durante comparações visuais.
- Remova uma captura de tela que não será usada para monitoramento visual.

Para obter mais informações sobre como usar o console do CloudWatch para editar um canário, consulte Editar ou excluir um canário.

Também é possível alterar a execução do canário que é usada como linha de base usando os parâmetros `nextrun` ou `lastrun` ou especificando um ID de execução do canário na API UpdateCanary.

Ao usar o esquema de monitoramento visual, insira a URL onde deseja que a captura de tela seja feita e especifique um limite de diferença em porcentagem. Após a execução da linha de base, as execuções futuras do canário que detectam uma diferença visual maior do que esse limite desencadeiam uma falha do canário. Após a execução da linha de base, também é possível editar o canário para “traçar” limites na captura de tela da linha de base que deseja ignorar durante o monitoramento visual.

O recurso de monitoramento visual é desenvolvido pelo toolkit de software de código aberto ImageMagick. Para obter mais informações, consulte ImageMagick.

Gravador do canário

Com o esquema do gravador do canário, é possível usar o CloudWatch Synthetics Recorder para registrar suas ações de clicar e digitar em um site e gerar automaticamente um script Node.js que

pode ser usado para criar um canário que segue as mesmas etapas. O CloudWatch Synthetics Recorder é uma extensão do Google Chrome fornecida pela Amazon.

Créditos: o CloudWatch Synthetics Recorder é baseado no Headless recorder .

Para ter mais informações, consulte Usar o CloudWatch Synthetics Recorder para Google Chrome.

Criador de fluxos de trabalho da GUI

O esquema criador de fluxos de trabalho da GUI verifica se as ações podem ser executadas em sua página da Web. Por exemplo, se você tiver uma página da Web com um formulário de login, o canário poderá preencher os campos de usuário e senha e enviá-lo para verificar se a página da Web está funcionando corretamente.

Ao usar um esquema para criar esse tipo de canário, especifique as ações a serem executadas pelo canário na página da Web. As ações que podem ser utilizadas são as seguintes:

- Clicar: seleciona o elemento especificado e simula um usuário clicando ou escolhendo o elemento.

  Para especificar o elemento em um script Node.js, use `[id=]` ou `a[class=]`.

  Para especificar o elemento em um script Python .js, use `xpath //*[@id=]` ou `//*[@class=]`.
- Verificar seletor: verifica se o elemento especificado existe na página da Web. Esse teste é útil para verificar se uma ação anterior faz com que os elementos corretos preencham a página.

  Para especificar o elemento a ser verificado em um script Node.js, use `[id=]` ou `a[class=]`.

  Para especificar o elemento a ser verificado em um script Python .js, use `xpath //*[@id=]` ou `//*[class=]`.
- Verificar texto: verifica se a string especificada está contida no elemento de destino. Esse teste é útil para verificar se uma ação anterior fez o texto correto ser exibido.

  Para especificar o elemento em um script do Node.js, use um formato como `div[@id=]//h1` porque essa ação usa a função `waitForXPath` no Puppeteer.

  Para especificar o elemento em um script Python, use o formato xpath como `//*[@id=]` ou //*[@class=] porque esta ação usa a função `implicitly_wait` no Selenium.
- Texto de entrada: grava o texto especificado no elemento de destino.

  Para especificar o elemento a ser verificado em um script Node.js, use `[id=]` ou `a[class=]`.

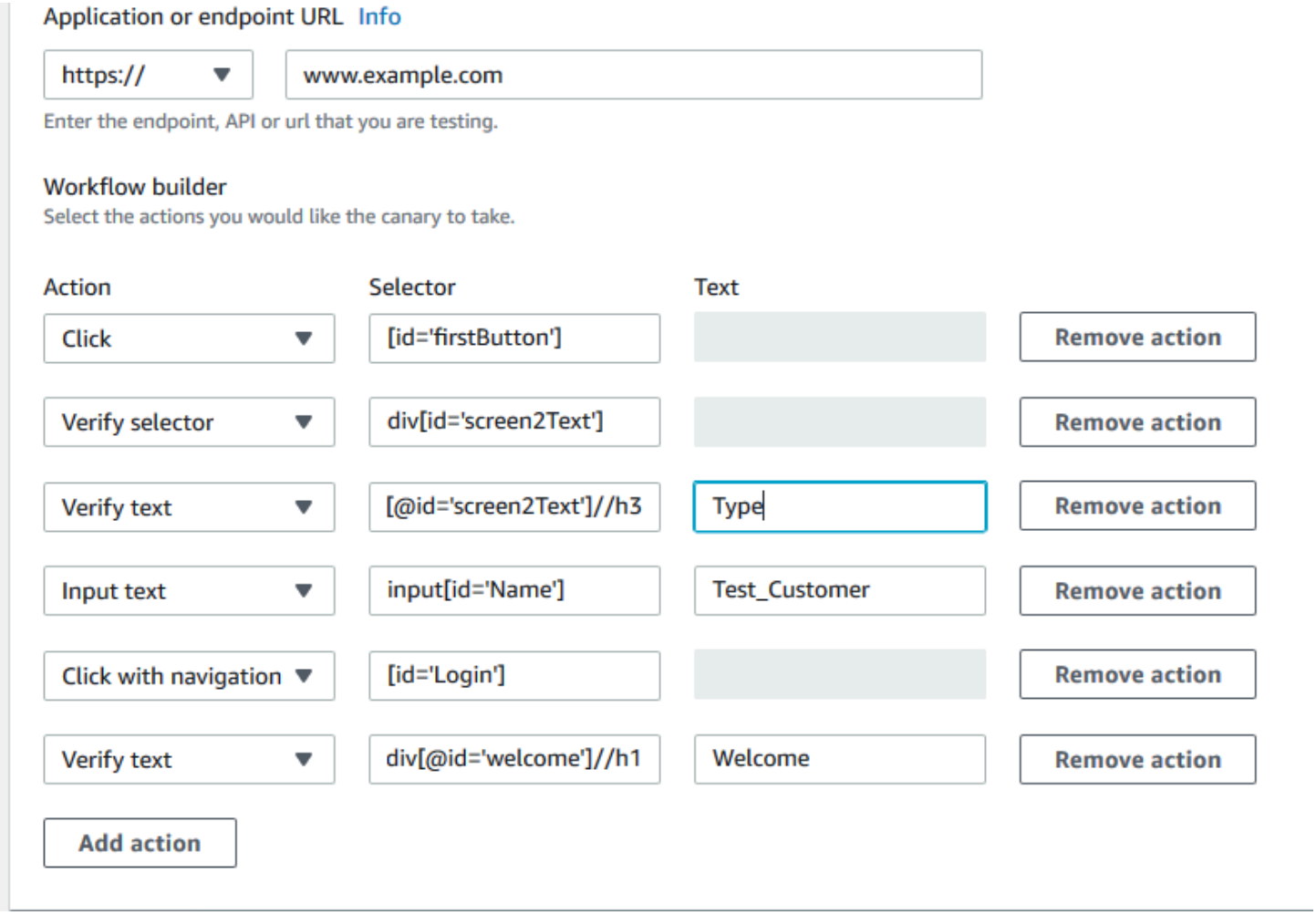
Para especificar o elemento a ser verificado em um script Python .js, use `xpath //*[@id=]` ou `//*[@class=]`.

- Clicar com a navegação: aguarda a página inteira ser carregada depois de escolher o elemento especificado. Isso é mais útil quando é necessário recarregar a página.

  Para especificar o elemento em um script Node.js, use `[id=]` ou `a[class=]`.

  Para especificar o elemento em um script Python .js, use `xpath //*[@id=]` ou `//*[@class=]`.

Por exemplo, o esquema a seguir usa Node.js. Ele clica no firstButton na URL especificada, verifica se o seletor esperado com o texto esperado é exibido, insere o nome `Test_Customer` no campo Name (Nome), clica no botão Login e confere se o login teve êxito verificando o texto de boas-vindas na página seguinte.



Os canários de fluxo de trabalho GUI que usam os tempos de execução a seguir também fornecem um resumo das etapas executadas para cada execução do canário. É possível usar as capturas de tela e a mensagem de erro associadas a cada etapa para encontrar a causa raiz da falha.

- `syn-nodejs-2.0` ou posterior
- `syn-python-selenium-1.0` ou posterior

Usar o CloudWatch Synthetics Recorder para Google Chrome

A Amazon fornece um CloudWatch Synthetics Recorder para ajudar você a criar canaries com mais facilidade. O gravador é uma extensão do Google Chrome.

O gravador registra suas ações de clicar e digitar em um site e gera automaticamente um script Node.js que pode ser usado para criar um canário que segue as mesmas etapas.

Depois de iniciar a gravação, o CloudWatch Synthetics Recorder detecta suas ações no navegador e as converte em script. É possível pausar e retomar a gravação conforme necessário. Quando você interrompe a gravação, o gravador produz um script Node.js de suas ações, que pode facilmente ser copiado com o botão Copy to Clipboard (Copiar para a área de transferência). Em seguida, é ossível usar esse script para criar um canário no CloudWatch Synthetics.

Créditos: o CloudWatch Synthetics Recorder é baseado no Headless recorder .

Instalar a extensão do CloudWatch Synthetics Recorder para Google Chrome

Para usar o CloudWatch Synthetics Recorder, você pode começar a criar um canário e escolher o esquema Canary Recorder (Gravador do canário). Se você fizer isso quando ainda não tiver baixado o gravador, o console do CloudWatch Synthetics fornecerá um link para baixá-lo.

Se preferir, você pode seguir estas etapas para baixar e instalar o gravador diretamente.

Para instalar o CloudWatch Synthetics Recorder

1. Usando o Google Chrome, acesse este site: https://chrome.google.com/webstore/detail/cloudwatch-synthetics-rec/bhdnlmmgiplmbcdmkkdfplenecpegfno
2. Selecione Add to Chrome (Adicionar ao Chrome) e escolhaAdd extension (Adicionar extensão).

Usar o CloudWatch Synthetics Recorder para Google Chrome

Para usar o CloudWatch Synthetics Recorder para facilitar a criação de um canário, escolha Create canary (Criar canário) no console do CloudWatch e escolha Use a blueprint (Usar um esquema), Canary Recorder (Gravador do canário). Para ter mais informações, consulte Criar um canário.

Se preferor, você pode usar o gravador para gravar etapas sem usá-las imediatamente para criar um canário.

Para usar o CloudWatch Synthetics Recorder para registrar suas ações em um site

1. Navegue até a página que você deseja monitorar.
2. Escolha o ícone de extensões do Chrome e escolha CloudWatch Synthetics Recorder.
3. Escolha Start Recording (Iniciar gravação).
4. Execute as etapas que você deseja registrar. Para pausar a gravação, escolha Pause.
5. Quando terminar de gravar o fluxo de trabalho, selecione Stop recording (Interromper a gravação).
6. Selecione Copy to clipboard (Copiar para a área de transferência) para copiar o script gerado para a área de transferência. Ou, se quiser recomeçar, escolha New recording (Nova gravação).
7. Para criar um canário com o script copiado, é possível colar seu script copiado no editor embutido do esquema do gravador ou salvá-lo em um bucket do Amazon S3 e importá-lo de lá.
8. Se não criar um canário imediatamente, você poderá salvar seu script gravado em um arquivo.

Limitações conhecidas do CloudWatch Synthetics Recorder

Os CloudWatch Synthetics Recorder atualmente para Google Chrome apresenta as limitações a seguir.

- Elementos HTML que não têm IDs usarão seletores CSS. Isso pode quebrar canaries, se a estrutura da página da Web for alterada posteriormente. Planejamos fornecer algumas opções de configuração (como usar data-id) sobre isso em uma versão futura do gravador.
- O gravador não oferece suporte a ações como clique duplo ou copiar/colar e não oferece suporte a combinações de teclas como CMD+0.
- Para verificar a presença de um elemento ou texto na página, os usuários deverão adicionar asserções após o script ser gerado. O gravador não é compatível com a verificação de um elemento sem executar qualquer ação nesse elemento. Isso é semelhante às opções “Verify

text” (“Verificar texto”) ou “Verify element” (“Verificar elemento”) no criador de fluxo de trabalho do canário. Pretendemos adicionar algumas afirmações de suporte em uma versão futura do gravador.

- O gravador registra todas as ações na guia onde a gravação é iniciada. Não registra pop-ups (por exemplo, para permitir o rastreamento de localização) ou navegação para páginas diferentes de pop-ups.

Versões do runtime do Synthetics

Quando você cria ou atualiza um canário, você escolhe uma versão de runtime Synthetics para o canário. Um runtime do Synthetics é uma combinação do código Synthetics que chama seu manipulador de scripts e as camadas do Lambda de dependências agrupadas.

Atualmente, o CloudWatch Synthetics oferece suporte a tempos de execução que usam Node.js para scripts e o framework do Puppeteer, além de tempos de execução que usam Python para desenvolvimento de scripts e Selenium Webdriver para o framework.

Recomendamos usar sempre a versão de runtime mais recente para seus canaries, para poder utilizar os últimos recursos e atualizações feitas na biblioteca Synthetics.

Ao criar um canário, uma das camadas criadas é a Synthetics prefixada com `Synthetics`. Essa camada pertence à conta de serviço Synthetics e contém o código de runtime.

> **Note**
>
> Sempre que você atualizar um canário para usar uma nova versão do runtime do Synthetics, todas as funções da biblioteca Synthetics que seu canário usar também serão automaticamente atualizadas para a mesma versão do NodeJS que o runtime do Synthetics ofereça suporte.

Tópicos

- Política de suporte ao runtime do CloudWatch Synthetics
- Versões de runtime que usam Node.js e Tuppeteer
- Versões de runtime usando Python e Selenium Webdriver

Política de suporte ao runtime do CloudWatch Synthetics

As versões de runtime do Synthetics estão sujeitas a atualizações de manutenção e segurança. Quando qualquer componente de uma versão de runtime não for mais compatível, essa versão de runtime do Synthetics será defasada.

Não é possível criar canários usando versões de runtime defasadas. Canaries que usam tempos de execução defasados continuam funcionando. Você pode parar, iniciar e apagar esses canaries. Você pode atualizar um canário existente que usa versões de runtime defasadas atualizando o canário para usar uma versão de runtime com suporte.

O CloudWatch Synthetics enviará uma notificação por e-mail, caso você tenha canaries que usam tempos de execução programados para serem defasados nos próximos 60 dias. Recomendamos migrar seus canaries para uma versão de runtime compatível para se beneficiar dos novos aprimoramentos de funcionalidade, segurança e performance incluídos em versões mais recentes.

Como atualizo um canário para uma nova versão de runtime?

É possível atualizar uma versão de runtime do canário usando o console do CloudWatch, o AWS CloudFormation, a AWS CLI ou o AWS SDK. Ao usar o console do CloudWatch, é possível atualizar até cinco canários ao mesmo tempo selecionando-os na página de lista do canário e escolhendo Ações, Atualizar Runtime.

É possível verificar a atualização clonando primeiro o canário com o console do CloudWatch e atualizando sua versão de runtime. Isso cria outro canário, que é um clone de seu canário original. Depois de verificar seu canário com a nova versão do runtime, é possível atualizar a versão do runtime do canário original e excluir o canário clone.

Você também pode atualizar vários canaries usando um script de atualização. Para ter mais informações, consulte Script de atualização do runtime do canário.

Se você atualizar um canário e ele falhar, consulte Solucionar problemas de um canário.

Datas de descontinuação do runtime

Versão do runtime	Data da defasagem
`syn-nodejs-puppeteer-6.1`	8 de março de 2024

Versão do runtime	Data da defasagem
`syn-nodejs-puppeteer-6.0`	8 de março de 2024
`syn-nodejs-puppeteer-5.1`	8 de março de 2024
`syn-nodejs-puppeteer-5.0`	8 de março de 2024
`syn-nodejs-puppeteer-4.0`	8 de março de 2024
`syn-nodejs-puppeteer-3.9`	8 de janeiro de 2024
`syn-nodejs-puppeteer-3.8`	8 de janeiro de 2024
`syn-python-selenium-2.0`	8 de março de 2024
`syn-python-selenium-1.3`	8 de março de 2024
`syn-python-selenium-1.2`	8 de março de 2024
`syn-python-selenium-1.1`	8 de março de 2024
`syn-python-selenium-1.0`	8 de março de 2024
`syn-nodejs-puppeteer-3.7`	8 de janeiro de 2024
`syn-nodejs-puppeteer-3.6`	8 de janeiro de 2024

Versão do runtime	Data da defasagem
`syn-nodejs-puppeteer-3.5`	8 de janeiro de 2024
`syn-nodejs-puppeteer-3.4`	13 de novembro de 2022
`syn-nodejs-puppeteer-3.3`	13 de novembro de 2022
`syn-nodejs-puppeteer-3.2`	13 de novembro de 2022
`syn-nodejs-puppeteer-3.1`	13 de novembro de 2022
`syn-nodejs-puppeteer-3.0`	13 de novembro de 2022
`syn-nodejs-2.2`	28 de maio de 2021
`syn-nodejs-2.1`	28 de maio de 2021
`syn-nodejs-2.0`	28 de maio de 2021
`syn-nodejs-2.0-beta`	8 de fevereiro de 2021
`syn-1.0`	28 de maio de 2021

Script de atualização do runtime do canário

Para atualizar um script do canário para uma versão de runtime compatível, use o script a seguir.

```
const AWS = require('aws-sdk');

// You need to configure your AWS credentials and Region.
//   https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/setting-
credentials-node.html
```

```
//   https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/setting-
region.html

const synthetics = new AWS.Synthetics();

const DEFAULT_OPTIONS = {
  /**
   * The number of canaries to upgrade during a single run of this script.
   */
  count: 10,
  /**
   * No canaries are upgraded unless force is specified.
   */
  force: false
};

/**
 * The number of milliseconds to sleep between GetCanary calls when
 * verifying that an update succeeded.
 */
const SLEEP_TIME = 5000;

(async () => {
  try {
    const options = getOptions();

    const versions = await getRuntimeVersions();
    const canaries = await getAllCanaries();
    const upgrades = canaries
      .filter(canary => !versions.isLatestVersion(canary.RuntimeVersion))
      .map(canary => {
        return {
          Name: canary.Name,
          FromVersion: canary.RuntimeVersion,
          ToVersion: versions.getLatestVersion(canary.RuntimeVersion)
        };
      });

    if (options.force) {
      const promises = [];

      for (const upgrade of upgrades.slice(0, options.count)) {
        const promise = upgradeCanary(upgrade);
        promises.push(promise);
```

```
        // Sleep for 100 milliseconds to avoid throttling.
        await usleep(100);
      }

      const succeeded = [];
      const failed = [];
      for (let i = 0; i < upgrades.slice(0, options.count).length; i++) {
        const upgrade = upgrades[i];
        const promise = promises[i];
        try {
          await promise;
          console.log(`The update of ${upgrade.Name} succeeded.`);
          succeeded.push(upgrade.Name);
        } catch (e) {
          console.log(`The update of ${upgrade.Name} failed with error: ${e}`);
          failed.push({
            Name: upgrade.Name,
            Reason: e
          });
        }
      }

      if (succeeded.length) {
        console.group('The following canaries were upgraded successfully.');
        for (const name of succeeded) {
          console.log(name);
        }
        console.groupEnd()
      } else {
        console.log('No canaries were upgraded successfully.');
      }

      if (failed.length) {
        console.group('The following canaries were not upgraded successfully.');
        for (const failure of failed) {
          console.log('\x1b[31m', `${failure.Name}: ${failure.Reason}`, '\x1b[0m');
        }
        console.groupEnd();
      }
    } else {
      console.log('Run with --force [--count <count>] to perform the first <count>
upgrades shown. The default value of <count> is 10.')
      console.table(upgrades);
    }
```

```
  } catch (e) {
    console.error(e);
  }
})();

function getOptions() {
  const force = getFlag('--force', DEFAULT_OPTIONS.force);
  const count = getOption('--count', DEFAULT_OPTIONS.count);
  return { force, count };

  function getFlag(key, defaultValue) {
    return process.argv.includes(key) || defaultValue;
  }
  function getOption(key, defaultValue) {
    const index = process.argv.indexOf(key);
    if (index < 0) {
      return defaultValue;
    }
    const value = process.argv[index + 1];
    if (typeof value === 'undefined' || value.startsWith('-')) {
      throw `The ${key} option requires a value.`;
    }
    return value;
  }
}

function getAllCanaries() {
  return new Promise((resolve, reject) => {
    const canaries = [];

    synthetics.describeCanaries().eachPage((err, data) => {
      if (err) {
        reject(err);
      } else {
        if (data === null) {
          resolve(canaries);
        } else {
          canaries.push(...data.Canaries);
        }
      }
    });
  });
}
```

```
function getRuntimeVersions() {
  return new Promise((resolve, reject) => {
    const jsVersions = [];
    const pythonVersions = [];
    synthetics.describeRuntimeVersions().eachPage((err, data) => {
      if (err) {
        reject(err);
      } else {
        if (data === null) {
          jsVersions.sort((a, b) => a.ReleaseDate - b.ReleaseDate);
          pythonVersions.sort((a, b) => a.ReleaseDate - b.ReleaseDate);
          resolve({
            isLatestVersion(version) {
              const latest = this.getLatestVersion(version);
              return latest === version;
            },
            getLatestVersion(version) {
              if (jsVersions.some(v => v.VersionName === version)) {
                return jsVersions[jsVersions.length - 1].VersionName;
              } else if (pythonVersions.some(v => v.VersionName === version)) {
                return pythonVersions[pythonVersions.length - 1].VersionName;
              } else {
                throw Error(`Unknown version ${version}`);
              }
            }
          });
        } else {
          for (const version of data.RuntimeVersions) {
            if (version.VersionName === 'syn-1.0') {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-nodejs-2.')) {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-nodejs-puppeteer-')) {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-python-selenium-')) {
              pythonVersions.push(version);
            } else {
              throw Error(`Unknown version ${version.VersionName}`);
            }
          }
        }
      }
    });
  });
```

```
}

async function upgradeCanary(upgrade) {
  console.log(`Upgrading canary ${upgrade.Name} from ${upgrade.FromVersion} to
 ${upgrade.ToVersion}`);
  await synthetics.updateCanary({ Name: upgrade.Name, RuntimeVersion:
 upgrade.ToVersion }).promise();
  while (true) {
    await usleep(SLEEP_TIME);
    console.log(`Getting the state of canary ${upgrade.Name}`);
    const response = await synthetics.getCanary({ Name: upgrade.Name }).promise();
    const state = response.Canary.Status.State;
    console.log(`The state of canary ${upgrade.Name} is ${state}`);
    if (state === 'ERROR' || response.Canary.Status.StateReason) {
      throw response.Canary.Status.StateReason;
    }
    if (state !== 'UPDATING') {
      return;
    }
  }
}

function usleep(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}
```

Versões de runtime que usam Node.js e Tuppeteer

A primeira versão de runtime para Node.js e Puppeteer foi nomeada `syn-1.0`. As versões de runtime posteriores têm a convenção de nomenclatura `syn-`*`language`*`-`*`majorversion`*`.`*`minorversion`*. Começando com `syn-nodejs-puppeteer-3.0`, a convenção de nomenclatura é `syn-`*`language`*`-`*`framework`*`-`*`majorversion`*`.`*`minorversion`*

Um sufixo adicional `-beta` mostra que a versão do runtime está atualmente em uma versão de pré-visualização beta.

As versões de runtime com o mesmo número de versão principal são compatíveis com versões anteriores.

Important

As versões de runtime do CloudWatch Synthetics a seguir estão programadas para desativação em 8 de março de 2024.

- `syn-nodejs-puppeteer-6.1`
- `syn-nodejs-puppeteer-6.0`
- `syn-nodejs-puppeteer-5.1`
- `syn-nodejs-puppeteer-5.0`
- `syn-nodejs-puppeteer-4.0`

Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Important

IMPORTANTE: a dependência incluída do AWS SDK para JavaScript v2 será removida e atualizada para uso do AWS SDK para JavaScript v3 em uma versão futura do runtime. Quando isso acontecer, será possível atualizar suas referências de código canário. Como alternativa, é possível continuar referenciando e usando a dependência incluída do AWS SDK para JavaScript v2 adicionando-a como uma dependência ao arquivo zip do código-fonte.

Observações para todas as versões do runtime

Ao usar a versão de runtime `syn-nodejs-puppeteer-3.0`, verifique se seu script do canário é compatível com Node.js 12.x. Se você usar uma versão mais antiga de um runtime `syn-nodejs`, verifique se seu script é compatível com Node.js 10.x.

O código do Lambda em um canário é configurado para ter no máximo 1 GB de memória. Cada execução de um canário expirará após um valor de tempo limite configurado. Se nenhum valor de tempo limite for especificado para um canário, o CloudWatch escolherá um valor de tempo limite com base na frequência do canário. Se você configurar um valor de tempo limite, ele não deverá ser inferior a 15 segundos para permitir que o Lambda seja iniciado a frio e que a instrumentação do canário seja inicializada.

> **Note**
>
> As seguintes versões de runtime do CloudWatch Synthetics se tornaram obsoletas em 8 de janeiro de 2024. Isso ocorre porque o AWS Lambda descontinuou o runtime do Lambda Node.js 14 em 4 de dezembro de 2023.
>
> - `syn-nodejs-puppeteer-3.9`
> - `syn-nodejs-puppeteer-3.8`
> - `syn-nodejs-puppeteer-3.7`
> - `syn-nodejs-puppeteer-3.6`
> - `syn-nodejs-puppeteer-3.5`
>
> As seguintes versões de runtime do CloudWatch Synthetics se tornaram obsoletas em 13 de novembro de 2022. Isso ocorre porque o AWS Lambda descontinuou o runtime do Lambda Node.js 12 em 14 de novembro de 2022.
>
> - `syn-nodejs-puppeteer-3.4`
> - `syn-nodejs-puppeteer-3.3`
> - `syn-nodejs-puppeteer-3.2`
> - `syn-nodejs-puppeteer-3.1`
> - `syn-nodejs-puppeteer-3.0`
>
> Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

syn-nodejs-puppeteer-7.0

O runtime `syn-nodejs-puppeteer-7.0` é a versão de runtime mais recente para Node.js 18.x de runtime do Lambda. Ele usa Node.js e Puppeteer.

Principais dependências:

- Runtime Node.js 18.x do Lambda
- Puppeteer-core versão 21.9.0
- Chromium versão 121.0.6167.139

Tamanho do código:

O tamanho do código e das dependências que você pode empacotar nesse runtime é de 80 MB.

Novos recursos no syn-nodejs-puppeteer-7.0:

- Versões atualizadas das bibliotecas empacotadas no Puppeteer e no Chromium: as dependências do Puppeteer e do Chromium foram atualizadas para novas versões.

> **Important**
>
> A mudança do Puppeteer 19.7.0 para o Puppeteer 21.9.0 promove mudanças significativas em relação a testes e filtros. Para obter mais informações, consulte as seções PRINCIPAIS ALTERAÇÕES em puppeteer: v20.0.0 e puppeteer-core: v21.0.0.

Atualização recomendada para o AWS SDK v3

O runtime do Lambda nodejs18.x não é compatível com o AWS SDK v2. É altamente recomendável migrar para o AWS SDK v3.

syn-nodejs-puppeteer-6.2

Principais dependências:

- Runtime Node.js 18.x do Lambda
- Puppeteer-core versão 19.7.0
- Chromium versão 111.0.5563.146

Novos recursos no syn-nodejs-puppeteer-6.2:

- Versões atualizadas das bibliotecas agrupadas no Chromium
- Monitoramento de armazenamento efêmero — Este runtime adiciona monitoramento de armazenamento efêmero às contas dos clientes.
- Correções de erros

syn-nodejs-puppeteer-5.2

O runtime `syn-nodejs-puppeteer-5.2` é a versão de runtime mais recente para Node.js 16.x de runtime do Lambda. Ele usa Node.js e Puppeteer.

Principais dependências:

- Runtime Node.js 16.x do Lambda
- Puppeteer-core versão 19.7.0
- Chromium versão 111.0.5563.146

Novos recursos no syn-nodejs-puppeteer-5.2:

- Versões atualizadas das bibliotecas agrupadas no Chromium
- Correções de erros

syn-nodejs-puppeteer-6.1

> **Important**
>
> Esta versão de runtime está programada para defasagem em 8 de março de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 18.x do Lambda
- Puppeteer-core versão 19.7.0
- Chromium versão 111.0.5563.146

Novos recursos no syn-nodejs-puppeteer-6.1:

- Melhorias na estabilidade: adicionada uma lógica de repetição automática para lidar com erros intermitentes de execução do Puppeteer.
- Upgrades de dependências: atualiza alguns pacotes de dependências de terceiros.
- Canários sem permissões do Amazon S3: correções de bugs para que os canários que não têm qualquer permissão do Amazon S3 ainda possam ser executados. Esses canários sem

permissões do Amazon S3 não poderão carregar capturas de tela ou outros artefatos para o Amazon S3. Para obter mais informações sobre permissões para canários, consulte Funções e permissões necessárias para canaries.

> **Important**
>
> IMPORTANTE: a dependência incluída do AWS SDK para JavaScript v2 será removida e atualizada para uso do AWS SDK para JavaScript v3 em uma versão futura do runtime. Quando isso acontecer, será possível atualizar suas referências de código canário. Como alternativa, é possível continuar referenciando e usando a dependência incluída do AWS SDK para JavaScript v2 adicionando-a como uma dependência ao arquivo zip do código-fonte.

syn-nodejs-puppeteer-6.0

> **Important**
>
> Esta versão de runtime está programada para defasagem em 8 de março de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 18.x do Lambda
- Puppeteer-core versão 19.7.0
- Chromium versão 111.0.5563.146

Novos recursos no syn-nodejs-puppeteer-6.0:

- Atualização de dependência: a dependência do Node.js foi atualizada para 18.x.
- Suporte ao modo de interceptação: o suporte ao modo de interceptação cooperativa do Puppeteer foi adicionado à biblioteca de runtime de um canário do Synthetics.
- Alteração do comportamento de rastreamento: o comportamento de rastreamento padrão foi alterado para rastrear somente as solicitações fetch e xhr, e não rastrear as solicitações de

recursos. Você pode ativar o rastreamento de solicitações de recursos configurando a opção `traceResourceRequests`.

- Métrica de duração refinada: a métrica `Duration` agora exclui o tempo de operação que o canário usa para carregar artefatos, fazer capturas de tela e gerar métricas do CloudWatch. Os valores da métrica `Duration` são relatados ao CloudWatch, e você também pode visualizá-los no console do Synthetics.
- Correção de erro: limpa o core dump gerado quando o Chromium trava durante uma execução de canário.

> **Important**
>
> IMPORTANTE: a dependência incluída do AWS SDK para JavaScript v2 será removida e atualizada para uso do AWS SDK para JavaScript v3 em uma versão futura do runtime. Quando isso acontecer, será possível atualizar suas referências de código canário. Como alternativa, é possível continuar referenciando e usando a dependência incluída do AWS SDK para JavaScript v2 adicionando-a como uma dependência ao arquivo zip do código-fonte.

syn-nodejs-puppeteer-5.1

> **Important**
>
> Esta versão de runtime está programada para defasagem em 8 de março de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 16.x do Lambda
- Puppeteer-core versão 19.7.0
- Chromium versão 111.0.5563.146

Correções de erros no syn-nodejs-puppeteer-5.1:

- Correção de erros: este runtime corrige um bug no `syn-nodejs-puppeteer-5.0` em que os arquivos HAR criados pelos canários não tinham cabeçalhos de solicitação.

syn-nodejs-puppeteer-5.0

> **Important**
>
> Esta versão de runtime está programada para defasagem em 8 de março de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 16.x do Lambda
- Puppeteer-core versão 19.7.0
- Chromium versão 111.0.5563.146

Novos recursos no syn-nodejs-puppeteer-5.0:

- Atualização de dependência: a versão Puppeteer-core foi atualizada para 19.7.0. A versão do Chromium foi atualizada para 111.0.5563.146.

> **Important**
>
> A nova versão do Puppeteer-core não é totalmente compatível com as versões anteriores do Puppeteer. Algumas das mudanças nesta versão podem fazer com que os canários existentes que usam funções obsoletas do Puppeteer falhem. Para obter mais informações, consulte as alterações mais importantes nos logs de alterações das versões 19.7.0 até 6.0 do Puppeteer Core, nos Logs de alterações do Puppeteer.

syn-nodejs-puppeteer-4.0

> **Important**
>
> Esta versão de runtime está programada para defasagem em 8 de março de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 16.x do Lambda
- Puppeteer-core versão 5.5.0
- Chromium versão 92.0.4512

Novos recursos no syn-nodejs-puppeteer-4.0:

- Atualização de dependência: a dependência do Node.js foi atualizada para 16.x.

Runtimes que foram descontinuados para Node.js e Tuppeteer

Os seguintes runtimes para Node.js e Puppeteer foram descontinuados.

syn-nodejs-puppeteer-3.9

> **Important**
>
> Esta versão de runtime foi defasada em 8 de janeiro de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 14.x do Lambda
- Puppeteer-core versão 5.5.0
- Chromium versão 92.0.4512

Novos recursos no syn-nodejs-puppeteer-3.9:

- Upgrades de dependências: atualiza alguns pacotes de dependências de terceiros.

syn-nodejs-puppeteer-3.8

> **Important**
>
> Esta versão de runtime foi defasada em 8 de janeiro de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 14.x do Lambda
- Puppeteer-core versão 5.5.0
- Chromium versão 92.0.4512

Novos recursos no syn-nodejs-puppeteer-3.8:

- Limpeza de perfis: os perfis do Chromium agora são limpos após cada execução de canário.

Correções de erros no syn-nodejs-puppeteer-3.8:

- Correções de erros: antes, os canários de monitoramento visual, às vezes, paravam de funcionar bem após uma execução, sem capturas de tela. Esse problema já foi corrigido.

syn-nodejs-puppeteer-3.7

> **Important**
>
> Esta versão de runtime foi defasada em 8 de janeiro de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 14.x do Lambda
- Puppeteer-core versão 5.5.0

- Chromium versão 92.0.4512

Novos recursos no syn-nodejs-puppeteer-3.7:

- Aperfeiçoamento de registros em log: o canário carregará os logs para o Amazon S3 mesmo se o tempo limite expirar ou o canário falhar.
- Redução do tamanho da camada do Lambda: o tamanho da camada do Lambda usada para canários é reduzido em 34%.

Correções de bugs em syn-nodejs-puppeteer-3.7:

- Correções de erros: as fontes em japonês, chinês simplificado e chinês tradicional serão renderizadas corretamente.

syn-nodejs-puppeteer-3.6

> **Important**
>
> Esta versão de runtime foi defasada em 8 de janeiro de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 14.x do Lambda
- Puppeteer-core versão 5.5.0
- Chromium versão 92.0.4512

Novos recursos no syn-nodejs-puppeteer-3.6:

- Carimbos de data/hora mais precisos: as horas de início e de parada das execuções de canários agora têm precisão de milissegundos.

syn-nodejs-puppeteer-3.5

> **Important**
>
> Esta versão de runtime foi defasada em 8 de janeiro de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 14.x do Lambda
- Puppeteer-core versão 5.5.0
- Chromium versão 92.0.4512

Novos recursos no syn-nodejs-puppeteer-3.5:

- Dependências atualizadas: os únicos novos recursos neste runtime são as dependências atualizadas.

syn-nodejs-puppeteer-3.4

> **Important**
>
> Esta versão do runtime tornou-se obsoleta em 13 de novembro de 2022. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 12.x do Lambda
- Puppeteer-core versão 5.5.0
- Chromium versão 88.0.4298.0

Novos recursos em syn-nodejs-puppeteer-3.4:

- Função de manipulador personalizado: agora você pode usar uma função de manipulador personalizado para seus scripts do canário. Os tempos de execução anteriores exigiam que o ponto de entrada do script incluísse `.handler`.

  Você também pode colocar scripts do canário em qualquer pasta e passar o nome da pasta como parte do manipulador. Por exemplo, `MyFolder/MyScriptFile.functionname` pode ser usado como um ponto de entrada.
- Informações sobre o arquivo HAR expandido: agora você pode ver solicitações ruins, pendentes e incompletas nos arquivos HAR produzidos por canaries.

syn-nodejs-puppeteer-3.3

> **Important**
>
> Esta versão do runtime tornou-se obsoleta em 13 de novembro de 2022. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 12.x do Lambda
- Puppeteer-core versão 5.5.0
- Chromium versão 88.0.4298.0

Novos recursos em syn-nodejs-puppeteer-3.3:

- Mais opções para a criptografia de artefatos: em canários que usam esse runtime ou uma versão posterior, é possível optar por usar uma chave do AWS KMS gerenciada pelo cliente ou uma chave gerenciada pelo Amazon S3 em vez de usar uma chave gerenciada da AWS para criptografar artefatos que o canário armazena no Amazon S3. Para ter mais informações, consulte Criptografar artefatos do canário.

syn-nodejs-puppeteer-3.2

> **Important**
>
> Esta versão do runtime tornou-se obsoleta em 13 de novembro de 2022. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 12.x do Lambda
- Puppeteer-core versão 5.5.0
- Chromium versão 88.0.4298.0

Novos recursos em syn-nodejs-puppeteer-3.2:

- Monitoramento visual com capturas de tela: canaries que usam esse runtime ou posteriores podem comparar uma captura de tela feita durante uma execução a uma versão de linha de base da mesma captura de tela. Se as capturas de tela forem mais diferentes do que um limite de porcentagem especificado, o canário falhará. Para obter mais informações, consulte Monitoramento visual ou Esquema de monitoramento visual.
- Novas funções relativas a dados confidenciais: você pode impedir que dados sigilosos sejam exibidos em logs e relatórios do canário. Para ter mais informações, consulte SyntheticsLogHelper class.
- Função desafada: a classe `RequestResponseLogHelper` está defasada em favor de outras opções de configuração. Para ter mais informações, consulte RequestResponseLogHelper class.

syn-nodejs-puppeteer-3.1

> **Important**
>
> Esta versão do runtime tornou-se obsoleta em 13 de novembro de 2022. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 12.x do Lambda
- Puppeteer-core versão 5.5.0
- Chromium versão 88.0.4298.0

Novos recursos em syn-nodejs-puppeteer-3.1:

- Capacidade de configurar métricas do CloudWatch: com esse runtime, é possível desabilitar as métricas que não são necessárias. Caso contrário, os canários publicam várias métricas do CloudWatch para cada execução do canário.
- Vinculação de captura de tela: é possível vincular uma captura de tela a uma etapa do canário após a conclusão da etapa. Para fazer isso, faça a captura de tela pelo método takeScreenshot, usando o nome da etapa à qual você deseja associar a captura de tela. Por exemplo, convém executar uma etapa, adicionar um tempo de espera e depois fazer a captura de tela.
- O esquema do monitor de heartbit pode monitorar várias URLs: é possível usar o esquema de monitoramento de pulsação no console do CloudWatch para monitorar várias URLs e ver o status, a duração, as capturas de tela associadas e o motivo da falha de cada URL no resumo da etapa do relatório de execução do canário.

syn-nodejs-puppeteer-3.0

> **Important**
>
> Esta versão do runtime tornou-se obsoleta em 13 de novembro de 2022. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 12.x do Lambda
- Puppeteer-core versão 5.5.0
- Chromium versão 88.0.4298.0

Novos recursos em syn-nodejs-puppeteer-3.0:

- Dependências atualizadas: essa versão usa o Puppeteer versão 5.5.0, Node.js 12.x e Chromium 88.0.4298.0.

- Acesso entre regiões: agora é possível especificar um bucket do S3 em outra região como o bucket onde o canário armazena seus arquivos de log, capturas de tela e arquivos HAR.
- Novas funções disponíveis: essa versão adiciona funções de biblioteca para recuperar o nome do canário e a versão do runtime do Synthetics.

  Para ter mais informações, consulte Classe Synthetics.

syn-nodejs-2.2

Esta seção contém informações sobre a versão de runtime `syn-nodejs-2.2`.

> **Important**
>
> Esta versão de runtime foi defasada em 28 de maio de 2021. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 10.x do Lambda
- Puppeteer-core versão 3.3.0
- Chromium versão 83.0.4103.0

Novos recursos em syn-nodejs-2.2:

- Monitore seus canários como etapas HTTP: agora você pode testar várias APIs em um único canário. Cada API é testada como uma etapa HTTP separada, e o CloudWatch Synthetics monitora o status de cada etapa usando métricas de etapas e o relatório de etapas do CloudWatch Synthetics. O CloudWatch Synthetics cria as métricas `SuccessPercent` e `Duration` para cada etapa HTTP.

  Essa funcionalidade é implementada pela função executeHttpStep(stepName, requestOptions, callback, stepConfig). Para ter mais informações, consulte executeHttpStep(stepName, requestOptions, [callback], [stepConfig]).

  O esquema do canário da API é atualizado para usar esse novo recurso.
- Relatórios de solicitações HTTP: agora é possível exibir relatórios detalhados de solicitações HTTP que capturam detalhes como cabeçalhos de solicitação/resposta, corpo de resposta,

código de status, tempos de erro e performance, tempo de conexão TCP, tempo de handshake TLS, tempo de primeiro byte e tempo de transferência de conteúdo. Todas as solicitações HTTP que usam o módulo HTTP/HTTPS nos bastidores são capturadas aqui. Cabeçalhos e corpo de resposta não são capturados por padrão, mas podem ser habilitados definindo opções de configuração.

- Configuração global e no nível da etapa: é possível definir as configurações do CloudWatch Synthetics no nível global, que são aplicadas a todas as etapas dos canaries. Também é possível substituir essas configurações no nível de etapa aprovando pares de chave-valor de configuração para habilitar ou desabilitar determinadas opções.

  Para ter mais informações, consulte Classe SyntheticsConfiguration.

- Continuar na configuração de falha da etapa: é possível escolher continuar a execução do canário quando uma etapa falhar. Para a função `executeHttpStep`, isso é ativado por padrão. Você pode definir essa opção uma vez no nível global ou configurá-la de modo diferente por etapa.

syn-nodejs-2.1

> **Important**
>
> Esta versão de runtime foi defasada em 28 de maio de 2021. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 10.x do Lambda
- Puppeteer-core versão 3.3.0
- Chromium versão 83.0.4103.0

Novos recursos em syn-nodejs-2.1:

- Comportamento de tela configurável: fornece a capacidade de desativar a obtenção de capturas de tela por canaries de interface do usuário. Em canaries que usam versões anteriores dos tempos de execução, os canaries de interface do usuário sempre obtêm capturas de tela antes e depois de cada etapa. Com `syn-nodejs-2.1`, isso é configurável. A desativação de capturas de tela pode reduzir os custos de armazenamento do Amazon S3 e ajudar você a cumprir as normas da HIPAA. Para ter mais informações, consulte Classe SyntheticsConfiguration.

- Personalizar os parâmetros de inicialização do Google Chrome: agora é possível configurar os argumentos usados quando um canário inicia uma janela do navegador Google Chrome. Para ter mais informações, consulte launch(options).

Pode haver um pequeno aumento na duração do canário ao usar syn-nodejs-2.0 ou posterior, comparado a versões anteriores dos tempos de execução do canário.

syn-nodejs-2.0

> **Important**
>
> Esta versão de runtime foi defasada em 28 de maio de 2021. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 10.x do Lambda
- Puppeteer-core versão 3.3.0
- Chromium versão 83.0.4103.0

Novos recursos em syn-nodejs-2.0:

- Dependências atualizadas: essa versão de runtime usa o Puppeteer-core versão 3.3.0 e Chromium versão 83.0.4103.0
- Compatibilidade com rastreamento ativo do X-Ray. Quando um canário tem o rastreamento habilitado, os rastreamentos de X-Ray são enviados a todas as chamadas realizadas pelo canário que usam o navegador, o AWS SDK ou módulos HTTP ou HTTPS. Canários com rastreamento habilitado aparecem no mapa de rastreamento do X-Ray, mesmo quando não enviam solicitações a outros serviços ou aplicações que tenham rastreamento habilitado. Para ter mais informações, consulte Canaries e rastreamento do X-Ray.
- Relatórios do Synthetics: para cada execução do canário, o CloudWatch Synthetics cria um relatório chamado `SyntheticsReport-PASSED.json` ou `SyntheticsReport-FAILED.json` que registra dados como hora de início, hora de término, status e falhas. Ele também registra o status PASSAD/FAILED de cada etapa do script do canário e falhas e capturas de tela obtidas em cada etapa.

- Relatório do verificador de link quebrado: a nova versão do verificador de link quebrado incluído neste runtime cria um relatório contendo os links verificados, o código de status, o motivo da falha (se houver) e as capturas de tela da página de origem e de destino.
- Novas métricas do CloudWatch: o Synthetics publica métricas denominadas `2xx`, `4xx`, `5xx` e `RequestFailed` no namespace `CloudWatchSynthetics`. Essas métricas mostram o número de 200s, 400s, 500s e falhas de solicitação nas execuções do canário. Com essa versão de runtime, essas métricas são relatadas apenas para canaries de interface do usuário e não são relatadas para canaries de API. Também são relatadas para canaries de API que começam com a versão de runtime `syn-nodejs-puppeteeer-2.2`.
- Arquivos HAR classificáveis: agora é possível classificar seus arquivos HAR por código de status, tamanho da solicitação e duração.
- Métricas de carimbo de data/hora: as métricas do CloudWatch agora são relatadas com base no tempo de invocação do Lambda em vez do horário de término da execução do canário.

Correções de bugs em syn-nodejs-2.0:

- Corrigiu-se o problema em que os erros de carregamento de artefatos do canário não eram relatados. Esses erros são agora apresentados como erros de execução.
- Corrigiu-se o problema em que solicitações redirecionadas (3xx) eram registradas incorretamente como erros.
- Corrigiu-se o problema das capturas de tela numeradas a partir de 0. Agora elas devem começar a partir de 1.
- Corrigiu-se o problema de capturas de tela ilegíveis para fontes chinesas e japonesas.

Pode haver um pequeno aumento na duração do canário ao usar syn-nodejs-2.0 ou posterior, comparado a versões anteriores dos tempos de execução do canário.

syn-nodejs-2.0-beta

> **Important**
>
> Esta versão de runtime foi defasada em 8 de fevereiro de 2021. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Runtime Node.js 10.x do Lambda
- Puppeteer-core versão 3.3.0
- Chromium versão 83.0.4103.0

Novos recursos em syn-nodejs-2.0-beta:

- Dependências atualizadas: essa versão de runtime usa o Puppeteer-core versão 3.3.0 e Chromium versão 83.0.4103.0
- Relatórios do Synthetics: para cada execução do canário, o CloudWatch Synthetics cria um relatório chamado `SyntheticsReport-PASSED.json` ou `SyntheticsReport-FAILED.json` que registra dados como hora de início, hora de término, status e falhas. Ele também registra o status PASSAD/FAILED de cada etapa do script do canário e falhas e capturas de tela obtidas em cada etapa.
- Relatório do verificador de link quebrado: a nova versão do verificador de link quebrado incluído neste runtime cria um relatório contendo os links verificados, o código de status, o motivo da falha (se houver) e as capturas de tela da página de origem e de destino.
- Novas métricas do CloudWatch: o Synthetics publica métricas denominadas `2xx`, `4xx`, `5xx` e `RequestFailed` no namespace `CloudWatchSynthetics`. Essas métricas mostram o número de 200s, 400s, 500s e falhas de solicitação nas execuções do canário. Essas métricas são relatadas apenas para canaries de interface do usuário e não são relatadas para canaries de API.
- Arquivos HAR classificáveis: agora é possível classificar seus arquivos HAR por código de status, tamanho da solicitação e duração.
- Métricas de carimbo de data/hora: as métricas do CloudWatch agora são relatadas com base no tempo de invocação do Lambda em vez do horário de término da execução do canário.

Correções de bugs no syn-nodejs-2.0-beta:

- Corrigiu-se o problema em que os erros de carregamento de artefatos do canário não eram relatados. Esses erros são agora apresentados como erros de execução.
- Corrigiu-se o problema em que solicitações redirecionadas (3xx) eram registradas incorretamente como erros.
- Corrigiu-se o problema das capturas de tela numeradas a partir de 0. Agora elas devem começar a partir de 1.
- Corrigiu-se o problema de capturas de tela ilegíveis para fontes chinesas e japonesas.

syn-1.0

> **Important**
>
> Essa versão de runtime está programada para defasagem em 28 de maio de 2021. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

A primeira versão de runtime do Synthetics é `syn-1.0`.

Principais dependências:

- Runtime Node.js 10.x do Lambda
- Puppeteer-core versão 1.14.0
- A versão do Chromium que corresponde ao Puppeteer-core 1.14.0

Versões de runtime usando Python e Selenium Webdriver

As seções a seguir contêm informações sobre as versões do runtime do CloudWatch Synthetics para Python e Selenium Webdriver. O Selenium é uma ferramenta de automação de navegador de código aberto. Para obter mais informações sobre o Selenium, consulte www.selenium.dev/

A convenção de nomenclatura para essas versões do runtime é `syn-`*`language`*`-`*`framework`*`-`*`majorversion`*`.`*`minorversion`*.

> **Important**
>
> As versões de runtime do CloudWatch Synthetics a seguir estão programadas para desativação em 8 de março de 2024.
>
> - `syn-python-selenium-2.0`
> - `syn-python-selenium-1.3`
> - `syn-python-selenium-1.2`
> - `syn-python-selenium-1.1`
> - `syn-python-selenium-1.0`

Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

syn-python-selenium-3.0

A versão 3.0 é o runtime mais recente do CloudWatch Synthetics para Python e Selenium.

Principais dependências:

- Python 3.8
- Selenium 4.15.1
- Chromium versão 121.0.6167.139

Novos recursos no syn-python-selenium-3.0:

- Versões atualizadas das bibliotecas empacotadas no Chromium: a dependência do Chromium está atualizada para uma nova versão.

syn-python-selenium-2.1

Principais dependências:

- Python 3.8
- Selenium 4.15.1
- Chromium versão 111.0.5563.146

Novos recursos no syn-python-selenium-2.1:

- Versões atualizadas das bibliotecas empacotadas no Chromium: as dependências do Chromium e do Selenium foram atualizadas para novas versões.

syn-python-selenium-2.0

> **Important**
>
> Esta versão de runtime está programada para defasagem em 8 de março de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Python 3.8
- Selenium 4.10.0
- Chromium versão 111.0.5563.146

Novos recursos no syn-python-selenium-2.0:

- Dependências atualizadas: as dependências do Chromium e do Selenium foram atualizadas para novas versões.

Correções de erros no syn-python-selenium-2.0:

- Carimbo de data/hora adicionado: um carimbo de data/hora foi adicionado aos logs do canário.
- Reutilização de sessão: um bug foi corrigido para que os canários agora sejam impedidos de reutilizar a sessão da execução anterior do canário.

syn-python-selenium-1.3

> **Important**
>
> Esta versão de runtime está programada para defasagem em 8 de março de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Python 3.8
- Selenium 3.141.0

- Chromium versão 92.0.4512.0

Novos recursos no syn-python-selenium-1.3:

- Carimbos de data/hora mais precisos: as horas de início e de parada das execuções de canários agora têm precisão de milissegundos.

syn-python-selenium-1.2

> **Important**
>
> Esta versão de runtime está programada para defasagem em 8 de março de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Python 3.8
- Selenium 3.141.0
- Chromium versão 92.0.4512.0

- Dependências atualizadas: os únicos novos recursos neste runtime são as dependências atualizadas.

syn-python-selenium-1.1

> **Important**
>
> Esta versão de runtime está programada para defasagem em 8 de março de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Python 3.8
- Selenium 3.141.0

- Chromium versão 83.0.4103.0

Recursos:

- Função de manipulador personalizado: agora você pode usar uma função de manipulador personalizado para seus scripts do canário. Os tempos de execução anteriores exigiam que o ponto de entrada do script incluísse `.handler`.

  Você também pode colocar scripts do canário em qualquer pasta e passar o nome da pasta como parte do manipulador. Por exemplo, `MyFolder/MyScriptFile.functionname` pode ser usado como um ponto de entrada.
- Opções de configuração para adicionar métricas e configurações de falha de etapas: essas opções já estavam disponíveis em tempos de execução para canaries Node.js. Para ter mais informações, consulte Classe SyntheticsConfiguration.
- Argumentos personalizados no Chrome: agora você pode abrir um navegador no modo anônimo ou passar a configuração do servidor de proxy. Para ter mais informações, consulte Chrome().
- Buckets de artefatos entre regiões: um canário pode armazenar artefatos em um bucket do Amazon S3 em uma região diferente.
- Correções de erros, incluindo uma correção para o problema **`index.py`**: com os tempos de execução anteriores, um arquivo canário chamado `index.py` causava exceções, porque entrava em conflito com o nome do arquivo da biblioteca. Esse problema já foi corrigido.

syn-python-selenium-1.0

> **Important**
>
> Esta versão de runtime está programada para defasagem em 8 de março de 2024. Para ter mais informações, consulte Política de suporte ao runtime do CloudWatch Synthetics.

Principais dependências:

- Python 3.8
- Selenium 3.141.0
- Chromium versão 83.0.4103.0

Recursos:

- Suporte ao Selenium: é possível escrever scripts do canário usando o framework de teste do Selenium. Você pode levar seus scripts Selenium de outro lugar ao CloudWatch Synthetics com alterações mínimas, e eles funcionarão com produtos da AWS.

Escrever um script do canário

As seções a seguir explicam como escrever um script de canário e como integrar um canário a outros serviços da AWS e com dependências e bibliotecas externas.

Tópicos

Gravar um script do canário Node.js

Tópicos

Criar um canário do CloudWatch Synthetics do zero

Veja a seguir um exemplo de script do canário mínimo do Synthetics. Esse script é executado com êxito e retorna uma string. Para ver como é um canário com falha, altere `let fail = false;` para `let fail = true;`.

É necessário definir uma função de ponto de entrada para o script do canário. Para ver como os arquivos são carregados no local do Amazon S3 especificado como `ArtifactS3Location` do

canário, crie esses arquivos na pasta /tmp. Após a execução do script, o status de aprovação/falha e as métricas de duração são publicadas no CloudWatch, e os arquivos em /tmp são carregados no S3.

```
const basicCustomEntryPoint = async function () {

    // Insert your code here

    // Perform multi-step pass/fail check

    // Log decisions made and results to /tmp

    // Be sure to wait for all your code paths to complete
    // before returning control back to Synthetics.
    // In that way, your canary will not finish and report success
    // before your code has finished executing

    // Throw to fail, return to succeed
    let fail = false;
    if (fail) {
        throw "Failed basicCanary check.";
    }

    return "Successfully completed basicCanary checks.";
};

exports.handler = async () => {
    return await basicCustomEntryPoint();
};
```

Depois, expandiremos o script para usar o registro em log do Synthetics e fazer uma chamada usando o AWS SDK. Para fins de demonstração, esse script criará um cliente do Amazon DynamoDB e fará uma chamada para a API listTables do DynamoDB. Ele registra a resposta à solicitação e os logs são aprovados ou falham dependendo se a solicitação foi bem-sucedida.

```
const log = require('SyntheticsLogger');
const AWS = require('aws-sdk');
// Require any dependencies that your script needs
// Bundle additional files and dependencies into a .zip file with folder structure
// nodejs/node_modules/additional files and folders

const basicCustomEntryPoint = async function () {
```

```
    log.info("Starting DynamoDB:listTables canary.");

    let dynamodb = new AWS.DynamoDB();
    var params = {};
    let request = await dynamodb.listTables(params);
    try {
        let response = await request.promise();
        log.info("listTables response: " + JSON.stringify(response));
    } catch (err) {
        log.error("listTables error: " + JSON.stringify(err), err.stack);
        throw err;
    }

    return "Successfully completed DynamoDB:listTables canary.";
};

exports.handler = async () => {
    return await basicCustomEntryPoint();
};
```

Empacotamento dos arquivos do canário para Node.js

Se você estiver carregando scripts do canário usando um local do Amazon S3, o arquivo zip deverá incluir seu script sob essa estrutura de pastas: `nodejs/node_modules/`*`myCanaryFilename.js file`*.

Se você tiver mais de um arquivo `.js` ou se o script estiver condicionado a uma dependência, será possível agrupá-los em um único arquivo ZIP que contenha a estrutura da pasta `nodejs/node_modules/`*`myCanaryFilename.js file and other folders and files`*. Se estiver usando `syn-nodejs-puppeteer-3.4` ou posterior, você também poderá colocar os arquivos do canário em outra pasta e criar a estrutura de pastas dessa forma: `nodejs/node_modules/`*`myFolder`*`/`*`myCanaryFilename.js file and other folders and files`*.

Nome do manipulador

Defina o ponto de entrada do script do canário (manipulador) como `myCanaryFilename.functionName` para corresponder ao nome do arquivo do ponto de entrada do script. Se estiver usando um runtime anterior a `syn-nodejs-puppeteer-3.4`, então `functionName`deverá ser `handler`. Se estiver usando `syn-nodejs-puppeteer-3.4` ou posterior, você poderá escolher qualquer nome de função como manipulador. Se estiver usando `syn-nodejs-puppeteer-3.4` ou posterior, você também poderá armazenar o canário em uma

pasta separada, como `nodejs/node_modules/myFolder/my_canary_filename`. Se você armazenar em uma pasta separada, especifique esse caminho no ponto de entrada do script, como `myFolder/my_canary_filename.functionName`.

Alterar um script existente do Puppeteer para ser usado como canário do Synthetics

Esta seção explica como modificar scripts do Puppeteer para executá-los como scripts do canário do Synthetics. Para obter mais informações sobre o Puppeteer, consulte API do Puppeteer v1.14.0.

Vamos começar com este exemplo de script do Puppeteer:

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');
  await page.screenshot({path: 'example.png'});

  await browser.close();
})();
```

As etapas de conversão são as seguintes:

- Crie e exporte uma função `handler`. O manipulador é a função de ponto de entrada para o script. Se estiver usando um runtime anterior a `syn-nodejs-puppeteer-3.4`, a função do manipulador deverá ser nomeada `handler`. Se estiver usando `syn-nodejs-puppeteer-3.4` ou posterior, a função poderá ter qualquer nome, mas deverá ser o mesmo nome usado no script. Além disso, se você estiver usando `syn-nodejs-puppeteer-3.4` ou posterior, poderá armazenar scripts em qualquer pasta e especificar essa pasta como parte do nome do manipulador.

  ```
  const basicPuppeteerExample = async function () {};

  exports.handler = async () => {
      return await basicPuppeteerExample();
  };
  ```

- Use a dependência `Synthetics`.

  ```
  var synthetics = require('Synthetics');
  ```

- Use a função `Synthetics.getPage` para obter um objeto `Page` do Puppeteer.

```
const page = await synthetics.getPage();
```

  O objeto de página retornado pela função Synthetics.getPage tem os eventos page.on `request`, `response`e `requestfailed` instrumentados para registro em log. O Synthetics também define a geração de arquivos HAR para solicitações e respostas na página e adiciona o ARN do canário aos cabeçalhos do atendente do usuário de solicitações de saída na página.

O script agora está pronto para ser executado como um canário do Synthetics. Veja a seguir o script atualizado:

```
var synthetics = require('Synthetics');  // Synthetics dependency

const basicPuppeteerExample = async function () {
    const page = await synthetics.getPage(); // Get instrumented page from Synthetics
    await page.goto('https://example.com');
    await page.screenshot({path: '/tmp/example.png'}); // Write screenshot to /tmp
 folder
};

exports.handler = async () => {  // Exported handler function
    return await basicPuppeteerExample();
};
```

Variáveis de ambiente

É possível usar variáveis de ambiente ao criar canaries. Isso permite escrever um único script o canário e usar esse script com valores diferentes para criar rapidamente vários canários que tenham uma tarefa semelhante.

Por exemplo, suponhamos que sua organização tenha endpoints como `prod`, `dev` e `pre-release` para os diferentes estágios do desenvolvimento de seu software, e você necessite criar canaries para testar cada um desses endpoints. É possível escrever um único script do canário que testa seu software e especificar valores diferentes para a variável de ambiente do endpoint ao criar cada um dos três canários. Em seguida, ao criar um canário, especifique o script e os valores a serem usados para as variáveis de ambiente.

Os nomes das variáveis de ambiente podem conter letras, números e o caractere de sublinhado. Devem começar com uma letra e ter pelo menos dois caracteres. O tamanho total das variáveis

de ambiente não pode exceder 4 KB. Você não pode especificar nenhuma variável de ambiente reservada do Lambda como os nomes de suas variáveis de ambiente. Para obter mais informações sobre variáveis de ambiente reservadas, consulte Variáveis de ambiente do runtime.

> **Important**
>
> As chaves e os valores de variáveis de ambiente não são criptografados. Não armazene informações sigilosas neles.

O exemplo de script a seguir usa duas variáveis de ambiente. Esse script é para um canário que verifica se uma página da Web está disponível. Utiliza variáveis de ambiente para parametrizar tanto a URL que ele verifica como o nível de logs do CloudWatch Synthetics que ele usa.

A função a seguir define `LogLevel` para o valor da variável de ambiente `LOG_LEVEL`.

```
synthetics.setLogLevel(process.env.LOG_LEVEL);
```

Essa função define `URL` para o valor da variável de ambiente `URL`.

```
const URL = process.env.URL;
```

Este é o script completo. Ao criar um canário usando esse script, especifique os valores a serem usados para as variáveis de ambiente `LOG_LEVEL` e `URL`.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadEnvironmentVariable = async function () {

    // Setting the log level (0-3)
    synthetics.setLogLevel(process.env.LOG_LEVEL);
    // INSERT URL here
    const URL = process.env.URL;

    let page = await synthetics.getPage();
    //You can customize the wait condition here. For instance,
    //using 'networkidle2' may be less restrictive.
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});
    if (!response) {
```

```
        throw "Failed to load page!";
    }
    //Wait for page to render.
    //Increase or decrease wait time based on endpoint being monitored.
    await page.waitFor(15000);
    await synthetics.takeScreenshot('loaded', 'loaded');
    let pageTitle = await page.title();
    log.info('Page title: ' + pageTitle);
    log.debug('Environment variable:' + process.env.URL);

    //If the response status code is not a 2xx success code
    if (response.status() < 200 || response.status() > 299) {
        throw "Failed to load page!";
    }
};

exports.handler = async () => {
    return await pageLoadEnvironmentVariable();
};
```

Aprovar variáveis de ambiente para seu script

Para transmitir variáveis de ambiente para o script ao criar um canário no console, especifique as chaves e os valores das variáveis de ambiente na seção Environment variables (Variáveis de ambiente) no console. Para ter mais informações, consulte Criar um canário.

Para transmitir variáveis de ambiente pela API ou pela AWS CLI, use o parâmetro `EnvironmentVariables` na seção `RunConfig`. O exemplo a seguir é um comando AWS CLI que cria um canário que usa duas variáveis de ambiente com chaves de `Environment` e `Region`.

```
aws synthetics create-canary --cli-input-json '{
   "Name":"nameofCanary",
   "ExecutionRoleArn":"roleArn",
   "ArtifactS3Location":"s3://cw-syn-results-123456789012-us-west-2",
   "Schedule":{
      "Expression":"rate(0 minute)",
      "DurationInSeconds":604800
   },
   "Code":{
      "S3Bucket": "canarycreation",
      "S3Key": "cwsyn-mycanaryheartbeat-12345678-d1bd-1234-
abcd-123456789012-12345678-6a1f-47c3-b291-123456789012.zip",
      "Handler":"pageLoadBlueprint.handler"
```

```
   },
   "RunConfig": {
      "TimeoutInSeconds":60,
      "EnvironmentVariables": {
         "Environment":"Production",
         "Region": "us-west-1"
      }
   },
   "SuccessRetentionPeriodInDays":13,
   "FailureRetentionPeriodInDays":13,
   "RuntimeVersion":"syn-nodejs-2.0"
}'
```

Integrar o canário a outros produtos da AWS

Todos os canaries podem usar a biblioteca do AWS SDK. É possível usar essa biblioteca ao escrever o canário de forma a integrá-lo a outros serviços da AWS.

Para fazer isso, é necessário adicionar o código a seguir ao canário. Para estes exemplos, o AWS Secrets Manager é usado como o serviço ao qual o canário está se integrando.

- Importe o AWS SDK.

```
const AWS = require('aws-sdk');
```

- Crie um cliente para o serviço da AWS ao qual você está realizando a integração.

```
const secretsManager = new AWS.SecretsManager();
```

- Use o cliente para fazer chamadas de API para esse serviço.

```
var params = {
  SecretId: secretName
};
return await secretsManager.getSecretValue(params).promise();
```

O trecho de código de script do canário a seguir demonstra um exemplo de integração ao Secrets Manager com mais detalhes.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');
```

```
const AWS = require('aws-sdk');
const secretsManager = new AWS.SecretsManager();

const getSecrets = async (secretName) => {
    var params = {
        SecretId: secretName
    };
    return await secretsManager.getSecretValue(params).promise();
}

const secretsExample = async function () {
    let URL = "<URL>";
    let page = await synthetics.getPage();

    log.info(`Navigating to URL: ${URL}`);
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});

    // Fetch secrets
    let secrets = await getSecrets("secretname")

    /**
    * Use secrets to login.
    *
    * Assuming secrets are stored in a JSON format like:
    * {
    *   "username": "<USERNAME>",
    *   "password": "<PASSWORD>"
    * }
    **/
    let secretsObj = JSON.parse(secrets.SecretString);
    await synthetics.executeStep('login', async function () {
        await page.type(">USERNAME-INPUT-SELECTOR<", secretsObj.username);
        await page.type(">PASSWORD-INPUT-SELECTOR<", secretsObj.password);

        await Promise.all([
          page.waitForNavigation({ timeout: 30000 }),
          await page.click(">SUBMIT-BUTTON-SELECTOR<")
        ]);
    });

    // Verify login was successful
    await synthetics.executeStep('verify', async function () {
```

```
        await page.waitForXPath(">SELECTOR<", { timeout: 30000 });
    });
};

exports.handler = async () => {
    return await secretsExample();
};
```

Forçar seu canário a usar um endereço IP estático

É possível configurar um canário para que ele use um endereço IP estático.

Para forçar um canário a usar um endereço IP estático

1. Crie uma nova VPC. Para obter mais informações, consulte Using DNS with Your VPC.
2. Crie um novo gateway da Internet. Para obter mais informações, consulte Adicionar um gateway da Internet à VPC.
3. Crie uma sub-rede pública dentro de sua nova VPC.
4. Adicione uma nova tabela de rotas à VPC.
5. Adicione uma rota na nova tabela de rotas, que vai de `0.0.0.0/0` ao gateway da Internet.
6. Associe a nova tabela de rotas à sub-rede pública.
7. Crie um endereço de IP elástico. Para obter mais informações, consulte Endereços de IP elásticos.
8. Crie um novo gateway NAT e atribua-o à sub-rede pública e ao endereço de IP elástico.
9. Crie uma sub-rede privada dentro da VPC.
10. Adiciona uma rota à tabela de rotas padrão da VPC, que vai de `0.0.0.0/0` ao gateway NAT
11. Crie um canário.

Escrever um script o canário do Python

Esse script é executado com êxito e retorna uma string. Para ver como é um canário com falha, altere fail = False to fail = True

```
def basic_custom_script():
    # Insert your code here
    # Perform multi-step pass/fail check
    # Log decisions made and results to /tmp
    # Be sure to wait for all your code paths to complete
```

```
    # before returning control back to Synthetics.
    # In that way, your canary will not finish and report success
    # before your code has finished executing
    fail = False
    if fail:
        raise Exception("Failed basicCanary check.")
    return "Successfully completed basicCanary checks."
def handler(event, context):
    return basic_custom_script()
```

Empacotamento dos arquivos do canário para Python

Se você tiver mais de um arquivo .py ou se o script tiver uma dependência, será possível agrupá-los em um único arquivo ZIP. Se você usar o runtime `syn-python-selenium-1.1`, o arquivo ZIP deverá conter o arquivo .py canário principal dentro de uma pasta `python`, como `python/my_canary_filename.py`. Se você usar `syn-python-selenium-1.1` ou posterior, poderá usar uma pasta diferente, como `python/myFolder/my_canary_filename.py`.

Este arquivo ZIP deverá conter todas as pastas e arquivos necessários, mas os outros arquivos não precisam estar na pasta `python`.

Defina o ponto de entrada do script o canário como `my_canary_filename.functionName` para corresponder ao nome do arquivo e ao nome da função do ponto de entrada do script. Se você estiver usando o runtime `syn-python-selenium-1.0`, `functionName` deverá ser `handler`. Se estiver usando `syn-python-selenium-1.1` ou posterior, essa restrição de nome do manipulador não se aplicará, e você também poderá, opcionalmente, armazenar o canário em uma pasta separada, como `python/myFolder/my_canary_filename.py`. Se você armazenar em uma pasta separada, especifique esse caminho no ponto de entrada do script, como `myFolder/my_canary_filename.functionName`.

Alterar um script existente do Selenium para ser usado como canário do Synthetics

É possível modificar rapidamente um script existente para Python e Selenium a ser usado como canário. Para obter mais informações sobre o Selenium, consulte www.selenium.dev/

Neste exemplo, começaremos com o seguinte script Selenium:

```
from selenium import webdriver

def basic_selenium_script():
    browser = webdriver.Chrome()
```

```
    browser.get('https://example.com')
    browser.save_screenshot('loaded.png')

basic_selenium_script()
```

As etapas de conversão são as seguintes.

Para converter um script Selenium para ser usado como canário

1. Altere a instrução `import` para usar o Selenium a partir do módulo `aws_synthetics`:

   ```
   from aws_synthetics.selenium import synthetics_webdriver as webdriver
   ```

   O módulo Selenium a partir de `aws_synthetics` garante que o canário poderá emitir métricas e logs, gerar um arquivo HAR e trabalhar com outros recursos do CloudWatch Synthetics.
2. Crie uma função de manipulador e chame seu método Selenium. O manipulador é a função de ponto de entrada para o script.

   Se estiver usando `syn-python-selenium-1.0`, a função do manipulador deverá ser nomeada `handler`. Se estiver usando `syn-python-selenium-1.1` ou posterior, a função poderá ter qualquer nome, mas deverá ser o mesmo nome usado no script. Além disso, se você estiver usando `syn-python-selenium-1.1` ou posterior, poderá armazenar scripts em qualquer pasta e especificar essa pasta como parte do nome do manipulador.

   ```
   def handler(event, context):
       basic_selenium_script()
   ```

Agora, o script é atualizado para ser um canário do CloudWatch Synthetics. Veja a seguir o script atualizado:

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver

def basic_selenium_script():
    browser = webdriver.Chrome()
    browser.get('https://example.com')
    browser.save_screenshot('loaded.png')

def handler(event, context):
    basic_selenium_script()
```

Como alterar um script existente do Puppeteer Synthetics para autenticar certificados não padrão

Um caso de uso importante dos canários Synthetics é monitorar seus próprios endpoints. Se você quiser monitorar um endpoint que não esteja pronto para tráfego externo, esse monitoramento poderá, às vezes, significar que você não tenha um certificado adequado assinado por uma autoridade de certificação terceirizada confiável.

Duas soluções possíveis para esse cenário são as seguintes:

- Para autenticar um certificado de cliente, consulte Como validar a autenticação usando o Amazon CloudWatch Synthetics – Parte 2.
- Para autenticar um certificado autoassinado, consulte Como validar a autenticação com certificados autoassinados no Amazon CloudWatch Synthetics

Você não está limitado a essas duas opções ao usar os canários CloudWatch Synthetics. Você pode estender esses recursos e adicionar sua lógica de negócios estendendo o código canário.

> **Note**
>
> Os canários Synthetics executados em runtimes do Python têm o sinalizador `--ignore-certificate-errors` habilitado de forma inata, portanto, esses canários não devem ter problemas para acessar sites com configurações de certificado não padrão.

Funções da biblioteca disponíveis para scripts o canário

O CloudWatch Synthetics contém várias classes e funções incorporadas que podem ser chamadas ao gravar scripts Node.js a serem usados como canaries.

Algumas se aplicam a canaries de interface do usuário e de API. Outros se aplicam apenas a canaries de interface do usuário. Um canário de interface do usuário é um canário que usa a função `getPage()` e usa o Puppeteer como um driver da web para navegar e interagir com páginas da web.

> **Note**
>
> Sempre que você atualizar um canário para usar uma nova versão do runtime do Synthetics, todas as funções da biblioteca Synthetics que seu canário usar também serão

automaticamente atualizadas para a mesma versão do NodeJS que o runtime do Synthetics ofereça suporte.

Tópicos

- Funções de biblioteca disponíveis para scripts do canário do Node.js
- Funções da biblioteca disponíveis para scripts do canário do Python usando Selenium

Funções de biblioteca disponíveis para scripts do canário do Node.js

Esta seção lista as funções de biblioteca disponíveis para scripts do canário Node.js.

Tópicos

- Classes de biblioteca do Node.js e funções que se aplicam a todos os canaries
- Classes de biblioteca do Node.js e funções que se aplicam somente a canaries de interface do usuário
- Classes de biblioteca do Node.js e funções que se aplicam apenas a canaries de API

Classes de biblioteca do Node.js e funções que se aplicam a todos os canaries

As seguintes funções de biblioteca para Node.js do CloudWatch Synthetics são úteis para todos os canaries.

Tópicos

- Classe Synthetics
- Classe SyntheticsConfiguration
- Synthetics Logger
- SyntheticsLogHelper class

Classe Synthetics

As funções a seguir para todos os canaries estão na classe Synthetics.

addExecutionError(errorMessage, ex);

`errorMessage` descreve o erro, e `ex` é a exceção encontrada

Você pode usar `addExecutionError` para definir erros de execução eu seu canário. Ele faz o canário falhar sem interromper a execução do script. Também não afeta suas métricas `successPercent`.

Convém rastrear erros como erros de execução somente se eles não forem importantes para indicar o sucesso ou falha do script do canário.

A seguir há um exemplo de uso do elemento `addExecutionError`. Você está monitorando a disponibilidade de seu endpoint e fazendo capturas de tela depois que a página foi carregada. Como a falha de obter uma captura de tela não determina a disponibilidade do endpoint, é possível detectar quaisquer erros encontrados durante a captura de tela e adicioná-los como erros de execução. Suas métricas de disponibilidade ainda indicarão que o endpoint está ativo e em execução, mas o status do canário será marcado como falha. O bloco de código de exemplo a seguir captura esse erro e adiciona-o como erro de execução.

```
try {
        await synthetics.takeScreenshot(stepName, "loaded");
    } catch(ex) {
        synthetics.addExecutionError('Unable to take screenshot ', ex);
    }
```

getCanaryName();

Retorna o nome do canário.

getCanaryArn();

Retorna o ARN do canário.

getCanaryUserAgentString();

Retorna o agente de usuário personalizado do canário.

getRuntimeVersion();

Essa função está disponível na versão de runtime `syn-nodejs-puppeteer-3.0` e posteriores. Ele retorna a versão de runtime do Synthetics do canário. Por exemplo, o valor de retorno pode ser `syn-nodejs-puppeteer-3.0`.

getLogLevel();

Recupera o nível de log atual para a biblioteca do Synthetics. Os valores possíveis são os seguintes:

- `0`: debug
- `1`: info
- `2`: warn
- `3`: error

Exemplo:

```
let logLevel = synthetics.getLogLevel();
```

setLogLevel();

Define o nível de log para a biblioteca do Synthetics. Os valores possíveis são os seguintes:

- `0`: debug
- `1`: info
- `2`: warn
- `3`: error

Exemplo:

```
synthetics.setLogLevel(0);
```

Classe SyntheticsConfiguration

Essa classe está disponível apenas na versão de runtime `syn-nodejs-2.1` ou posteriores.

É possível usar a classe SyntheticsConfiguration para configurar o comportamento das funções da biblioteca do Synthetics. Por exemplo, você pode usar essa classe para configurar a função `executeStep()` para não obter capturas de tela.

É possível definir as configurações do CloudWatch Synthetics no nível global, que são aplicadas a todas as etapas dos canaries. Também é possível substituir essas configurações no nível de etapa aprovando pares de chave-valor de configuração.

Você pode transmitir opções no nível de etapa. Veja exemplos em async executeStep(stepName, functionToExecute, [stepConfig]); e executeHttpStep(stepName, requestOptions, [callback], [stepConfig]).

Definições de função:

setConfig(options)

`options` é um objeto, que é um conjunto de opções configuráveis para seu canário. As seções a seguir explicam os campos possíveis em *`options`*.

setConfig(options) para todos os canaries

Para canaries que usam `syn-nodejs-puppeteer-3.2` ou posteriores, (options) para o SetConfig pode incluir estes parâmetros:

- `includeRequestHeaders` (booliano): se deve incluir ou não cabeçalhos de solicitação no relatório. O padrão é `false`.
- `includeResponseHeaders` (booliano): se deve incluir ou não cabeçalhos de resposta no relatório. O padrão é `false`.
- `restrictedHeaders` (matriz): uma lista de valores de cabeçalho a serem ignorados, se os cabeçalhos forem incluídos. Isso se aplica aos cabeçalhos de solicitação e de resposta. Por exemplo, é possível ocultar suas credenciais aprovando includeRequestHeaders como `true` e restrictedHeaders como `['Authorization']`.
- `includeRequestBody` (booliano): se deve incluir ou não o corpo da solicitação no relatório. O padrão é `false`.
- `includeResponseBody` (booliano): se deve incluir ou não o corpo da resposta no relatório. O padrão é `false`.

setConfig(options) em relação a métricas do CloudWatch

Para canários que usam `syn-nodejs-puppeteer-3.1` ou posteriores, (options) para setConfig pode incluir os seguintes parâmetros boolianos que determinam quais métricas serão publicadas pelo canário. O padrão para cada uma dessas opções é `true`. As opções que começam com `aggregated` determinam se a métrica será emitida sem a dimensão `CanaryName`. É possível usar essas métricas para ver os resultados agregados de todos os seus canaries. As outras opções determinam se a métrica será emitida com a dimensão `CanaryName`. Você pode usar essas métricas para ver os resultados de cada canário individual.

Para obter uma lista de métricas do CloudWatch emitidas por canaries, consulte Métricas do CloudWatch publicadas por canaries.

- `failedCanaryMetric` (booliano): se deverá ou não emitir a métrica `Failed` (com a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `failedRequestsMetric` (booliano): se deverá ou não emitir a métrica `Failed requests` (com a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `_2xxMetric` (booliano): se deverá ou não emitir a métrica `2xx` (com a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `_4xxMetric` (booliano): se deverá ou não emitir a métrica `4xx` (com a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `_5xxMetric` (booliano): se deverá ou não emitir a métrica `5xx` (com a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `stepDurationMetric` (booliano): se deverá ou não emitir a métrica `Step duration` (com as dimensões `CanaryName` `StepName`) para esse canário. O padrão é `true`.
- `stepSuccessMetric` (booliano): se deverá ou não emitir a métrica `Step success` (com as dimensões `CanaryName` `StepName`) para esse canário. O padrão é `true`.
- `aggregatedFailedCanaryMetric` (booliano): se deverá ou não emitir a métrica `Failed` (sem a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `aggregatedFailedRequestsMetric` (booliano): se deverá ou não emitir a métrica `Failed Requests` (sem a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `aggregated2xxMetric` (booliano): se deverá ou não emitir a métrica `2xx` (sem a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `aggregated4xxMetric` (booliano): se deverá ou não emitir a métrica `4xx` (sem a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `aggregated5xxMetric` (booliano): se deverá ou não emitir a métrica `5xx` (sem a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `visualMonitoringSuccessPercentMetric` (booliano): se deverá ou não emitir a métrica `visualMonitoringSuccessPercent` para esse canário. O padrão é `true`.
- `visualMonitoringTotalComparisonsMetric` (booliano): se deverá ou não emitir a métrica `visualMonitoringTotalComparisons` para esse canário. O padrão é `false`.
- `stepsReport` (booliano): se deverá ou não reportar um resumo de execução de etapa. O padrão é `true`.
- `includeUrlPassword` (boolieano): se deverá ou não incluir uma senha que aparece na URL. Por padrão, as senhas que aparecem em URLs são editadas de logs e relatórios, para evitar a divulgação de dados sigilosos. O padrão é `false`.

- `restrictedUrlParameters` (matriz): uma lista de parâmetros de caminho de URL ou consulta a serem editados. Aplica-se a URLs que aparecem em logs, relatórios e erros. O parâmetro faz distinção entre maiúsculas e minúsculas. É possível transmitir um asterisco (*) como um valor para editar todos os valores de parâmetro de consulta e caminho de URL. O padrão é uma matriz vazia.
- `logRequest` (booliano): se cada solicitação em logs do canário será registrada ou não. Para canaries de interface do usuário, isso registra cada solicitação enviada pelo navegador. O padrão é `true`.
- `logResponse` (booliano): se cada resposta em logs do canário será registrada ou não. Para canaries de interface do usuário, registra todas as respostas recebidas pelo navegador. O padrão é `true`.
- `logRequestBody` (booliano): se os corpos da solicitação serão registrados junto com as solicitações em logs do canário. Essa configuração se aplica somente se `logRequest` é `true`. O padrão é `false`.
- `logResponseBody` (booliano): se os corpos da resposta serão registrados junto com as respostas em logs do canário. Essa configuração se aplica somente se `logResponse` é `true`. O padrão é `false`.
- `logRequestHeaders` (booliano): se os cabeçalhos da solicitação serão registrados junto com as solicitações em logs do canário. Essa configuração se aplica somente se `logRequest` é `true`. O padrão é `false`.

  `includeRequestHeaders` habilita cabeçalhos em artefatos.
- `logResponseHeaders` (booliano): se os cabeçalhos da resposta serão registrados junto com as respostas em logs do canário. Essa configuração se aplica somente se `logResponse` é `true`. O padrão é `false`.

  `includeResponseHeaders` habilita cabeçalhos em artefatos.

> **Note**
>
> As métricas `Duration` e `SuccessPercent` são sempre emitidas para cada canário, tanto com como sem a métrica `CanaryName`.

Métodos para habilitar ou desabilitar métricas

disableAggregatedRequestMetrics()

Não permite que o canário emita todas as métricas de solicitação emitidas sem a dimensão `CanaryName`.

disableRequestMetrics()

Desabilita todas as métricas de solicitação, inclusive métricas por canário e métricas agregadas em todos os canários.

disableStepMetrics()

Desativa todas as métricas de etapa, incluindo métricas de sucesso e de duração da etapa.

enableAggregatedRequestMetrics()

Habilita o canário a emitir todas as métricas de solicitação emitidas sem a dimensão `CanaryName`.

enableRequestMetrics()

Habilita todas as métricas de solicitação, inclusive métricas por canário e métricas agregadas em todos os canários.

enableStepMetrics()

Ativa todas as métricas de etapa, incluindo métricas de sucesso e métricas de duração da etapa.

get2xxMetric()

Retorna se o canário emitirá ou não uma métrica `2xx` com a dimensão `CanaryName`.

get4xxMetric()

Retorna se o canário emitirá ou não uma métrica `4xx` com a dimensão `CanaryName`.

get5xxMetric()

Retorna se o canário emitirá ou não uma métrica `5xx` com a dimensão `CanaryName`.

getAggregated2xxMetric()

Retorna se o canário emitirá ou não uma métrica `2xx` sem a dimensão.

getAggregated4xxMetric()

Retorna se o canário emitirá ou não uma métrica `4xx` sem a dimensão.

getAggregatedFailedCanaryMetric()

Retorna se o canário emitirá ou não uma métrica `Failed` sem a dimensão.

getAggregatedFailedRequestsMetric()

Retorna se o canário emitirá ou não uma métrica `Failed requests` sem a dimensão .

getAggregated5xxMetric()

Retorna se o canário emitirá ou não uma métrica `5xx` sem a dimensão .

getFailedCanaryMetric()

Retorna se o canário emitirá ou não uma métrica `Failed` com a dimensão `CanaryName`.

getFailedRequestsMetric()

Retorna se o canário emitirá ou não uma métrica `Failed requests` com a dimensão `CanaryName`.

getStepDurationMetric()

Retorna se o canário emitirá ou não uma métrica `Duration` com a dimensão `CanaryName` para esse canário.

getStepSuccessMetric()

Retorna se o canário emitirá ou não uma métrica `StepSuccess` com a dimensão `CanaryName` para esse canário.

with2xxMetric(_2xxMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `2xx` com a dimensão `CanaryName` para esse canário.

with4xxMetric(_4xxMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `4xx` com a dimensão `CanaryName` para esse canário.

with5xxMetric(_5xxMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `5xx` com a dimensão `CanaryName` para esse canário.

withAggregated2xxMetric(aggregated2xxMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `2xx` sem dimensão para esse canário.

withAggregated4xxMetric(aggregated4xxMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `4xx` sem dimensão para esse canário.

withAggregated5xxMetric(aggregated5xxMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `5xx` sem dimensão para esse canário.

withAggregatedFailedCanaryMetric(aggregatedFailedCanaryMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `Failed` sem dimensão para esse canário.

withAggregatedFailedRequestsMetric(aggregatedFailedRequestsMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `Failed requests` sem dimensão para esse canário.

withFailedCanaryMetric(failedCanaryMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `Failed` com a dimensão `CanaryName` para esse canário.

withFailedRequestsMetric(failedRequestsMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `Failed requests` com a dimensão `CanaryName` para esse canário.

withStepDurationMetric(stepDurationMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `Duration` com a dimensão `CanaryName` para esse canário.

withStepSuccessMetric(stepSuccessMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `StepSuccess` com a dimensão `CanaryName` para esse canário.

Métodos para habilitar ou desabilitar outros recursos

withHarFile()

Aceita um argumento booliano, que especifica se deverá ou não ser criado um arquivo HAR para esse canário.

withStepsReport()

Aceita um argumento booliano, que especifica se deverá ou não ser emitido um relatório do resumo de execução de etapas para esse canário.

withIncludeUrlPassword()

Aceita um argumento booliano, que especifica se as senhas que aparecem nas URLs em logs e relatórios serão incluídas ou não.

withRestrictedUrlParameters()

Aceira uma matriz de parâmetros de caminho de URL ou consulta a serem editados. Aplica-se a URLs que aparecem em logs, relatórios e erros. É possível passar um asterisco (*) como um valor para editar todos os valores de parâmetro de consulta e caminho de URL

withLogRequest()

Aceita um argumento booliano, que especifica se cada solicitação deverá ou não ser registrada nos logs do canário.

withLogResponse()

Aceita um argumento booliano, que especifica se cada resposta deverá ou não ser registrada nos logs do canário.

withLogRequestBody()

Aceita um argumento booliano, que especifica se cada corpo da solicitação deverá ou não ser registrado nos logs do canário.

withLogResponseBody()

Aceita um argumento booliano, que especifica se cada corpo da resposta deverá ou não ser registrado nos logs do canário.

withLogRequestHeaders()

Aceita um argumento booliano, que especifica se cada cabeçalho de solicitação deverá ou não ser registrado nos logs do canário.

withLogResponseHeaders()

Aceita um argumento booliano, que especifica se cada cabeçalho de resposta deverá ou não ser registrado nos logs do canário.

getHarFile()

Retorna se o canário criará ou não um arquivo HAR.

getStepsReport()

Retorna se o canário emitirá ou não um relatório do resumo de execução de etapa.

getIncludeUrlPassword()

Retorna se o canário incluirá ou não as senhas que aparecem nas URLs em logs e relatórios.

getRestrictedUrlParameters()

Retorna se o canário editará ou não o caminho da URL ou os parâmetros de consulta.

getLogRequest()

Retorna se o canário registrará ou não cada solicitação nos logs do canário.

getLogResponse()

Retorna se o canário registrará ou não cada resposta nos logs do canário.

getLogRequestBody()

Retorna se o canário registrará ou não cada corpo da solicitação nos logs do canário.

getLogResponseBody()

Retorna se o canário registrará ou não cada corpo da resposta nos logs do canário.

getLogRequestHeaders()

Retorna se o canário registrará ou não cada cabeçalho de solicitação nos logs do canário.

getLogResponseHeaders()

Retorna se o canário registrará ou não cada cabeçalho de resposta nos logs do canário.

Funções para todos os canaries

- `withIncludeRequestHeaders`(includeRequestHeaders)
- `withIncludeResponseHeaders`(IncludeResponseHeaders)
- `withRestrictedHeaders`(restrictedHeaders)
- `withIncludeRequestBody`(includeRequestBody)
- `withIncludeResponseBody`(includeResponseBody)
- `enableReportingOptions`(): habilita todas as opções de relatórios (includeRequestBody, includeResponseHeaders, includeRequestBody e includeResponseBody).
- `disableReportingOptions`(): desabilita todas as opções de relatórios (includeRequestBody, includeResponseHeaders, includeRequestBody e includeResponseBody).

setConfig(options) para os canaries de interface do usuário

Para os canaries de interface do usuário, setConfig pode incluir os seguintes parâmetros boolianos:

- `continueOnStepFailure` (booliano): se continuará ou não executando o script do canário após a falha de uma etapa (isso se refere à função executeStep). Se alguma etapa falhar, a execução do canário ainda será marcada como falha. O padrão é `false`.
- `harFile` (booliano): se criará ou não um arquivo HAR. O padrão é `True`.
- `screenshotOnStepStart` (booliano): se fará ou nãi uma captura de tela antes de iniciar uma etapa.
- `screenshotOnStepSuccess` (booliano): se fará ou não uma captura de tela depois de concluir uma etapa bem-sucedida.
- `screenshotOnStepFailure` (booliano): se fará ou não uma captura de tela depois de que uma etapa falhar.

Métodos para habilitar ou desabilitar capturas de tela

disableStepScreenshots()

Desabilita todas as opções de captura de tela (screenshotOnStepStart, screenshotOnStepSuccess e screenshotOnStepFailure).

enableStepScreenshots()

Habilita todas as opções de captura de tela (screenshotOnStepStart, screenshotOnStepSuccess e screenshotOnStepFailure). Por padrão, todas essas métricas são permitidas.

getScreenshotOnStepFailure()

Retorna se o canário fará ou não uma captura de tela depois que uma etapa falhar.

getScreenshotOnStepStart()

Retorna se o canário fará ou não uma captura de tela antes de iniciar uma etapa.

getScreenshotOnStepSuccess()

Retorna se o canário fará ou não uma captura de tela depois que uma etapa for concluída.

withScreenshotOnStepStart(screenshotOnStepStart)

Aceita um argumento booliano, que indica se uma captura de tela deverá ou não ser obtida antes de iniciar uma etapa.

withScreenshotOnStepSuccess(screenshotOnStepSuccess)

Aceita um argumento booliano, que indica se uma captura de tela deverá ou não ser obtida após a conclusão de uma etapa.

withScreenshotOnStepFailure(screenshotOnStepFailure)

Aceita um argumento booliano, que indica se uma captura de tela deverá ou não ser obtida depois que uma etapa falhar.

Uso em canaries de interface do usuário

Primeiro, importe a dependência do Synthetics e busque a configuração.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();
```

Em seguida, defina a configuração para cada opção chamando o método SetConfig usando uma das opções a seguir.

```
// Set configuration values
    synConfig.setConfig({
        screenshotOnStepStart: true,
        screenshotOnStepSuccess: false,
        screenshotOnStepFailure: false
    });
```

Ou

```
synConfig.withScreenshotOnStepStart(false).withScreenshotOnStepSuccess(true).withScreenshotOnSt
```

Para desabilitar todas as capturas de tela, use a função disableStepScreenshots() como neste exemplo.

```
synConfig.disableStepScreenshots();
```

É possível habilitar e desabilitar as capturas de tela em qualquer ponto do código. Por exemplo, para desabilitar capturas de tela apenas para uma etapa, desabilite-as antes de executar essa etapa e habilite-as após a etapa.

setConfig(options) para canaries de API

Para os canaries de API, o setConfig pode incluir os seguintes parâmetros boolianos:

- `continueOnHttpStepFailure` (booliano): se continuará ou não executando o script do canário após a falha de uma etapa HTTP (isso se refere à função executeHttpStep). Se alguma etapa falhar, a execução do canário ainda será marcada como falha. O padrão é `true`.

Monitoramento visual

O monitoramento visual compara capturas de tela feitas durante uma execução do canário com capturas de tela feitas durante uma execução do canário de linha de base. Se a discrepância entre as duas capturas de tela ultrapassar a porcentagem de limite, o canário falhará, e você poderá ver as áreas com diferenças destacadas na cor no relatório de execução do canário. O monitoramento visual é compatível com canaries que executam syn-puppeteer-node-3.2 e posterior. Atualmente não é compatível com canaries que executam Python e Selenium.

Para habilitar o monitoramento visual, adicione a seguinte linha de código ao script do canário. Para obter mais detalhes, consulte Classe SyntheticsConfiguration.

```
syntheticsConfiguration.withVisualCompareWithBaseRun(true);
```

A primeira vez que o canário é executado corretamente após essa linha ser adicionada ao script, ele usa as capturas de tela obtidas durante a execução como linha de base para comparação. Após a primeira execução do canário, é possível usar o console do CloudWatch para editar o canário para fazer qualquer um destes procedimentos:

- Defina a próxima execução do canário como a nova linha de base.
- Estabeleça limites na captura de tela de linha de base atual para designar as áreas da captura de tela que deverão ser ignoradas durante comparações visuais.
- Remova uma captura de tela que não será usada para monitoramento visual.

Para obter mais informações sobre como usar o console do CloudWatch para editar um canário, consulte Editar ou excluir um canário.

Outras opções para monitoramento visual

syntheticsConfiguration.withVisualVarianceThresholdPercentage(desiredPercentage)

Defina a porcentagem aceitável para a variação da captura de tela em comparações visuais.

syntheticsConfiguration.withVisualVarianceHighlightHexColor("#fafa00")

Defina a cor de realce que designa as áreas de variação ao examinar relatórios de execução do canário que usam monitoramento visual.

syntheticsConfiguration.withFailCanaryRunOnVisualVariance(failCanary)

Defina se o canário falha ou não quando há uma diferença visual que é maior do que o limite. O padrão é o canário falhar.

Synthetics Logger

O SyntheticsLogger grava logs no console e em um arquivo de log local no mesmo nível de log. Este arquivo de logs é gravado em ambos os locais apenas se o nível de registo estiver no nível de registo pretendido ou abaixo da função de registo que foi chamada.

As instruções de log no arquivo de log local são precedidas por "DEBUG: ", "INFO: " e assim por diante para corresponder ao nível de log da função que foi chamada.

Você pode usar o SyntheticsLogger presumindo que você deseja executar a Biblioteca do Synthetics no mesmo nível de log que seu log do canário do Synthetics.

Não é necessário usar o SyntheticsLogger para criar um arquivo de log que está carregando o local de resultados do S3. Em vez disso, você pode criar um arquivo de log diferente na pasta `/tmp`. Todos os arquivos criados sob a pasta `/tmp` são carregados para o local de resultados no S3 como artefatos.

Como usar o registrador da biblioteca do Synthetics:

```
const log = require('SyntheticsLogger');
```

Definições úteis de função:

log.debug (***message***, ***ex***);

Parâmetros: *message* é a mensagem a ser registrada e *ex* é a exceção, se houver, a ser registrada em log.

Exemplo:

```
log.debug("Starting step - login.");
```

log.error(***message***, ***ex***);

Parâmetros: *message* é a mensagem a ser registrada e *ex* é a exceção, se houver, a ser registrada em log.

Exemplo:

```
try {
  await login();
catch (ex) {
  log.error("Error encountered in step - login.", ex);
}
```

log.info(***message***, ***ex***);

Parâmetros: *message* é a mensagem a ser registrada e *ex* é a exceção, se houver, a ser registrada em log.

Exemplo:

```
log.info("Successfully completed step - login.");
```

log.log(***message***, ***ex***);

Este é um alias para `log.info`.

Parâmetros: *message* é a mensagem a ser registrada e *ex* é a exceção, se houver, a ser registrada em log.

Exemplo:

```
 log.log("Successfully completed step - login.");
```

log.warn(***message***, ***ex***);

Parâmetros: *message* é a mensagem a ser registrada e *ex* é a exceção, se houver, a ser registrada em log.

Exemplo:

```
log.warn("Exception encountered trying to publish CloudWatch Metric.", ex);
```

SyntheticsLogHelper class

A classe `SyntheticsLogHelper` está disponível no runtime `syn-nodejs-puppeteer-3.2` e em tempos de execução posteriores. Ela já foi inicializada na biblioteca do CloudWatch Synthetics e está definida com a configuração do Synthetics. É possível adicioná-la como uma dependência em seu script. Essa classe permite limpar URLs, cabeçalhos e mensagens de erro para editar informações sigilosas.

> **Note**
>
> O Synthetics limpa todas as URLs e mensagens de erro que registra antes de incluí-las em logs, relatórios, arquivos HAR e erros de execução do canário com base na configuração `restrictedUrlParameters` do Synthetics. Somente será necessário usar

`getSanitizedUrl` ou `getSanitizedErrorMessage` se você estiver registrando URLs ou erros em seu script. O Synthetics não armazena artefatos do canário, exceto erros do canário lançados pelo script. Artefatos de execução do canário são armazenados em sua conta do cliente. Para ter mais informações, consulte Considerações de segurança para canaries do Synthetics.

getSanitizedUrl(url, stepConfig = null)

Essa função está disponível em `syn-nodejs-puppeteer-3.2` e posteriores. Ela retorna strings de url limpas com base na configuração. Você pode optar por editar parâmetros de URL sigilosos, como password e access_token, definindo a propriedade `restrictedUrlParameters`. Por padrão, as senhas em URLs são editadas. É possível habilitar senhas de URL, se necessário, definindo `includeUrlPassword`como true.

Essa função lançará um erro se a URL passada não for uma URL válida.

Parâmetros

- *`url`*é uma string e é a URL a ser limpa.
- *`stepConfig`* (Opcional) substitui a configuração global de Synthetics para essa função. Se `stepConfig` não for passado, será usada a configuração global para limpar a URL.

Exemplo

Este exemplo usa a seguinte URL de exemplo: `https://example.com/learn/home?access_token=12345&token_type=Bearer&expires_in=1200`. Neste exemplo, `access_token` contém suas informações sigilosas que não devem ser registradas. Observe que os serviços do Synthetics não armazenam artefatos de execução do canário. Artefatos como logs, capturas de tela e relatórios são armazenados em um bucket do Amazon S3 em sua conta de cliente.

A primeira etapa é definir a configuração do Synthetics.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Import Synthetics logger for logging url
const log = require('SyntheticsLogger');
```

```
// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();

// Set restricted parameters
synConfig.setConfig({
   restrictedUrlParameters: ['access_token'];
});
```

Em seguida, limpar e registrar a URL

```
// Import SyntheticsLogHelper dependency
const syntheticsLogHelper = require('SyntheticsLogHelper');

const sanitizedUrl = synthetics.getSanitizedUrl('https://example.com/learn/home?
access_token=12345&token_type=Bearer&expires_in=1200');
```

Isso registra o seguinte em seu log do canário.

```
My example url is: https://example.com/learn/home?
access_token=REDACTED&token_type=Bearer&expires_in=1200
```

É possível substituir a configuração Synthetics para uma URL aprovando um parâmetro opcional que contenha opções de configuração Synthetics, como no exemplo a seguir.

```
const urlConfig = {
   restrictedUrlParameters = ['*']
};
const sanitizedUrl = synthetics.getSanitizedUrl('https://example.com/learn/home?
access_token=12345&token_type=Bearer&expires_in=1200', urlConfig);
logger.info('My example url is: ' + sanitizedUrl);
```

O exemplo anterior edita todos os parâmetros de consulta e é registrado desta forma:

```
My example url is: https://example.com/learn/home?
access_token=REDACTED&token_type=REDACTED&expires_in=REDACTED
```

getSanitizedErrorMessage

Essa função está disponível em `syn-nodejs-puppeteer-3.2` e posteriores. Retorna strings de erro limpas, limpando as URLs presentes com base na configuração do Synthetics. Você pode

escolher substituir a configuração global do Synthetics ao chamar essa função aprovando um parâmetro `stepConfig` opcional.

Parâmetros

- *`erro`* é o erro ao limpar. Pode ser um objeto Error ou uma string.
- *`stepConfig`* (Opcional) substitui a configuração global de Synthetics para essa função. Se `stepConfig` não for passado, será usada a configuração global para limpar a URL.

Exemplo

Este exemplo usa o seguinte erro: `Failed to load url: https://example.com/learn/home?access_token=12345&token_type=Bearer&expires_in=1200`

A primeira etapa é definir a configuração do Synthetics.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Import Synthetics logger for logging url
const log = require('SyntheticsLogger');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();

// Set restricted parameters
synConfig.setConfig({
   restrictedUrlParameters: ['access_token'];
});
```

Em seguida, limpe e registre a mensagem de erro

```
// Import SyntheticsLogHelper dependency
const syntheticsLogHelper = require('SyntheticsLogHelper');

try {
   // Your code which can throw an error containing url which your script logs
} catch (error) {
    const sanitizedErrorMessage = synthetics.getSanitizedErrorMessage(errorMessage);
    logger.info(sanitizedErrorMessage);
}
```

Isso registra o seguinte no log do canário.

```
Failed to load url: https://example.com/learn/home?
access_token=REDACTED&token_type=Bearer&expires_in=1200
```

getSanitizedHeaders(headers, stepConfig=null)

Essa função está disponível em `syn-nodejs-puppeteer-3.2` e posteriores. Retorna cabeçalhos limpos com base na propriedade `restrictedHeaders` de `syntheticsConfiguration`. Os cabeçalhos especificados na propriedade `restrictedHeaders` são editados a partir de logs, arquivos HAR e relatórios.

Parâmetros

- *`headers`* é um objeto que contém os cabeçalhos a serem limpos.
- *`stepConfig`* (Opcional) substitui a configuração global de Synthetics para essa função. Se `stepConfig` não for passado, será usada a configuração global para limpar os cabeçalhos.

Classes de biblioteca do Node.js e funções que se aplicam somente a canaries de interface do usuário

As seguintes funções de biblioteca para Node.js do CloudWatch Synthetics são úteis apenas para canaries de interface do usuário.

Tópicos

- Classe Synthetics
- Classe BrokenLinkCheckerReport
- Classe SyntheticsLink

Classe Synthetics

As funções a seguir estão na classe Synthetics.

async addUserAgent(page, userAgentString);

Esta função acrescenta *`userAgentString`* ao cabeçalho "User-Agent" da página especificada.

Exemplo:

```
await synthetics.addUserAgent(page, "MyApp-1.0");
```

Resulta no cabeçalho "User-Agent" da página que está sendo definido como *browsers-user-agent-header-value*`MyApp-1.0`

async executeStep(stepName, functionToExecute, [stepConfig]);

Executa o passo fornecido, envolvendo-o com logs de iniciar/passar/falhar, capturas de tela de iniciar/passar/falhar e métricas de aprovação/reprovação e duração.

> **Note**
>
> Se você estiver usando `syn-nodejs-2.1` ou um runtime posterior, será possível configurar se as capturas de tela serão obtidas ou não e quando serão obtidas. Para ter mais informações, consulte Classe SyntheticsConfiguration.

A função `executeStep` também faz o seguinte:

- Registra que a etapa começou.
- Faz uma captura de tela chamada `<stepName>-starting`.
- Inicia um temporizador.
- Executa a função fornecida.
- Se a função retornar normalmente, ela contará como aprovada. Se a função apresentar erro, ela contará como falha.
- Encerra o temporizador.
- Registra se a etapa foi aprovada ou falhou
- Faz uma captura de tela chamada `<stepName>-succeeded` ou `<stepName>-failed`.
- Emite a métrica `stepName SuccessPercent`, 100 para aprovação ou 0 para falha.
- Emite a métrica `stepName Duration` com um valor de acordo com os horários inicial e final da etapa.
- Finalmente, retorna o que `functionToExecute` retornou ou lança novamente o que `functionToExecute` lançou.

Se o canário usar `syn-nodejs-2.0` ou um runtime posterior, essa função também adicionará um resumo de execução de etapa ao relatório do canário. O resumo contém detalhes sobre cada etapa,

como hora de início, hora de término, status (PASSED/FAILED), motivo da falha (se for o caso) e capturas de tela obtidas durante a execução de cada etapa.

Exemplo:

```
await synthetics.executeStep('navigateToUrl', async function (timeoutInMillis = 30000)
 {
           await page.goto(url, {waitUntil: ['load', 'networkidle0'], timeout:
 timeoutInMillis});});
```

Resposta:

Retorna o que `functionToExecute` retornou.

Atualizações com syn-nodejs-2.2

Começando com `syn-nodejs-2.2`, você pode, opcionalmente, passar configurações de etapa para substituir as configurações do CloudWatch Synthetics no nível de etapa. Para obter uma lista de opções que você pode passar para `executeStep`, consulte Classe SyntheticsConfiguration.

O exemplo a seguir substitui a configuração padrão `false` de `continueOnStepFailure` para `true` e especifica quando obter capturas de tela.

```
var stepConfig = {
    'continueOnStepFailure': true,
    'screenshotOnStepStart': false,
    'screenshotOnStepSuccess': true,
    'screenshotOnStepFailure': false
}

await executeStep('Navigate to amazon', async function (timeoutInMillis = 30000) {
      await page.goto(url, {waitUntil: ['load', 'networkidle0'], timeout:
 timeoutInMillis});
 }, stepConfig);
```

getDefaultLaunchOptions();

A função `getDefaultLaunchOptions()` retorna as opções de inicialização do navegador que serão usadas pelo CloudWatch Synthetics. Para obter mais informações, consulte Tipos de opções de lançamento

```
// This function returns default launch options used by Synthetics.
```

```
const defaultOptions = await synthetics.getDefaultLaunchOptions();
```

getPage();

Retorna a página aberta atual como um objeto Puppeteer. Para obter mais informações, consulte Puppeteer API v1.14.0.

Exemplo:

```
let page = synthetics.getPage();
```

Resposta:

A página (objeto Puppeteer) que está aberta atualmente na sessão atual do navegador.

getRequestResponseLogHelper();

> **Important**
>
> Em canários que usam o runtime `syn-nodejs-puppeteer-3.2` ou posteriores, essa função está descontinuada junto com a classe `RequestResponseLogHelper`. Qualquer uso desta função faz exibe um aviso em seus registros do canário. Essa função será removida em futuras versões do runtime. Se você estiver usando esta função, substitua por RequestResponseLogHelper class.

Use essa função como padrão de criador para ajustar os sinalizadores de log de solicitação e resposta.

Exemplo:

```
synthetics.setRequestResponseLogHelper(getRequestResponseLogHelper().withLogRequestHeaders(fals
```

Resposta:

```
{RequestResponseLogHelper}
```

launch(options)

As opções para essa função estão disponíveis apenas na versão do runtime `syn-nodejs-2.1` ou posteriores.

Essa função é usada apenas para canaries de interface do usuário. Ela fecha o navegador existente e inicia um novo.

> **Note**
>
> O CloudWatch Synthetics sempre inicia um navegador antes de começar a executar o script. Não é necessário chamar launch(), a menos que você queira iniciar um novo navegador com opções personalizadas.

(options) é um conjunto configurável de opções a serem definidas no navegador. Para obter mais informações, consulte Tipos de opções de lançamento.

Se você chamar esta função sem opções, o Synthetics iniciará um navegador com argumentos padrão, `executablePath` e `defaultViewport`. O visor padrão no CloudWatch Synthetics é 1920 por 1080.

É possível substituir os parâmetros de inicialização usados pelo CloudWatch Synthetics e passar outros parâmetros ao iniciar o navegador. Por exemplo, o trecho de código a seguir inicia um navegador com argumentos padrão e um caminho executável padrão, mas com um visor de 800 x 600.

```
await synthetics.launch({
        defaultViewport: {
            "deviceScaleFactor": 1,
            "width": 800,
            "height": 600
    }});
```

O código de exemplo a seguir adiciona um novo parâmetro `ignoreHTTPSErrors`para os parâmetros de inicialização do CloudWatch Synthetics:

```
await synthetics.launch({
        ignoreHTTPSErrors: true
 });
```

É possível desabilitar a segurança da Web adicionando uma sinalização `--disable-web-security` para args nos parâmetros de inicialização do CloudWatch Synthetics:

```
// This function adds the --disable-web-security flag to the launch parameters
```

```
const defaultOptions = await synthetics.getDefaultLaunchOptions();
const launchArgs = [...defaultOptions.args, '--disable-web-security'];
await synthetics.launch({
     args: launchArgs
  });
```

RequestResponseLogHelper class

> **Important**
>
> Em canários que usam o runtime `syn-nodejs-puppeteer-3.2` ou posteriores, essa classe está defasada. Qualquer uso dessa classe exibe um aviso em seus registros do canário. Essa função será removida em futuras versões do runtime Se você estiver usando esta função, substitua por RequestResponseLogHelper class.

Lida com a configuração minuciosa e a criação de representações de string de cargas úteis de solicitação e resposta.

```
class RequestResponseLogHelper {

    constructor () {
        this.request = {url: true, resourceType: false, method: false, headers: false,
 postData: false};
        this.response = {status: true, statusText: true, url: true, remoteAddress:
 false, headers: false};
    }

    withLogRequestUrl(logRequestUrl);

    withLogRequestResourceType(logRequestResourceType);

    withLogRequestMethod(logRequestMethod);

    withLogRequestHeaders(logRequestHeaders);

    withLogRequestPostData(logRequestPostData);


    withLogResponseStatus(logResponseStatus);

    withLogResponseStatusText(logResponseStatusText);
```

```
    withLogResponseUrl(logResponseUrl);

    withLogResponseRemoteAddress(logResponseRemoteAddress);

    withLogResponseHeaders(logResponseHeaders);
```

Exemplo:

```
synthetics.setRequestResponseLogHelper(getRequestResponseLogHelper()
.withLogRequestPostData(true)
.withLogRequestHeaders(true)
.withLogResponseHeaders(true));
```

Resposta:

```
{RequestResponseLogHelper}
```

setRequestResponseLogHelper();

> **Important**
>
> Em canaries que usam o runtime `syn-nodejs-puppeteer-3.2` ou posteriores, essa função está defasada junto com a classe `RequestResponseLogHelper`. Qualquer uso desta função faz exibe um aviso em seus registros do canário. Essa função será removida em futuras versões do runtime. Se você estiver usando esta função, substitua por RequestResponseLogHelper class.

Use essa função como padrão de criador para ajustar os marcadores de log de solicitação e resposta.

Exemplo:

```
synthetics.setRequestResponseLogHelper().withLogRequestHeaders(true).withLogResponseHeaders(tru
```

Resposta:

```
{RequestResponseLogHelper}
```

async takeScreenshot(name, suffix);

Faz uma captura de tela (.PNG) da página atual com o some e sufixo (opcional).

Exemplo:

```
await synthetics.takeScreenshot("navigateToUrl", "loaded")
```

Este exemplo captura e carrega uma captura de tela chamada `01-navigateToUrl-loaded.png`para o bucket do S3 do canário.

É possível fazer uma captura de tela para uma etapa do canário específica aprovando `stepName` como o primeiro parâmetro. As capturas de tela são vinculadas à etapa do canário em seus relatórios, para ajudar você a rastrear cada etapa durante a depuração.

Os canários do CloudWatch Synthetics fazem capturas de tela automaticamente antes de iniciar uma etapa (a função `executeStep`) e após a conclusão da etapa (a menos que você configure o canário para desabilitar capturas de tela). É possível fazer mais capturas de tela aprovando o nome da etapa na função `takeScreenshot`.

O exemplo a seguir faz uma captura de tela com `signupForm` como o valor de `stepName`. A captura de tela será chamada `02-signupForm-address` e será vinculada à etapa chamada `signupForm` no relatório do canário.

```
await synthetics.takeScreenshot('signupForm', 'address')
```

Classe BrokenLinkCheckerReport

Essa classe fornece métodos para adicionar um link sintético. É compatível apenas em canaries que usam a versão `syn-nodejs-2.0-beta` do runtime ou posteriores.

Para usar `BrokenLinkCheckerReport`, inclua as seguintes linhas no script:

```
const BrokenLinkCheckerReport = require('BrokenLinkCheckerReport');

const brokenLinkCheckerReport = new BrokenLinkCheckerReport();
```

Definições úteis de função:

addLink(***syntheticsLink***, isBroken)

`syntheticsLink` é um objeto `SyntheticsLink` representando um link. Essa função adiciona o link de acordo com o código de status. Por padrão, considera um link a ser quebrado se o código de status não estiver disponível ou se o código de status for 400 ou superior. Você pode substituir esse comportamento padrão aprovando o parâmetro opcional `isBrokenLink` com um valor de `true` ou `false`.

Essa função não retorna valor.

getLinks()

Essa função retorna uma matriz de objetos `SyntheticsLink` que estão incluídos no relatório do verificador de links quebrados.

getTotalBrokenLinks()

Essa função retorna um número que representa o total de links quebrados.

getTotalLinksChecked()

Essa função retorna um número que representa o total de links incluídos no relatório.

Como usar BrokenLinkCheckerReport

O trecho de código de script do canário a seguir demonstra um exemplo de como navegar para um link e adicioná-lo ao relatório do verificador de links quebrados.

1. Importar `SyntheticsLink`, `BrokenLinkCheckerReport` e `Synthetics`.

```
const BrokenLinkCheckerReport = require('BrokenLinkCheckerReport');
const SyntheticsLink = require('SyntheticsLink');

// Synthetics dependency
const synthetics = require('Synthetics');
```

2. Para adicionar um link ao relatório, crie uma instância de `BrokenLinkCheckerReport`.

```
let brokenLinkCheckerReport = new BrokenLinkCheckerReport();
```

3. Navegue até a URL e adicione-a ao relatório do verificador de links quebrados.

```
let url = "https://amazon.com";

let syntheticsLink = new SyntheticsLink(url);
```

```
// Navigate to the url.
let page = await synthetics.getPage();

// Create a new instance of Synthetics Link
let link = new SyntheticsLink(url)

try {
    const response = await page.goto(url, {waitUntil: 'domcontentloaded', timeout:
 30000});
} catch (ex) {
    // Add failure reason if navigation fails.
    link.withFailureReason(ex);
}

if (response) {
    // Capture screenshot of destination page
    let screenshotResult = await synthetics.takeScreenshot('amazon-home', 'loaded');

    // Add screenshot result to synthetics link
    link.addScreenshotResult(screenshotResult);

    // Add status code and status description to the link
    link.withStatusCode(response.status()).withStatusText(response.statusText())
}

// Add link to broken link checker report.
brokenLinkCheckerReport.addLink(link);
```

4. Adicione o relatório ao Synthetics. Isso criará um arquivo JSON chamado `BrokenLinkCheckerReport.json` em seu bucket do S3 para cada execução do canário. Será possível ver um relatório de links no console para cada execução do canário, além de capturas de tela, logs e arquivos HAR.

```
await synthetics.addReport(brokenLinkCheckerReport);
```

Classe SyntheticsLink

Essa classe fornece métodos para quebrar informações. É compatível apenas em canários que usam a versão `syn-nodejs-2.0-beta` do runtime ou posteriores.

Para usar `SyntheticsLink`, inclua as seguintes linhas no script:

```
const SyntheticsLink = require('SyntheticsLink');

const syntheticsLink = new SyntheticsLink("https://www.amazon.com");
```

Essa função retorna `syntheticsLink`*`Object`*

Definições úteis de função:

withUrl(***`url`***)

`url` é uma string de URL. Essa função retorna `syntheticsLink`*`Object`*

withText(***`text`***)

`text` é uma string que representa o texto de ancoragem. Essa função retorna `syntheticsLink`*`Object`*. Adiciona texto de ancoragem correspondente ao link.

withParentUrl(***`parentUrl`***)

`parentUrl` é uma string que representa a URL mãe (página de origem). Essa função retorna `syntheticsLink`*`Object`*

withStatusCode(***`statusCode`***)

`statusCode` é uma string que representa o código de status. Essa função retorna `syntheticsLink`*`Object`*

withFailureReason(***`failureReason`***)

`failureReason` é uma string que representa o motivo da falha. Essa função retorna `syntheticsLink`*`Object`*

addScreenshotResult(***`screenshotResult`***)

`screenshotResult` é um objeto. É uma instância de `ScreenshotResult` que foi retornada pela função `takeScreenshot` do Synthetics. O objeto inclui o seguinte:

- `fileName`: uma string que representa `screenshotFileName`
- `pageUrl` (opcional)
- `error` (Opcional)

Classes de biblioteca do Node.js e funções que se aplicam apenas a canaries de API

As seguintes funções de biblioteca para Node.js do CloudWatch Synthetics são úteis apenas para canaries de API.

Tópicos

- executeHttpStep(stepName, requestOptions, [callback], [stepConfig])

executeHttpStep(stepName, requestOptions, [callback], [stepConfig])

Executa a solicitação HTTP fornecida como uma etapa e publica `SuccessPercent` (aprovação/falha) e métricas `Duration`.

executeHttpStep usa funções nativas HTTP ou HTTPS nos bastidores, conforme o protocolo especificado na solicitação.

Essa função também adicionará um resumo de execução de etapa ao relatório do canário. O resumo contém detalhes sobre cada solicitação HTTP, como o seguinte:

- Horário de início
- End Time
- Status (PASSED/FAILED)
- Motivo da falha, se for o caso
- Detalhes da chamada HTTP, como cabeçalhos e corpo da solicitação/resposta, código de status, mensagem de status e tempos de performance.

Parâmetros

stepName(***`String`***)

Especifica o nome da etapa. Esse nome também é usado para publicar métricas do CloudWatch para essa etapa.

requestOptions(***`Object or String`***)

O valor desse parâmetro poderá ser uma URL, uma string de URL ou um objeto. Se for um objeto, ele deverá ser um conjunto de opções configuráveis para fazer uma solicitação HTTP. É compatível com todas as opções em http.request(options[, callback]) na documentação do Node.js.

Além dessas opções do Node.js, requestOptions oferece suporte ao parâmetro adicional body. Você pode usar o parâmetro body para aprovar dados como um corpo da solicitação.

callback(***response***)

(Opcional) Esaa é uma função de usuário que é invocada com a resposta HTTP. A resposta é do tipo Class: http.IncomingMessage.

stepConfig(***object***)

(Opcional) Use esae parâmetro para substituir configurações globais do Synthetics por uma configuração diferente para essa etapa.

Exemplos de uso do executeHttpStep

As séries de exemplos a seguir são desenvolvidas entre si para ilustrar os vários usos dessa opção.

Este primeiro exemplo configura parâmetros de solicitação. Você pode aprovar uma URL como requestOptions:

```
let requestOptions = 'https://www.amazon.com';
```

Ou pode aprovar um conjunto de opções:

```
let requestOptions = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/product/validProductName',
        'port': 443,
        'protocol': 'https:'
    };
```

O próximo exemplo cria uma função de retorno de chamada que aceita uma resposta. Por padrão, se você não especificar callback, o CloudWatch Synthetics validará que o status está entre 200 e 299, inclusive.

```
// Handle validation for positive scenario
    const callback = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 200 || res.statusCode > 299) {
                throw res.statusCode + ' ' + res.statusMessage;
            }
```

```
            let responseBody = '';
            res.on('data', (d) => {
                responseBody += d;
            });

            res.on('end', () => {
                // Add validation on 'responseBody' here if required. For ex, your
 status code is 200 but data might be empty
                resolve();
            });
        });
    };
```

O próximo exemplo cria uma configuração para essa etapa que substitui a configuração global do CloudWatch Synthetics. A configuração de etapa desse exemplo permite cabeçalhos de solicitação, cabeçalhos de resposta, corpo da solicitação (dados de postagem) e corpo da resposta em seu relatório e restringir valores de cabeçalho ‘X-Amz-Security-Token’ e ‘Authorization’. Por padrão, esses valores não são incluídos no relatório por motivos de segurança. Se você escolher incluí-los, os dados serão armazenados apenas no bucket do S3.

```
// By default headers, post data, and response body are not included in the report for
 security reasons.
// Change the configuration at global level or add as step configuration for individual
 steps
let stepConfig = {
    includeRequestHeaders: true,
    includeResponseHeaders: true,
    restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted header
 values do not appear in report generated.
    includeRequestBody: true,
    includeResponseBody: true
};
```

Este exemplo final aprova sua solicitação para executeHttpStep e nomeia a etapa.

```
await synthetics.executeHttpStep('Verify GET products API', requestOptions, callback,
 stepConfig);
```

Com esse conjunto de exemplos, o CloudWatch Synthetics adiciona os detalhes de cada etapa do relatório e produz métricas para cada etapa usando stepName.

Você verá as métricas `successPercent` e `duration` para a etapa `Verify GET products API`. É possível monitorar a performance de sua API monitorando as métricas para suas etapas de chamada de API.

Para obter um script completo de exemplo que usa essas funções, consulte Canário de API de várias etapas.

Funções da biblioteca disponíveis para scripts do canário do Python usando Selenium

Esta seção lista as funções de biblioteca do Selenium disponíveis para scripts do canário Python.

Tópicos

- Classes e funções da biblioteca Python e Selenium que se aplicam a todos os canaries
- Classes e funções de biblioteca Python e Selenium que se aplicam apenas a canaries de interface do usuário

Classes e funções da biblioteca Python e Selenium que se aplicam a todos os canaries

As seguintes funções de biblioteca do Selenium do CloudWatch Synthetics para Python são úteis para todos os canaries.

Tópicos

- Classe SyntheticsConfiguration
- Classe SyntheticsLogger

Classe SyntheticsConfiguration

É possível usar a classe SyntheticsConfiguration para configurar o comportamento das funções da biblioteca do Synthetics. Por exemplo, você pode usar essa classe para configurar a função `executeStep()` para não obter capturas de tela.

É possível definir as configurações do CloudWatch Synthetics no nível global.

Definições de função:

set_config(options)

```
from aws_synthetics.common import synthetics_configuration
```

options é um objeto, que é um conjunto de opções configuráveis para seu canário. As seções a seguir explicam os campos possíveis em *options*.

- screenshot_on_step_start (booliano): se fará ou nãi uma captura de tela antes de iniciar uma etapa.
- screenshot_on_step_success (booliano): se fará ou não uma captura de tela depois de concluir uma etapa bem-sucedida.
- screenshot_on_step_failure (booliano): se fará ou não uma captura de tela depois de que uma etapa falhar.

with_screenshot_on_step_start(screenshot_on_step_start)

Aceita um argumento booliano, que indica se uma captura de tela deverá ou não ser obtida antes de iniciar uma etapa.

with_screenshot_on_step_success(screenshot_on_step_success)

Aceita um argumento booliano, que indica se uma captura de tela deverá ou não ser obtida após a conclusão de uma etapa.

with_screenshot_on_step_failure(screenshot_on_step_failure)

Aceita um argumento booliano, que indica se uma captura de tela deverá ou não ser obtida depois que uma etapa falhar.

get_screenshot_on_step_start()

Retorna se uma captura de tela será ou não obtida antes de iniciar uma etapa.

get_screenshot_on_step_success()

Retorna se uma captura de telam será ou não obtida depois que uma etapa for concluída.

get_screenshot_on_step_failure()

Retorna se uma captura de tela será ou não obtida depois que uma etapa falhar.

disable_step_screenshots()

Desabilita todas as opções de captura de tela (get_screenshot_on_step_start, get_screenshot_on_step_success e get_screenshot_on_step_failure).

enable_step_screenshots()

Habilita todas as opções de captura de tela (get_screenshot_on_step_start, get_screenshot_on_step_success e get_screenshot_on_step_failure). Por padrão, todas essas métricas são permitidas.

setConfig(options) em relação a métricas do CloudWatch

Para canários que usam `syn-python-selenium-1.1` ou posteriores, (options) para setConfig pode incluir os seguintes parâmetros boolianos que determinam quais métricas serão publicadas pelo canário. O padrão para cada uma dessas opções é `true`. As opções que começam com `aggregated` determinam se a métrica será emitida sem a dimensão `CanaryName`. É possível usar essas métricas para ver os resultados agregados de todos os seus canaries. As outras opções determinam se a métrica será emitida com a dimensão `CanaryName`. Você pode usar essas métricas para ver os resultados de cada canário individual.

Para obter uma lista de métricas do CloudWatch emitidas por canaries, consulte Métricas do CloudWatch publicadas por canaries.

- `failed_canary_metric` (booliano): se deverá ou não emitir a métrica `Failed` (com a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `failed_requests_metric` (booliano): se deverá ou não emitir a métrica `Failed requests` (com a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `2xx_metric` (booliano): se deverá ou não emitir a métrica `2xx` (com a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `4xx_metric` (booliano): se deverá ou não emitir a métrica `4xx` (com a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `5xx_metric` (booliano): se deverá ou não emitir a métrica `5xx` (com a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `step_duration_metric` (booliano): se deverá ou não emitir a métrica `Step duration` (com as dimensões `CanaryName StepName`) para esse canário. O padrão é `true`.
- `step_success_metric` (booliano): se deverá ou não emitir a métrica `Step success` (com as dimensões `CanaryName StepName`) para esse canário. O padrão é `true`.
- `aggregated_failed_canary_metric` (booliano): se deverá ou não emitir a métrica `Failed` (sem a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `aggregated_failed_requests_metric` (booliano): se deverá ou não emitir a métrica `Failed Requests` (sem a dimensão `CanaryName`) para esse canário. O padrão é `true`.

- `aggregated_2xx_metric` (booliano): se deverá ou não emitir a métrica `2xx` (sem a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `aggregated_4xx_metric` (booliano): se deverá ou não emitir a métrica `4xx` (sem a dimensão `CanaryName`) para esse canário. O padrão é `true`.
- `aggregated_5xx_metric` (booliano): se deverá ou não emitir a métrica `5xx` (sem a dimensão `CanaryName`) para esse canário. O padrão é `true`.

with_2xx_metric(2xx_metric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `2xx` com a dimensão `CanaryName` para esse canário.

with_4xx_metric(4xx_metric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `4xx` com a dimensão `CanaryName` para esse canário.

with_5xx_metric(5xx_metric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `5xx` com a dimensão `CanaryName` para esse canário.

withAggregated2xxMetric(aggregated2xxMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `2xx` sem dimensão para esse canário.

withAggregated4xxMetric(aggregated4xxMetric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `4xx` sem dimensão para esse canário.

with_aggregated_5xx_metric(aggregated_5xx_metric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `5xx` sem dimensão para esse canário.

with_aggregated_failed_canary_metric(aggregated_failed_canary_metric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `Failed` sem dimensão para esse canário.

with_aggregated_failed_requests_metric(aggregated_failed_requests_metric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `Failed requests` sem dimensão para esse canário.

with_failed_canary_metric(failed_canary_metric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `Failed` com a dimensão `CanaryName` para esse canário.

with_failed_requests_metric(failed_requests_metric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `Failed requests` com a dimensão `CanaryName` para esse canário.

with_step_duration_metric(step_duration_metric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `Duration` com a dimensão `CanaryName` para esse canário.

with_step_success_metric(step_success_metric)

Aceita um argumento booliano, que especifica se deverá emitir ou não uma métrica `StepSuccess` com a dimensão `CanaryName` para esse canário.

Métodos para habilitar ou desabilitar métricas

disable_aggregated_request_metrics()

Não permite que o canário emita todas as métricas de solicitação emitidas sem a dimensão `CanaryName`.

disable_request_metrics()

Desabilita todas as métricas de solicitação, inclusive métricas por canário e métricas agregadas em todos os canários.

disable_step_metrics()

Desativa todas as métricas de etapa, incluindo métricas de sucesso e de duração da etapa.

enable_aggregated_request_metrics()

Habilita o canário a emitir todas as métricas de solicitação emitidas sem a dimensão `CanaryName`.

enable_request_metrics()

Habilita todas as métricas de solicitação, inclusive métricas por canário e métricas agregadas em todos os canários.

enable_step_metrics()

Ativa todas as métricas de etapa, incluindo métricas de sucesso e métricas de duração da etapa.

Uso em canaries de interface do usuário

Primeiro, importe a dependência do Synthetics e busque a configuração. Em seguida, defina a configuração para cada opção chamando o método SetConfig usando uma das opções a seguir.

```
from aws_synthetics.common import synthetics_configuration

synthetics_configuration.set_config(
     {
        "screenshot_on_step_start": False,
        "screenshot_on_step_success": False,
        "screenshot_on_step_failure": True
     }
)

or
```

Ou

```
synthetics_configuration.with_screenshot_on_step_start(False).with_screenshot_on_step_success(F
```

Para desabilitar todas as capturas de tela, use a função disableStepScreenshots() como neste exemplo.

```
synthetics_configuration.disable_step_screenshots()
```

É possível habilitar e desabilitar as capturas de tela em qualquer ponto do código. Por exemplo, para desabilitar capturas de tela apenas para uma etapa, desabilite-as antes de executar essa etapa e habilite-as após a etapa.

set_config(options) para canaries de interface do usuário

Começando com `syn-python-selenium-1.1`, para os canaries de interface do usuário, `set_config` pode incluir os seguintes parâmetros boolianos:

- `continue_on_step_failure` (booliano): se continuará ou não executando o script do canário após a falha de uma etapa (isso se refere à função executeStep). Se alguma etapa falhar, a execução do canário ainda será marcada como falha. O padrão é `false`.

Classe SyntheticsLogger

O `synthetics_logger` grava logs no console e em um arquivo de log local no mesmo nível de log. Este arquivo de logs é gravado em ambos os locais apenas se o nível de registo estiver no nível de registo pretendido ou abaixo da função de registo que foi chamada.

As instruções de log no arquivo de log local são precedidas por "DEBUG: ", "INFO: " e assim por diante para corresponder ao nível de log da função que foi chamada.

Não é necessário usar o `synthetics_logger` para criar um arquivo de log que está carregando o local de resultados do Amazon S3. Em vez disso, você pode criar um arquivo de log diferente na pasta `/tmp`. Todos os arquivos criados sob a pasta `/tmp` são carregados para o local de resultados no bucket do S3 como artefatos.

Para usar `synthetics_logger`:

```
from aws_synthetics.common import synthetics_logger
```

Definições úteis de função:

Obter nível do log:

```
log_level = synthetics_logger.get_level()
```

Definir nível do log:

```
synthetics_logger.set_level()
```

Registre uma mensagem com um nível especificado. O nível pode ser `DEBUG`, `INFO`, `WARN` ou `ERROR`, como nos exemplos de sintaxe a seguir:

```
synthetics_logger.debug(message, *args, **kwargs)
```

```
synthetics_logger.info(message, *args, **kwargs)
```

```
synthetics_logger.log(message, *args, **kwargs)
```

```
synthetics_logger.warn(message, *args, **kwargs)
```

```
synthetics_logger.error(message, *args, **kwargs)
```

Para obter informações sobre parâmetros de depuração, consulte a documentação padrão do Python em logging.debug

Nessas funções de log, `message` é a string do formato da mensagem. `args` são os argumentos que são mesclados em `msg` usando o operador de formatação de string.

Há três argumentos de palavra-chave em `kwargs`:

- `exc_info`: se não for avaliado como false, adicionará informações de exceção à mensagem de log.
- `stack_info`: o padrão é false. Se true, adicionará informações de pilha à mensagem de log, incluindo a chamada de log real.
- `extra`: o terceiro argumento opcional de palavra-chave, que você pode usar para aprovar um dicionário que é usado para preencher o `__dict__` do `LogRecord` criado para o evento de log com atributos definidos pelo usuário.

Exemplos:

Registrar uma mensagem com o nível `DEBUG`:

```
synthetics_logger.debug('Starting step - login.')
```

Registrar uma mensagem com o nível `INFO`. `logger.log` é sinônimo de `logger.info`:

```
synthetics_logger.info('Successfully completed step - login.')
```

ou

```
synthetics_logger.log('Successfully completed step - login.')
```

Registrar uma mensagem com o nível `WARN`:

```
synthetics_logger.warn('Warning encountered trying to publish %s', 'CloudWatch Metric')
```

Registrar uma mensagem com o nível `ERROR`:

```
synthetics_logger.error('Error encountered trying to publish %s', 'CloudWatch Metric')
```

Registre uma exceção:

```
synthetics_logger.exception(message, *args, **kwargs)
```

Regista uma mensagem com o nível `ERROR`. Informações de exceção são adicionadas à mensagem de log. Você deve chamar essa função apenas a partir de um manipulador de exceção.

Para obter informações sobre parâmetros de exceção, consulte a documentação padrão do Python em logging.exception

A `message` é a string de formato da mensagem. `args` são os argumentos, que são mesclados em `msg` usando o operador de formatação de string.

Há três argumentos de palavra-chave em `kwargs`:

- `exc_info`: se não for avaliado como false, adicionará informações de exceção à mensagem de log.
- `stack_info`: o padrão é false. Se true, adicionará informações de pilha à mensagem de log, incluindo a chamada de log real.
- `extra`: o terceiro argumento opcional de palavra-chave, que você pode usar para aprovar um dicionário que é usado para preencher o `__dict__` do `LogRecord` criado para o evento de log com atributos definidos pelo usuário.

Exemplo:

```
synthetics_logger.exception('Error encountered trying to publish %s', 'CloudWatch
 Metric')
```

Classes e funções de biblioteca Python e Selenium que se aplicam apenas a canaries de interface do usuário

As seguintes funções de biblioteca do Selenium do CloudWatch Synthetics para Python são úteis somente para canaries de interface do usuário.

Tópicos

Classe SyntheticsBrowser

Quando você cria uma instância do navegador chamando `synthetics_webdriver.Chrome()`, a instância do navegador retornada é do tipo `SyntheticsBrowser`. A classe `SyntheticsBrowser` controla o ChromeDriver e permite que o script do canário conduza o navegador, permitindo que o Selenium WebDriver funcione com o Synthetics.

Além dos métodos padrão do Selenium, ele também fornece os métodos abaixo.

set_viewport_size(width, height)

Define o visor do navegador. Exemplo:

```
browser.set_viewport_size(1920, 1080)
```

save_screenshot(filename, suffix)

Salva capturas de tela no diretório `/tmp`. As capturas de tela são carregadas de lá para a pasta de artefatos do canário no bucket do S3.

filename é o nome do arquivo para a captura de tela, e suffix é uma string opcional a ser usada para nomear a captura de tela.

Exemplo:

```
browser.save_screenshot('loaded.png', 'page1')
```

Classe SyntheticsWebDriver

Para usar essa classe, use o seguinte no script:

```
from aws_synthetics.selenium import synthetics_webdriver
```

add_execution_error(errorMessage, ex);

`errorMessage` descreve o erro, e `ex` é a exceção encontrada

Você pode usar `add_execution_error` para definir erros de execução eu seu canário. Ele faz o canário falhar sem interromper a execução do script. Também não afeta suas métricas `successPercent`.

Convém rastrear erros como erros de execução somente se eles não forem importantes para indicar o sucesso ou falha do script do canário.

A seguir há um exemplo de uso do elemento `add_execution_error`. Você está monitorando a disponibilidade de seu endpoint e fazendo capturas de tela depois que a página foi carregada. Como a falha de obter uma captura de tela não determina a disponibilidade do endpoint, é possível detectar quaisquer erros encontrados durante a captura de tela e adicioná-los como erros de execução. Suas métricas de disponibilidade ainda indicarão que o endpoint está ativo e em execução, mas o status do canário será marcado como falha. O bloco de código de exemplo a seguir captura esse erro e adiciona-o como erro de execução.

```
try:
    browser.save_screenshot("loaded.png")
except Exception as ex:
   self.add_execution_error("Unable to take screenshot", ex)
```

add_user_agent(user_agent_str)

Anexa o valor de `user_agent_str` ao cabeçalho do atendente do usuário do navegador. É necessário atribuir `user_agent_str` antes de criar a instância do navegador.

Exemplo:

```
synthetics_webdriver.add_user_agent('MyApp-1.0')
```

execute_step(step_name, function_to_execute)

Processa uma função. Ela também faz o seguinte:

- Registra que a etapa começou.
- Faz uma captura de tela chamada `<stepName>-starting`.

- Inicia um temporizador.
- Executa a função fornecida.
- Se a função retornar normalmente, ela contará como aprovada. Se a função apresentar erro, ela contará como falha.
- Encerra o temporizador.
- Registra se a etapa foi aprovada ou falhou
- Faz uma captura de tela chamada `<stepName>-succeeded` ou `<stepName>-failed`.
- Emite a métrica `stepName` `SuccessPercent`, 100 para aprovação ou 0 para falha.
- Emite a métrica `stepName` `Duration` com um valor de acordo com os horários inicial e final da etapa.
- Finalmente, retorna o que `functionToExecute` retornou ou lança novamente o que `functionToExecute` lançou.

Exemplo:

```
from selenium.webdriver.common.by import By

def custom_actions():
        #verify contains
        browser.find_element(By.XPATH, "//*[@id=\"id_1\"][contains(text(),'login')]")
        #click a button
        browser.find_element(By.XPATH, '//*[@id="submit"]/a').click()

    await synthetics_webdriver.execute_step("verify_click", custom_actions)
```

Chrome()

Inicia uma instância do navegador Chromium e retorna a instância criada do navegador.

Exemplo:

```
browser = synthetics_webdriver.Chrome()
browser.get("https://example.com/)
```

Para iniciar um navegador no modo anônimo, use o seguinte:

```
add_argument('—-incognito')
```

Para adicionar configurações de proxy, use o seguinte:

```
add_argument('--proxy-server=%s' % PROXY)
```

Exemplo:

```
from selenium.webdriver.chrome.options import Options
chrome_options = Options()
chrome_options.add_argument("—-incognito")
browser = syn_webdriver.Chrome(chrome_options=chrome_options)
```

Agendamento de execuções do canário usando cron

Usar uma expressão cron lhe dá flexibilidade quando você programa um canário. As expressões cron contêm cinco ou seis campos na ordem listada na tabela a seguir. Os campos são separados por espaços. A sintaxe será diferente se você usar o console do CloudWatch para criar o canário, a AWS CLI ou SDKs da AWS. Ao usar o console, especifique apenas os cinco primeiros campos. Usando a AWS CLI ou os SDKs da AWS, especifique todos os seis campos, e você deverá especificar * para o campo `Year`.

Campo	Valores permitidos	Caracteres especiais permitidos
minutos	0-59	, - * /
Horas	0-23	, - * /
Dia do mês	1-31	, - * ? / L W
Mês	1-12 ou JAN-DEZ	, - * /
Dia da semana	1-7 ou DOM-SÁB	, - * ? L #
Ano	*	

Caracteres especiais

- A ,(vírgula) inclui vários valores na expressão de um campo. Por exemplo, no campo Month (Mês), JAN,FEB,MAR incluiria janeiro, fevereiro e março.

- O caractere especial -(traço) especifica faixas. No campo Dia, 1-15 incluiria dias 1 a 15 do mês especificado.
- O caractere especial * (asterisco) inclui todos os valores no campo. No campo Hours (Horas), * inclui todas as horas. Não é possível usar * nos campos Day-of-month (Dia do mês) e Day-of-week (Dia da semana) na mesma expressão. Se você usá-lo em um deles, utilize ? no outro.
- A / (barra) especifica incrementos. No campo Minutes (Minutos), é possível inserir 1/10 para especificar cada décimo minuto a partir do primeiro minuto da hora (por exemplo, o 11.º, 21.º e 31.º minuto etc.).
- O ? (ponto de interrogação) especifica um ou outro. No campo Day-of-month (Dia do mês), se você inserir 7 e for indiferente a qual dia da semana é o 7º, poderá inserir ? no campo Day-of-week (Dia da semana).
- O curinga L nos campos Dia do mês ou Dia da semana especifica o último dia do mês ou da semana.
- O curinga **W** no campo Dia do mês especifica um dia da semana. No campo Dia do mês, **3W** especifica o dia mais próximo do terceiro dia da semana do mês.
- O curinga # no campo Dia da semana especifica uma determinada instância do dia da semana definido dentro de um mês. Por exemplo, 3#2 é a segunda terça-feira do mês. O 3 refere-se a terça-feira, porque é o terceiro dia de cada semana, e o 2 refere-se ao segundo dia desse tipo dentro do mês.

Limitações

- Você não pode especificar os campos Dia do mês e Dia da semana na mesma expressão cron. Se você especificar um valor ou * (asterisco) em um dos campos, deverá usar ? (ponto de interrogação) no outro.
- Não há suporte para expressões Cron que causam taxas mais rápidas que um minuto.
- Não é possível configurar um canário para esperar mais de um ano para ser executado. Portanto, você pode especificar apenas * no campo `Year`.

Exemplos

Você pode consultar as seguintes sequências de caracteres cron de exemplo ao criar um canário. Os exemplos a seguir são a sintaxe correta para usar a AWS CLI ou os SDKs da AWS para criar ou atualizar um canário. Caso esteja usando o console do CloudWatch, omita a * final em cada exemplo.

Expressão	Significado
`0 10 * * ? *`	Executada às 10h (UTC) todos os dias
`15 12 * * ? *`	Executada às 12h15 (UTC) todos os dias
`0 18 ? * MON-FRI *`	Executada às 18h (UTC) de segunda a sexta
`0 8 1 * ? *`	Executar às 8h (UTC) no primeiro dia de cada mês
`0/10 * ? * MON-SAT *`	Executar a cada 10 minutos de segunda a sábado de cada semana
`0/5 8-17 ? * MON-FRI *`	Executada a cada cinco minutos, de segunda-feira a sexta-feira, entre 8h e 17h55 (UTC)

Grupos

Você pode criar grupos para associar canários entre si, incluindo canários entre regiões. O uso de grupos pode ajudar você a gerenciar e automatizar os canários, e você também pode visualizar resultados de execução agregados e estatísticas de todos os canários em um grupo.

Grupos são recursos globais. Quando você cria um grupo, ele é replicado em todas as regiões da AWS que oferecem suporte a grupos, e você pode adicionar canários de qualquer uma dessas regiões a ele e visualizá-lo em qualquer uma dessas regiões. Embora o formato do ARN do grupo reflita o nome da região em que foi criado, um grupo não está restrito a qualquer região. Isso significa que você pode colocar canários de diversas regiões no mesmo grupo e, em seguida, usar esse grupo para visualizar e gerenciar todos esses canários em um modo de exibição único.

Os grupos são compatíveis com todas as regiões, exceto com as regiões desabilitadas por padrão. Para obter mais informações sobre essas regiões, consulte Enabling a Region (Como habilitar uma região).

Cada grupo pode conter até dez canários. Você pode ter até 20 grupos em sua conta. Qualquer canário pode ser membro de até dez grupos.

Para criar um grupo

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

2. No painel de navegação, escolha Application Signals, Canários do Synthetics.

3. Selecione Create Group.
4. Em Group Name (Nome do grupo), insira um nome para o grupo.
5. Selecione canários para serem associados a este grupo. Para selecionar um canário, digite seu nome completo em Exact canary name (Nome exato do canário) e clique em Search (Pesquisar). Em seguida, marque a caixa de seleção ao lado do nome do canário. Se houver diversos canários com o mesmo nome em diferentes regiões, certifique-se de selecionar os canários desejados.

   Você pode repetir esta etapa para associar até dez canários ao grupo.
6. (Opcional) Em Tags (Etiquetas), adicione um ou mais pares chave-valor como etiquetas para esse grupo. As tags podem ajudar a identificar e organizar seus recursos da AWS e acompanhar seus custos da AWS. Para ter mais informações, consulte Etiquetar recursos do Amazon CloudWatch.
7. Selecione Create Group.

Como testar um canário localmente

Esta seção explica como modificar, testar e depurar os canários do CloudWatch Synthetics diretamente no editor de código do Microsoft Visual Studio ou no editor de código JetBrains IDE. O ambiente de depuração local usa um contêiner do Serverless Application Model (SAM) para simular uma função do Lambda com a finalidade de emular o comportamento de um canário do Synthetics.

> **Note**
>
> É impraticável executar canários de depuração local que dependem de monitoramento visual. O monitoramento visual depende da obtenção de capturas de tela básicas durante uma execução inicial e, em seguida, da comparação dessas capturas de tela com as capturas de tela de execuções subsequentes. Em um ambiente de desenvolvimento local, as execuções não são armazenadas ou rastreadas, e cada iteração corresponde a uma execução independente e autônoma. A ausência de um histórico de execução para um canário torna impraticável a depuração de canários que dependem de monitoramento visual.

Pré-requisitos

1. Escolha ou crie um bucket do Amazon S3 que deseja usar para armazenar artefatos de execuções de testes de canários locais, como arquivos em HAR e capturas de tela. Isso requer que você seja provisionado com o IAM. Se você ignorar a configuração de buckets do Amazon S3, ainda poderá testar o canário localmente, mas receberá uma mensagem de erro relacionada ao bucket ausente e não obterá acesso aos artefatos do canário.

   Se você usar um bucket do Amazon S3, recomendamos definir o ciclo de vida do bucket para excluir os objetos após alguns dias com a finalidade de economizar custos. Para obter mais informações, consulte Gerenciar seu ciclo de vida de armazenamento.
2. Configure um perfil padrão da AWS para a conta da AWS. Para obter mais informações, consulte Configuration and credential file settings.
3. Defina a região padrão da AWS do ambiente de depuração como sua região preferencial, por exemplo, `us-west-2`.
4. Instale a CLI do AWS SAM. Para obter mais informações, consulte Instalar a AWS SAM CLI.
5. Instale o Visual Studio Code Editor ou o JetBrains IDE. Para obter mais informações, consulte Visual Studio Code ou JetBrains IDE.
6. Instale o Docker para trabalhar com a CLI do AWS SAM. Certifique-se de iniciar o daemon do Docker. Para obter mais informações, consulte Installing Docker to use with the AWS SAM CLI.

   Como alternativa, é possível instalar outro software de gerenciamento de contêiner, como o Rancher, desde que essa ferramenta use o runtime do Docker.
7. Instale uma extensão do kit de ferramentas da AWS para seu editor preferencial. Para obter mais informações, consulte Installing the AWS Toolkit for Visual Studio Code ou Instalar o AWS Toolkit for JetBrains.

Tópicos

Configuração do ambiente de teste e de depuração

Primeiro, clone o repositório do Github fornecido pela AWS ao digitar o comando apresentado a seguir. O repositório contém amostras de código para canários em Node.js e canários em Python.

```
git clone https://github.com/aws-samples/synthetics-canary-local-debugging-sample.git
```

Em seguida, siga um dos procedimentos apresentados a seguir de acordo com a linguagem de programação dos seus canários.

Para canários em Node.js

1. Acesse o diretório de origem do canário em Node.js ao inserir o comando apresentado a seguir.

   ```
   cd synthetics-canary-local-debugging-sample/nodejs-canary/src
   ```

2. Insira o comando apresentado a seguir para instalar as dependências do canário.

   ```
   npm install
   ```

Para canários em Python

1. Acesse o diretório de origem do canário em Python ao inserir o comando apresentado a seguir.

   ```
   cd synthetics-canary-local-debugging-sample/python-canary/src
   ```

2. Insira o comando apresentado a seguir para instalar as dependências do canário.

   ```
   pip3 install -r requirements.txt -t .
   ```

Usar o Visual Studio Code IDE

O arquivo de configuração de inicialização do Visual Studio está localizado em `.vscode/launch.json`. O arquivo contém configurações para permitir que o arquivo de modelo seja descoberto pelo código do Visual Studio. Ele define uma carga útil do Lambda com os parâmetros obrigatórios para invocar o canário com êxito. Esta é a configuração de inicialização para um canário em Node.js:

```
{
            ...
            ...
            "lambda": {
                "payload": {
                    "json": {
                        // Canary name. Provide any name you like.
                        "canaryName": "LocalSyntheticsCanary",
                        // Canary artifact location
                        "artifactS3Location": {
                            "s3Bucket": "cw-syn-results-123456789012-us-west-2",
                            "s3Key": "local-run-artifacts",
                        },
                        // Your canary handler name
                        "customerCanaryHandlerName": "heartbeat-canary.handler"
                    }
                },
                // Environment variables to pass to the canary code
                "environmentVariables": {}
            }
        }
    ]
}
```

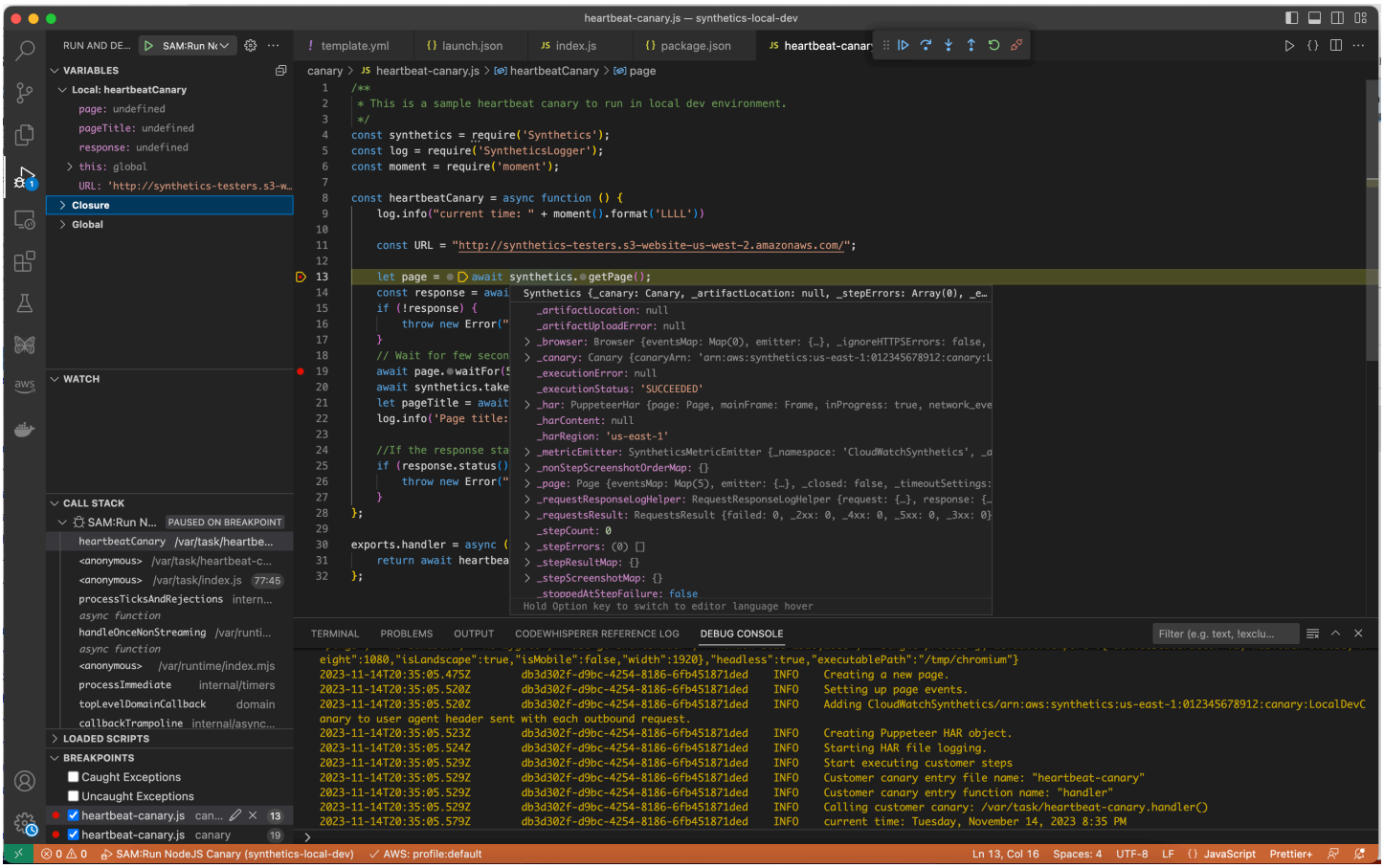
Como opção, também é possível fornecer os seguintes campos na carga útil em JSON:

- Valores válidos para `s3EncryptionMode`: `SSE_S3` | `SSE_KMS`
- Valor válido para `s3KmsKeyArn`: *`ARN da chave do KMS`*
- Valores válidos para `activeTracing`: `true` | `false`
- Valor válido para `canaryRunId`: *`UUID`* Este parâmetro é obrigatório se o rastreamento ativo estiver habilitado.

Para realizar a depuração do canário no Visual Studio, adicione pontos de interrupções no código do canário nos locais em que você deseja pausar a execução. Para adicionar um ponto de interrupção, escolha a margem do editor e acesse o modo Executar e Depurar no editor. Execute o canário ao clicar no botão de reprodução. Quando o canário for executado, os logs serão acompanhados no console de depuração, fornecendo insights em tempo real sobre o comportamento do canário. Se você adicionou pontos de interrupções, a execução do canário será pausada em cada ponto de

interrupção, permitindo a análise do código e a inspeção dos valores de variáveis, dos métodos de instância, dos atributos de objetos e da pilha de chamadas de funções.

Não há custos incorridos para executar e depurar os canários localmente, exceto para os artefatos armazenados no bucket do Amazon S3 e para as métricas do CloudWatch geradas por cada execução local.



Usar o JetBrains IDE

Após instalar a extensão AWS Toolkit for JetBrains, certifique-se de que o plug-in do Node.js e o depurador do JavaScript estejam habilitados para execução, se você estiver realizando a depuração de um canário em Node.js. Depois, siga as etapas abaixo.

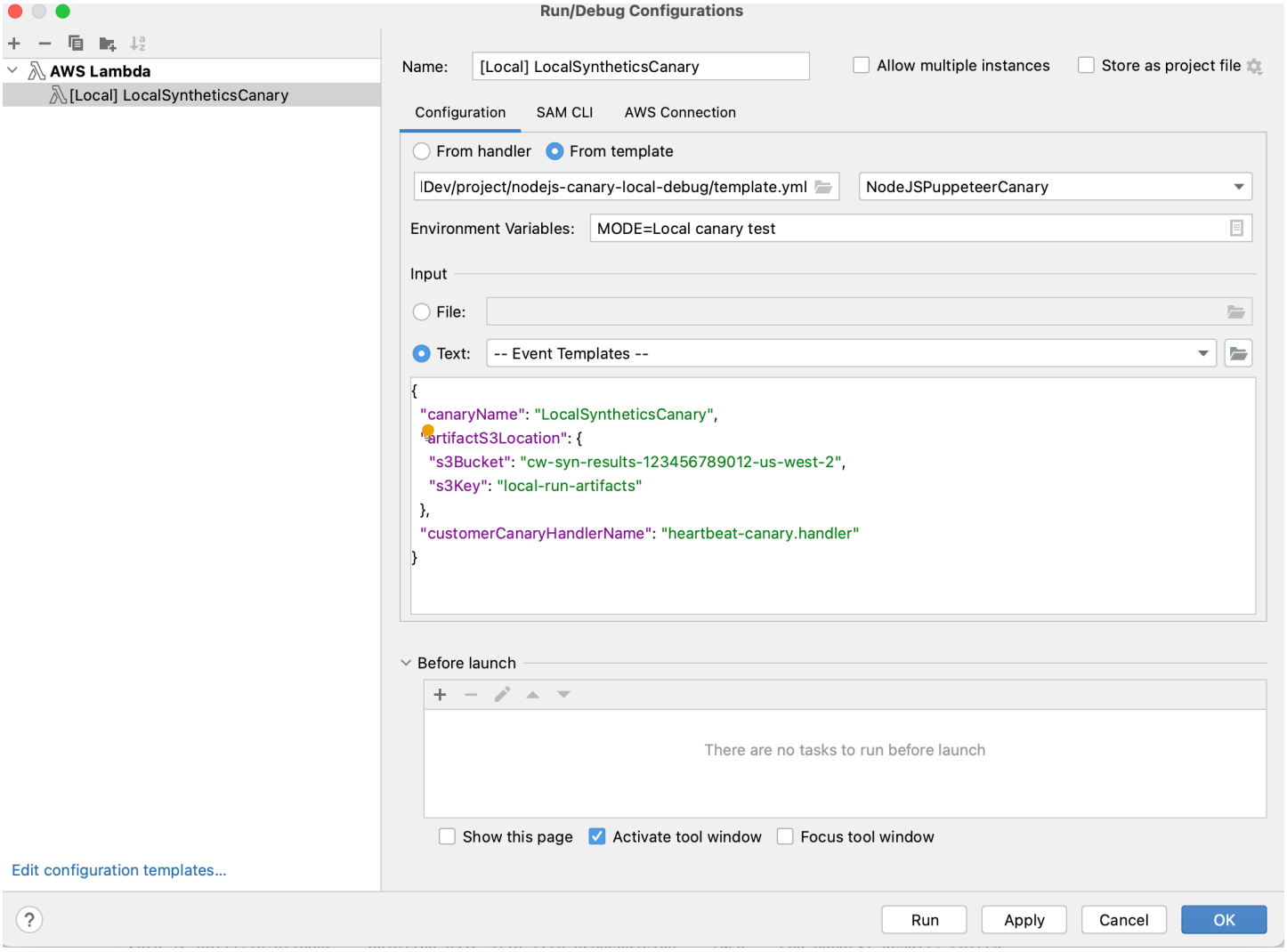
Depuração de um canário usando JetBrains IDE

1. No painel de navegação esquerdo do JetBrains IDE, escolha Lambda e, em seguida, selecione o modelo de configuração local.
2. Insira um nome para a configuração de execução, como **`LocalSyntheticsCanary`**.

3. Escolha From template, selecione o navegador de arquivos no campo de modelo e, em seguida, faça a seleção do arquivo template.yml do projeto, no diretório nodejs ou no diretório python.
4. Na seção Input, insira a carga útil do canário, conforme mostrado na tela apresentada a seguir.

```
{
 "canaryName": "LocalSyntheticsCanary",
 "artifactS3Location": {
     "s3Bucket": "cw-syn-results-123456789012-us-west-2",
     "s3Key": "local-run-artifacts"
 },
 "customerCanaryHandlerName": "heartbeat-canary.handler"
}
```

Além disso, é possível definir outras variáveis de ambiente na carga útil em JSON, conforme listado em Usar o Visual Studio Code IDE.



Execução de um canário localmente com a CLI do SAM

Use um dos procedimentos apresentados a seguir para executar o canário localmente usando a CLI do Serverless Application Model (SAM). Certifique-se de especificar seu próprio nome de bucket do Amazon S3 para `s3Bucket` em `event.json`.

Para usar a CLI do SAM para executar um canário em Node.js

1. Acesse o diretório de origem ao inserir o comando apresentado a seguir.

```
cd synthetics-canary-local-debugging-sample/nodejs-canary
```

2. Insira os comandos a seguir:

```
sam build
```

```
sam local invoke -e ../event.json
```

Para usar a CLI do SAM para executar um canário em Python

1. Acesse o diretório de origem ao inserir o comando apresentado a seguir.

```
cd synthetics-canary-local-debugging-sample/python-canary
```

2. Insira os comandos a seguir:

```
sam build
sam local invoke -e ../event.json
```

Integração do ambiente de teste local a um pacote de canário existente

É possível integrar a depuração do canário local ao seu pacote do canário existente ao copiar três arquivos:

- Copie o arquivo `template.yml` na raiz do pacote do canário. Certifique-se de modificar o caminho para `CodeUri`, de modo que o direcionamento seja para o diretório em que o código do canário está localizado.
- Se você estiver trabalhando com um canário em Node.js, copie o arquivo `cw-synthetics.js` para o diretório de origem do canário. Se você estiver trabalhando com um canário em Python, copie `cw-synthetics.py` para o diretório de origem do canário.
- Copie o arquivo de configuração de inicialização `.vscode/launch.json` para a raiz do pacote. Certifique-se de colocá-lo dentro do diretório `.vscode`. Caso o arquivo ainda não exista, crie-o.

Alteração do runtime do CloudWatch Synthetics

Como parte da depuração, é possível tentar executar um canário com um runtime diferente do CloudWatch Synthetics, em vez de usar o runtime mais recente. Para fazer isso, localize o runtime que deseja usar em uma das tabelas apresentadas a seguir. Certifique-se de selecionar o runtime para a região correta. Em seguida, cole o ARN desse runtime no local apropriado do arquivo `template.yml` e execute o canário.

Tempos de execução Node.js

ARNs para syn-nodejs-puppeteer-7.0

A tabela apresentada a seguir lista os ARNs a serem usados para a versão `syn-nodejs-puppeteer-7.0` do runtime do CloudWatch Synthetics em cada região da AWS na qual ele está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:44`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:46`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:44`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:47`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:44`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:45`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:20`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:26`
Ásia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:18`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:44`

Região	ARN
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:30`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:46`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:49`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:44`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:44`
Canadá (Central)	`arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:44`
Oeste do Canadá (Calgary)	`arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:76`
China (Pequim)	`arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:45`
China (Ningxia);	`arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:46`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:44`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:46`
Europa (Londres)	`arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:44`
Europa (Milão)	`arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:45`

Região	ARN
Europe (Paris)	`arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:44`
Europa (Espanha)	`arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:20`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:44`
Europa (Zurique)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:19`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:17`
Oriente Médio (Barém)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:44`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:19`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:45`
AWS GovCloud (Leste dos EUA)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:41`
AWS GovCloud (Oeste dos EUA)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:42`

ARNs para syn-nodejs-puppeteer-6.2

A tabela apresentada a seguir lista os ARNs a serem usados para a versão `syn-nodejs-puppeteer-6.2` do runtime do CloudWatch Synthetics em cada região da AWS na qual ele está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:41`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:43`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:41`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:44`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:41`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:42`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:17`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:23`
Ásia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:15`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:41`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:27`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:42`

Região	ARN
Ásia-Pacífico (Singapura)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:46
Ásia-Pacífico (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:41
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:41
Canadá (Central)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:41
Oeste do Canadá (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:73
China (Pequim)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:42
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:43
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:41
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:43
Europa (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:41
Europa (Milão)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:42
Europe (Paris)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:41
Europa (Espanha)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:17

Região	ARN
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:41`
Europa (Zurique)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:16`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:14`
Oriente Médio (Barém)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:41`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:16`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:42`
AWS GovCloud (Leste dos EUA)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:39`
AWS GovCloud (Oeste dos EUA)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:39`

ARNs para syn-nodejs-puppeteer-5.2

A tabela apresentada a seguir lista os ARNs a serem usados para a versão `syn-nodejs-puppeteer-5.2` do runtime do CloudWatch Synthetics em cada região da AWS na qual ele está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:42`

Região	ARN
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:44`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:42`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:45`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:42`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:43`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:18`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:24`
Ásia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:16`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:42`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:28`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:44`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:47`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:42`

Região	ARN
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:42
Canadá (Central)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:42
Oeste do Canadá (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:74
China (Pequim)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:43
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:44
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:42
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:44
Europa (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:42
Europa (Milão)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:43
Europe (Paris)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:42
Europa (Espanha)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:18
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:42
Europa (Zurique)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:17

Região	ARN
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:15`
Oriente Médio (Barém)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:42`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:17`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:43`
AWS GovCloud (Leste dos EUA)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:40`
AWS GovCloud (Oeste dos EUA)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:40`

Tempos de execução do Python

ARNs para syn-python-selenium-3.0

A tabela apresentada a seguir lista os ARNs a serem usados para a versão `syn-python-selenium-3.0` do runtime do CloudWatch Synthetics em cada região da AWS na qual ele está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`aarn:aws:lambda:us-east-1:378653112637:layer:Synthetics_Selenium:32`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics_Selenium:34`

Região	ARN
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics_Selenium:32`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics_Selenium:34`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics_Selenium:32`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics_Selenium:32`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics_Selenium:20`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics_Selenium:26`
Ásia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics_Selenium:18`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics_Selenium:32`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics_Selenium:30`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics_Selenium:34`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics_Selenium:37`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics_Selenium:32`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics_Selenium:32`

Região	ARN
Canadá (Central)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics_Selenium:32
Oeste do Canadá (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics_Selenium:76
China (Pequim)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics_Selenium:32
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics_Selenium:32
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics_Selenium:32
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics_Selenium:34
Europa (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics_Selenium:32
Europa (Milão)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics_Selenium:33
Europe (Paris)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics_Selenium:32
Europa (Espanha)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics_Selenium:20
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics_Selenium:32
Europa (Zurique)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics_Selenium:19
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:313249807427:layer:Synthetics_Selenium:17

Região	ARN
Oriente Médio (Barém)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics_Selenium:32`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics_Selenium:19`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics_Selenium:33`
AWS GovCloud (Leste dos EUA)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics_Selenium:30`
AWS GovCloud (Oeste dos EUA)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics_Selenium:31`

ARNs para syn-python-selenium-2.1

A tabela apresentada a seguir lista os ARNs a serem usados para a versão `syn-python-selenium-2.1` do runtime do CloudWatch Synthetics em cada região da AWS na qual ele está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:29`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:31`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:29`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:31`

Região	ARN
África (Cidade do Cabo)	arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:29
Ásia-Pacífico (Hong Kong)	arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:29
Ásia-Pacífico (Hyderabad)	arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:17
Ásia-Pacífico (Jacarta)	arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:23
Ásia-Pacífico (Melbourne)	arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:15
Ásia-Pacífico (Mumbai)	arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:29
Ásia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:27
Ásia-Pacífico (Seul)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:30
Ásia-Pacífico (Singapura)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:34
Ásia-Pacífico (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:29
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:29
Canadá (Central)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:29
Oeste do Canadá (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:73

Região	ARN
China (Pequim)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:29
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:29
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:29
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:31
Europa (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:29
Europa (Milão)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:30
Europe (Paris)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:29
Europa (Espanha)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:17
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:29
Europa (Zurique)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:16
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:14
Oriente Médio (Barém)	arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:29

Região	ARN
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:16`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:30`
AWS GovCloud (Leste dos EUA)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:29`
AWS GovCloud (Oeste dos EUA)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:29`

Erros comuns

Erro: a execução local de projetos do AWS SAM requer o Docker. Você o instalou e conferiu que ele está funcionando?

Certifique-se de iniciar o Docker em seu computador.

Falha ao invocar o SAM localmente: ocorreu um erro (ExpiredTokenException) ao chamar a operação GetLayerVersion: o token de segurança incluso na solicitação expirou

Certifique-se de que o perfil padrão da AWS esteja configurado.

Erros mais comuns

Para obter mais informações sobre erros comuns relacionados ao SAM, consulte AWS SAM CLI troubleshooting.

Solucionar problemas de um canário

Se o canário falhar, verifique o seguinte para solucionar problemas.

Solução de problemas gerais

- Use a página de detalhes do canário para encontrar mais informações. No console do CloudWatch, escolha Canaries (Canários) no painel de navegação e escolha o nome do canário

para abrir a página de detalhes do canário. Na guia Availability (Disponibilidade), marque a métrica SuccessPercent para ver se o problema é constante ou intermitente.

Ainda na guia Availability (Disponibilidade), escolha um ponto de dados com falha para ver capturas de tela, logs e relatórios de etapas (se disponível) para essa execução com falha.

Se um relatório de etapas estiver disponível porque as etapas fazem parte do script, verifique qual etapa falhou e veja as capturas de tela associadas para ver o problema que seus clientes estão vendo.

Também é possível verificar os arquivos HAR para ver se uma ou mais solicitações estão com falha. Você pode aprofundar usando logs para detalhar solicitações e erros com falha. Por fim, você pode comparar esses artefatos com os artefatos de uma execução bem-sucedida do canário para identificar o problema.

Por padrão, o CloudWatch Synthetics obtém capturas de tela para cada etapa em um canário de interface do usuário. Porém, seu script pode estar configurado para desabilitar capturas de tela. Durante a depuração, convém habilitar as capturas de tela novamente. Da mesma forma, para canaries de API, convém ver os cabeçalhos e o corpo da solicitação e da resposta HTTP durante a depuração. Para obter informações sobre como incluir dados no relatório de sessões, consulte executeHttpStep(stepName, requestOptions, [callback], [stepConfig]).

- Se você teve uma implantação recente em sua aplicação, reverta-a e a depure mais tarde.
- Conecte-se ao endpoint manualmente para ver se você consegue reproduzir o mesmo problema.

Tópicos

- Há falha no canário após a atualização do ambiente Lambda
- Meu canário está bloqueado pelo AWS WAF
- Aguardar que um elemento seja exibido
- O nó não é visível ou não é um HTMLElement para page.click()
- Unable to upload artifacts to S3, Exception: Unable to fetch S3 bucket location: Access Denied (Não é possível carregar artefatos para o S3, Exceção: Não é possível buscar a localização do bucket do S3: Acesso negado)
- Erro: erro de protocolo (Runtime.callFunctionOn): destino fechado.
- O canário falhou. Error: No datapoint (Erro: sem ponto de dados). O canário exibe erro de tempo limite

- Tentar acessar um endpoint interno
- Problemas de atualização e downgrade da versão do runtime do canário
- Problema de compartilhamento de solicitações de origem cruzada (CORS)
- Problemas de condições de interferências para o canário
- Solução de problemas de um canário em uma VPC

Há falha no canário após a atualização do ambiente Lambda

Os canários do CloudWatch Synthetics são implementados como funções do Lambda na sua conta. Essas funções do Lambda estão sujeitas a atualizações periódicas de runtime do Lambda que contêm atualizações de segurança, correções de bugs e outras melhorias. O Lambda se empenha em fornecer atualizações de runtime compatíveis com as funções existentes. No entanto, como acontece com a aplicação de patches de software, há casos raros em que uma atualização de runtime pode afetar negativamente uma função existente. Se você acredita que seu canário foi afetado por uma atualização de runtime do Lambda, use o modo manual de gerenciamento de runtime do Lambda (em regiões com suporte) para reverter temporariamente a versão de runtime do Lambda. Isso mantém seu canário funcionando e minimiza as interrupções, fornecendo tempo para corrigir a incompatibilidade antes de retornar à versão de runtime mais recente.

Se houver falhas no canário após uma atualização de runtime do Lambda, a melhor solução é fazer o upgrade para um dos runtimes mais recentes do Synthetics. Para obter mais informações sobre os runtimes mais recentes, consulte Versões do runtime do Synthetics.

Como solução alternativa, nas regiões em que os controles de gerenciamento de runtime do Lambda estão disponíveis, você pode reverter um canário para um runtime gerenciado do Lambda mais antigo, usando o modo manual para controles de gerenciamento de runtime. Você pode definir o modo manual usando a AWS CLI ou o console do Lambda, seguindo as etapas abaixo nas seções a seguir.

> **Warning**
>
> Quando você altera as configurações de runtime para o modo manual, a função do Lambda não recebe atualizações automáticas de segurança até que seja revertida para o modo automático. Durante esse período, a função do Lambda pode ficar suscetível a vulnerabilidades de segurança.

Pré-requisitos

- Instalar o jq
- Instalar a versão mais recente do AWS CLI. Para obter mais informações, consulte as instruções de instalação e atualização da AWS CLI.

Etapa 1: obter o ARN da função do Lambda

Execute o comando a seguir para recuperar o campo `EngineArn` da resposta. Esse `EngineArn` é o ARN da função do Lambda associada ao canário. Você usará esse ARN nas etapas a seguir.

```
aws synthetics get-canary --name my-canary | jq '.Canary.EngineArn'
```

Exemplo de saída para `EngingArn`:

```
"arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-dc5015c2-db17-4cb5-
afb1-EXAMPLE991:8"
```

Etapa 2: obter o ARN da última versão de runtime válida do Lambda

Para ajudar a entender se o canário foi afetado por uma atualização de runtime do Lambda, verifique se a data e a hora em que o ARN da versão de runtime do Lambda foi alterado nos logs apareceram na data e hora em que você viu o impacto no canário. Se não houver uma correspondência, provavelmente não é uma atualização de runtime do Lambda que está causando os problemas.

Se o canário for afetado por uma atualização de runtime do Lambda, é necessário identificar o ARN da versão de runtime do Lambda em funcionamento que estava em uso anteriormente. Siga as instruções em Identifying runtime version changes para encontrar o ARN do runtime anterior. Registre o ARN da versão de runtime e prossiga para a Etapa 3, a fim de definir a configuração de gerenciamento do runtime.

Se o canário ainda não foi afetado por uma atualização do ambiente Lambda, você pode encontrar o ARN da versão de runtime do Lambda que está usando atualmente. Execute o comando a seguir para recuperar o `RuntimeVersionArn` da função do Lambda da resposta.

```
aws lambda get-function-configuration \
--function-name "arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-
dc5015c2-db17-4cb5-afb1-EXAMPLE991:8" | jq '.RuntimeVersionConfig.RuntimeVersionArn'
```

Exemplo de saída para `RuntimeVersionArn`:

```
"arn:aws:lambda:us-
west-2::runtime:EXAMPLE647b82f490a45d7ddd96b557b916a30128d9dcab5f4972911ec0f"
```

Etapa 3: atualizar a configuração de gerenciamento de runtime do Lambda

Você pode usar a AWS CLI ou o console do Lambda para atualizar a configuração de gerenciamento de runtime.

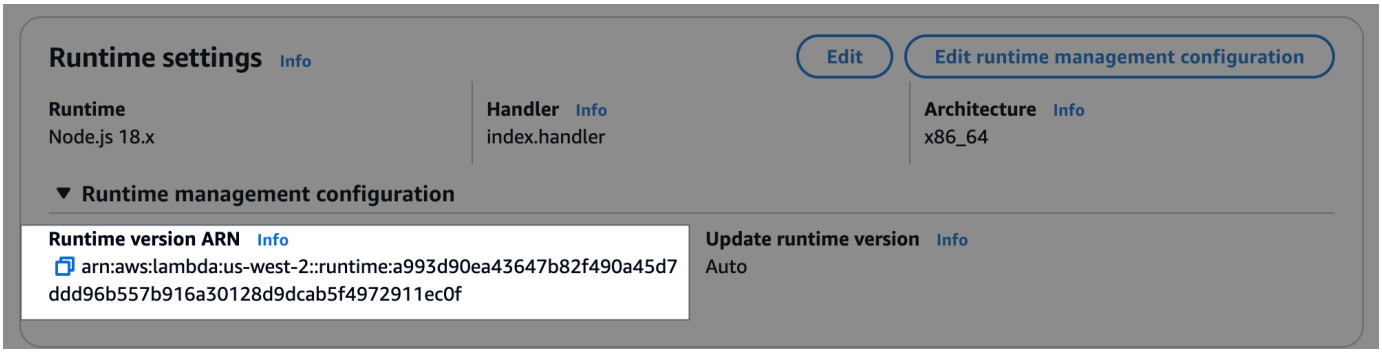
Definir o modo manual de configuração do gerenciamento de runtime do Lambda usando a AWS CLI

Insira o comando a seguir para alterar o gerenciamento de runtime da função do Lambda para o modo manual. Certifique-se de substituir *`function-name`* e *`qualificador`* pelo ARN da função do Lambda e pelo número da versão da função do Lambda, respectivamente, usando os valores encontrados na Etapa 1. Também substitua *`runtime-version-arn`* pelo ARN da versão encontrado na Etapa 2.

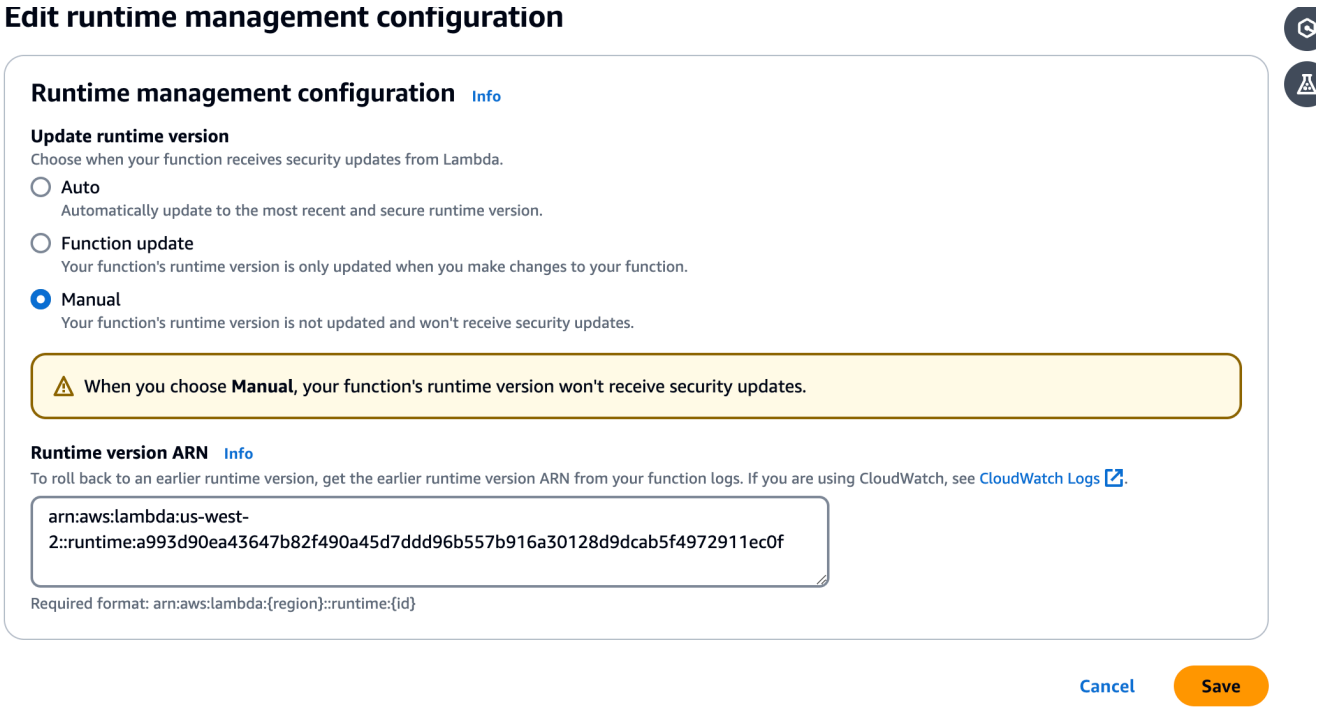
```
aws lambda put-runtime-management-config \
    --function-name "arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-
dc5015c2-db17-4cb5-afb1-EXAMPLE991" \
    --qualifier 8 \
    --update-runtime-on "Manual" \
    --runtime-version-arn "arn:aws:lambda:us-
west-2::runtime:a993d90ea43647b82f490a45d7ddd96b557b916a30128d9dcab5f4972911ec0f"
```

Mudar um canário para o modo manual usando o console do Lambda

1. Abra o console AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Escolha a guia Versões, selecione o link do número da versão correspondente ao seu ARN e a guia Código.
3. Role para baixo até Configurações de runtime, expanda Configuração de gerenciamento de runtime e copie o ARN da versão de runtime.



4. Escolha Editar configuração de gerenciamento de runtime, selecione Manual, cole o ARN da versão de runtime copiada anteriormente no campo ARN da versão de runtime. Em seguida, escolha Salvar.



Meu canário está bloqueado pelo AWS WAF

Para evitar que o AWS WAF bloqueie o canário, configure uma condição de correspondência de string do AWS WAF que permita a string `CloudWatchSynthetics`. Para obter mais informações, consulte Working with string match conditions na documentação do AWS WAF.

Aguardar que um elemento seja exibido

Depois de analisar seus logs e suas capturas de tela, se você perceber que seu script está aguardando que um elemento seja exibido na tela e ultrapassar o tempo limite, verifique a captura

de tela relevante para ver se o elemento é exibido na página. Verifique o `xpath` para se certificar de que está correto.

Para problemas relacionados ao Puppeteer, consulte a página do GitHub do Puppeteer ou fóruns da Internet.

O nó não é visível ou não é um HTMLElement para page.click()

Se um nó não estiver visível ou não for um `HTMLElement` para `page.click()`, verifique primeiro o `xpath` que você está usando para clicar no elemento. Além disso, se o elemento estiver na parte inferior da tela, ajuste o visor. Por padrão, o CloudWatch Synthetics usa um visor de 1920 * 1080. Você pode definir um visor diferente ao iniciar o navegador ou usando a função `page.setViewport` do Puppeteer.

Unable to upload artifacts to S3, Exception: Unable to fetch S3 bucket location: Access Denied (Não é possível carregar artefatos para o S3, Exceção: Não é possível buscar a localização do bucket do S3: Acesso negado)

Se o canário falha em decorrência de um erro do Amazon S3, significa que CloudWatch Synthetics não conseguiu carregar capturas de tela, logs ou relatórios criados para o canário devido a problemas de permissão. Verifique o seguinte:

- Verifique se a função do IAM do canário tem a permissão `s3:ListAllMyBuckets`, a permissão `s3:GetBucketLocation` para o bucket correto do Amazon S3 e a permissão `s3:PutObject` para o bucket no qual o canário armazena seus artefatos. Se o canário realizar monitoramento visual, a função também precisa da permissão `s3:GetObject` para o bucket. Essas mesmas permissões também são exigidas na política de endpoints de gateway da VPC para o Amazon S3, caso o canário seja implantado em uma VPC com um endpoint da VPC.
- Se o canário usar uma chave do AWS KMS gerenciada pelo cliente para criptografia em vez da chave gerenciada pela AWS (padrão), a função do IAM do canário poderá não ter permissão para criptografar ou descriptografar usando essa chave. Para ter mais informações, consulte Criptografar artefatos do canário.
- Sua política de bucket pode não permitir o mecanismo de criptografia usada pelo canário. Por exemplo, se a política de bucket exigir usar um mecanismo de criptografia específico ou uma chave do KMS, você deverá selecionar o mesmo modo de criptografia para o canário.

Se o canário realizar monitoramento visual, consulte Atualizar a localização e a criptografia do artefato ao usar o monitoramento visual para obter mais informações.

Erro: erro de protocolo (Runtime.callFunctionOn): destino fechado.

Esse erro aparecerá se houver algumas solicitações de rede depois que a página ou o navegador for fechado. Talvez você tenha se esquecido de aguardar uma operação assíncrona. Depois de executar seu script, o CloudWatch Synthetics fechará o navegador. A execução de qualquer operação assíncrona após o fechamento do navegador poderá causar `target closed error`.

O canário falhou. Error: No datapoint (Erro: sem ponto de dados). O canário exibe erro de tempo limite

Siggnifica que a execução do canário excedeu o tempo limite. A execução do canário foi interrompida antes que o CloudWatch Synthetics pudesse publicar métricas de porcentagem de sucesso do CloudWatch ou atualizar artefatos como arquivos HAR, logs e capturas de tela. Se o tempo limite for muito baixo, é possível aumentá-lo.

Por padrão, o valor de tempo limite do canário é igual à frequência dele. É possível ajustar manualmente o valor de tempo limite para ser menor que ou igual à frequência do canário. Se sua frequência do canário for baixa, será necessário aumentar a frequência para aumentar o tempo limite. É possível ajustar a frequência e o valor de tempo limite em Schedule (Programar) ao criar ou atualizar um canário usando o console do CloudWatch Synthetics.

Certifique-se de que valor do tempo limite do canário não seja menos de 15 segundos para permitir que o Lambda seja iniciado a frio e que a instrumentação do canário seja inicializada.

Os artefatos do canário não estão disponíveis para exibição no console do CloudWatch Synthetics quando esse erro ocorre. É possível usar o CloudWatch Logs para ver os logs do canário.

Para usar o CloudWatch Logs para ver os logs de um canário

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação esquerdo, escolha Log groups (Grupos de log).
3. Localize o grupo de logs inserindo o nome do canário na caixa de filtro. Os grupos de log para canários têm o nome /aws/lambda/cwsyn-***canaryName***-randomId.

Tentar acessar um endpoint interno

Para que seu canário acesse um endpoint em sua rede interna, recomendamos configurar o CloudWatch Synthetics para usar a VPC. Para ter mais informações, consulte Execução de um canário em uma VPC.

Problemas de atualização e downgrade da versão do runtime do canário

Se você atualizou recentemente o canário da versão do runtime `syn-1.0` para uma versão posterior, pode ser um problema de compartilhamento de solicitação de origem cruzada (CORS). Para ter mais informações, consulte Problema de compartilhamento de solicitações de origem cruzada (CORS).

Se você fez o downgrade do canário recentemente para uma versão de runtime mais antiga, verifique se as funções do CloudWatch Synthetics utilizadas estão disponíveis na versão de runtime mais antiga para a qual fez o downgrade. Por exemplo, a função `executeHttpStep` está disponível para a versão de runtime `syn-nodejs-2.2` e posteriores. Para conferir a disponibilidade de funções, consulte Escrever um script do canário.

> **Note**
>
> Ao planejar o upgrade ou downgrade da versão de runtime para um canário, recomendamos primeiro clonar o canário e atualizar a versão de runtime no canário clonado. Depois de verificar que o clone com a nova versão do runtime funciona, é possível atualizar a versão do runtime do canário original e excluir o clone.

Problema de compartilhamento de solicitações de origem cruzada (CORS)

Em um canário de interface do usuário, se algumas solicitações de rede estiverem apresentando falha com `403` ou `net::ERR_FAILED`, verifique se o canário tem o rastreamento ativo habilitado e também usa a função `page.setExtraHTTPHeaders` do Puppeteer para adicionar cabeçalhos. Nesse caso, as solicitações de rede com falha podem ser causadas por restrições de compartilhamento de solicitações de origem cruzada (CORS). Confirme se esse é o caso desabilitando o rastreamento ativo ou removendo os cabeçalhos HTTP excedentes.

Por que isso acontece?

Quando o rastreamento ativo é usado, adiciona-se um cabeçalho a mais a todas as solicitações de saída para rastrear a chamada. Modificar os cabeçalhos de solicitação adicionando um cabeçalho de rastreamento ou adicionando cabeçalhos a mais usando o `page.setExtraHTTPHeaders`do Puppeteer caus uma verificação CORS para solicitações XMLHttpRequest (XHR).

Se você não quiser desabilitar o rastreamento ativo ou remover os cabeçalhos excedentes, poderá atualizar sua aplicação Web para permitir acesso entre origens ou desabilitar a segurança da Web usando o sinalizador `disable-web-security` ao iniciar o navegador Chrome em seu script.

É possível substituir os parâmetros de inicialização usados pelo CloudWatch Synthetics e passar outros parâmetros de sinalização `disable-web-security` usando a função de inicialização do CloudWatch Synthetics. Para ter mais informações, consulte Funções de biblioteca disponíveis para scripts do canário do Node.js.

> **Note**
>
> Você pode substituir os parâmetros de inicialização usados pelo CloudWatch Synthetics ao usar a versão `syn-nodejs-2.1` do runtime ou versões posteriores.

Problemas de condições de interferências para o canário

Para obter a melhor experiência ao usar o CloudWatch Synthetics, certifique-se de que o código gravado para os canários seja idempotente. Caso contrário, em casos raros, as execuções do canário poderão encontrar condições de interferências quando o canário interagir com o mesmo recurso em execuções diferentes.

Solução de problemas de um canário em uma VPC

Se tiver problemas após criar ou atualizar um canário em uma VPC, uma das seções a seguir poderá ajudar você a solucionar o problema.

Novo canário em estado de erro ou não foi possível atualizar o canário

Se você criar um canário para ser executado em uma VPC, e ele imediatamente entrar em um estado de erro, ou você não conseguir atualizar um canário para ser executado em uma VPC, a função do canário pode não ter as permissões corretas. Para ser executado em uma VPC, um canário deve ter as permissões `ec2:CreateNetworkInterface`, `ec2:DescribeNetworkInterfaces` e `ec2:DeleteNetworkInterface`. Essas permissões estão todas contidas na política gerenciada `AWSLambdaVPCAccessExecutionRole`. Para obter mais informações, consulte Função de execução e permissões de usuário.

Se esse problema aconteceu quando você criou um canário, você deverá excluir o canário e criar um novo. Se você usar o console do CloudWatch para criar o canário, selecione Access Permissions

(Permissões de acesso) e selecione Create a new role (Criar uma função). É criada uma nova função com todas as permissões necessárias para executar o canário.

Se o problema acontecer ao atualizar um canário, será possível atualizar o canário novamente e fornecer uma nova função com as permissões necessárias.

Erro "Nenhum resultado de teste retornado"

Se um canário exibir um erro "nenhum resultado de teste retornado", um dos seguintes problemas pode ser a causa:

- Se sua VPC não tiver acesso à Internet, você deverá usar endpoints da VPC para conceder ao canário acesso ao CloudWatch e o Amazon S3. Você deverá habilitar as opções DNS resolution (Resolução DNS) e DNS hostname (Nome de host DNS) na VPC para que esses endpoints ajam para resolver rapidamente. Para obter mais informações, consulte Uso do DNS com sua VPC e Uso do CloudWatch e do CloudWatch Synthetics com endpoints da VPC de interface.
- Os canaries devem ser executadas em sub-redes privadas dentro de uma VPC. Para verificar isso, abra a página Subnets (Sub-redes) no console da VPC. Verifique as sub-redes que você selecionou ao configurar o canário. Se elas tiverem um caminho para um gateway de Internet (igw-), não são sub-redes privadas.

Para ajudar a solucionar esses problemas, consulte os logs do canário.

Como ver os eventos de log de um canário

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. Escolha o nome do grupo de logs do canário. O nome do grupo de logs começa com `/aws/lambda/cwsyn-`*`canary-name`*.

Código de exemplo para scripts do canário

Esta seção contém exemplos de código que ilustram algumas funções possíveis para scripts do canário do CloudWatch Synthetics.

Exemplos para Node.js e Puppeteer

Definir cookies

Os sites dependem de cookies para fornecer funcionalidades personalizadas ou rastrear usuários. Ao configurar cookies nos scripts do CloudWatch Synthetics, é possível imitar esse comportamento personalizado e validá-lo.

Por exemplo, um site pode exibir um link de login para um usuário recorrente em vez de um link de incrição.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadBlueprint = async function () {

    let url = "http://smile.amazon.com/";

    let page = await synthetics.getPage();

    // Set cookies.  I found that name, value, and either url or domain are required
 fields.
    const cookies = [{
      'name': 'cookie1',
      'value': 'val1',
      'url': url
    },{
      'name': 'cookie2',
      'value': 'val2',
      'url': url
    },{
      'name': 'cookie3',
      'value': 'val3',
      'url': url
    }];

    await page.setCookie(...cookies);

    // Navigate to the url
    await synthetics.executeStep('pageLoaded_home', async function (timeoutInMillis =
 30000) {
```

```
        var response = await page.goto(url, {waitUntil: ['load', 'networkidle0'],
 timeout: timeoutInMillis});

        // Log cookies for this page and this url
        const cookiesSet = await page.cookies(url);
        log.info("Cookies for url: " + url + " are set to: " +
 JSON.stringify(cookiesSet));
    });

};

exports.handler = async () => {
    return await pageLoadBlueprint();
};
```

Emulação de dispositivo

É possível gravar scripts que emulam vários dispositivos, aproximando-se da aparência e do comportamento de uma página nesses dispositivos.

O exemplo a seguir emula um dispositivo iPhone 6. Para obter mais informações sobre emulação, consult e[page.emulate(options)] na documentação do Puppeteer.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');
const puppeteer = require('puppeteer-core');

const pageLoadBlueprint = async function () {

    const iPhone = puppeteer.devices['iPhone 6'];

    // INSERT URL here
    const URL = "https://amazon.com";

    let page = await synthetics.getPage();
    await page.emulate(iPhone);

    //You can customize the wait condition here. For instance,
    //using 'networkidle2' may be less restrictive.
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});
    if (!response) {
        throw "Failed to load page!";
```

```
    }

    await page.waitFor(15000);

    await synthetics.takeScreenshot('loaded', 'loaded');

    //If the response status code is not a 2xx success code
    if (response.status() < 200 || response.status() > 299) {
        throw "Failed to load page!";
    }
};

exports.handler = async () => {
    return await pageLoadBlueprint();
};
```

Canário de API de várias etapas

Este código de exemplo demonstra um canário de API com duas etapas HTTP: testar a mesma API para casos de teste positivos e negativos. A configuração da etapa é aprovada para habilitar o relatório de cabeçalhos de solicitação/resposta. Além disso, ele oculta o cabeçalho de autorização e X-Amz-Security-Token, porque contêm credenciais de usuário.

Quando esse script é usado como um canário, é possível exibir detalhes sobre cada etapa e as solicitações HTTP associadas, como aprovação/reprovação da etapa, duração e métricas de performance, como tempo de pesquisa do DNS e tempo do primeiro byte. Você pode visualizar o número de 2xx, 4xx e 5xx da execução do canário.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const apiCanaryBlueprint = async function () {

    // Handle validation for positive scenario
    const validatePositiveCase = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 200 || res.statusCode > 299) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            let responseBody = '';
            res.on('data', (d) => {
```

```
                responseBody += d;
            });

            res.on('end', () => {
                // Add validation on 'responseBody' here if required. For ex, your
 status code is 200 but data might be empty
                resolve();
            });
        });
    };

    // Handle validation for negative scenario
    const validateNegativeCase = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 400) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            resolve();
        });
    };

    let requestOptionsStep1 = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/product/validProductName',
        'port': 443,
        'protocol': 'https:'
    };

    let headers = {};
    headers['User-Agent'] = [synthetics.getCanaryUserAgentString(), headers['User-
Agent']].join(' ');

    requestOptionsStep1['headers'] = headers;

    // By default headers, post data and response body are not included in the report
 for security reasons.
    // Change the configuration at global level or add as step configuration for
 individual steps
    let stepConfig = {
        includeRequestHeaders: true,
        includeResponseHeaders: true,
```

```
        restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted
 header values do not appear in report generated.
        includeRequestBody: true,
        includeResponseBody: true
    };


    await synthetics.executeHttpStep('Verify GET products API with valid name',
 requestOptionsStep1, validatePositiveCase, stepConfig);

    let requestOptionsStep2 = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/canary/InvalidName(',
        'port': 443,
        'protocol': 'https:'
    };

    headers = {};
    headers['User-Agent'] = [synthetics.getCanaryUserAgentString(), headers['User-
Agent']].join(' ');

    requestOptionsStep2['headers'] = headers;

    // By default headers, post data and response body are not included in the report
 for security reasons.
    // Change the configuration at global level or add as step configuration for
 individual steps
    stepConfig = {
        includeRequestHeaders: true,
        includeResponseHeaders: true,
        restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted
 header values do not appear in report generated.
        includeRequestBody: true,
        includeResponseBody: true
    };

    await synthetics.executeHttpStep('Verify GET products API with invalid name',
 requestOptionsStep2, validateNegativeCase, stepConfig);

};

exports.handler = async () => {
    return await apiCanaryBlueprint();
```

```
};
```

Amostras para Python e Selenium

O código de exemplo do Selenium a seguir é um canário que falha com uma mensagem de erro personalizada quando um elemento de destino não é carregado.

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver
from aws_synthetics.common import synthetics_logger as logger
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By

def custom_selenium_script():
    # create a browser instance
    browser = webdriver.Chrome()
    browser.get('https://www.example.com/')
    logger.info('navigated to home page')
    # set cookie
    browser.add_cookie({'name': 'foo', 'value': 'bar'})
    browser.get('https://www.example.com/')
    # save screenshot
    browser.save_screenshot('signed.png')
    # expected status of an element
    button_condition = EC.element_to_be_clickable((By.CSS_SELECTOR, '.submit-button'))
    # add custom error message on failure
    WebDriverWait(browser, 5).until(button_condition, message='Submit button failed to
 load').click()
    logger.info('Submit button loaded successfully')
    # browser will be quit automatically at the end of canary run,
    # quit action is not necessary in the canary script
    browser.quit()

# entry point for the canary
def handler(event, context):
    return custom_selenium_script()
```

Canaries e rastreamento do X-Ray

Você pode habilitar o rastreamento ativo AWS X-Ray somente para canários que usam o runtime `syn-nodejs-2.0` ou posteriores. Com o rastreamento habilitado, os rastreamentos são enviados a todas as chamadas realizadas pelo canário que usam o navegador, o AWS SDK ou módulos HTTP

ou HTTPS. Os canários com o rastreamento habilitado aparecem no mapa de rastreamento do X-Ray e no Application Signals depois que você o habilita para a aplicação.

> **Note**
>
> A ativação do rastreamento do X-Ray em canaries ainda não tem suporte na região Ásia-Pacífico (Jacarta).

Quando um canário aparece no mapa de rastreamento do X-Ray, ele aparece como um novo tipo de nó de cliente. Você pode passar o mouse sobre um nó do canário para ver dados sobre latência, solicitações e falhas. Também é possível escolher o nó do canário para ver mais dados na parte inferior da página. Nesta área da página, você pode escolher View in Synthetics (Ver no Synthetics) para acessar o console do CloudWatch Synthetics para obter mais detalhes sobre o canário, ou escolha View Traces (Visualizar os rastreamentos) para ver mais detalhes sobre os rastreamentos de execuções desse canário.

Um canário com rastreamento habilitado também tem uma guia Tracing (Rastreamento) em sua página de detalhes, com detalhes sobre rastreamentos e segmentos das execuções do canário.

Habilitar o rastreamento aumenta o runtime do canário em 2,5% a 7%.

Um canário com rastreamento habilitado deve usar uma função com as permissões a seguir. Se você usar o console para criar a função ao criar o canário, ela receberá essas permissões.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Sid230934",
            "Effect": "Allow",
            "Action": [
                "xray:PutTraceSegments"
            ],
            "Resource": "*"
        }
    ]
}
```

Os rastreamentos gerados por canaries incorrem em cobranças. Para obter mais informações sobre o preço do X-Ray, consulte Preço do AWS X-Ray.

Execução de um canário em uma VPC

É possível executar canaries em endpoints em uma VPC, bem como em endpoints internos públicos. Para executar um canário em uma VPC, você deve ter as opções DNS Resolution (Resolução DNS) e DNS hostnames (Nomes de host DNS) habilitadas na VPC. Para obter mais informações, consulte Using DNS with Your VPC.

Ao executar um canário em um endpoint da VPC, você deve fornecer um modo para que ele enviar suas métricas ao CloudWatch e seus artefatos ao Amazon S3. Se a VPC já estiver habilitada para acesso à Internet, não há mais nada a fazer. O canário será executado na VPC, mas poderá acessar a Internet para carregar métricas e artefatos.

Se a VPC ainda não estiver habilitada para acesso à Internet, você terá duas opções:

- Habilite-a para acesso à Internet. Para obter mais informações, consulte a seção Dar acesso à internet para o seu canário em uma VPC a seguir.
- Se quiser manter sua VPC privada, você poderá configurar o canário para enviar seus dados ao CloudWatch e ao Amazon S3 por meio de endpoints da VPC privados. Caso ainda não tenha feito isso, você deverá criar um endpoint da VPC para o CloudWatch (com.amazonaws).*região*.monitoring) e um endpoint de gateway para o Amazon S3. Para obter mais informações, consulte Usar o CloudWatch e o CloudWatch Synthetics com endpoints da VPC de interface e Amazon VPC Endpoints para Amazon S3.

Dar acesso à internet para o seu canário em uma VPC

Siga estas etapas para conceder acesso à Internet ao canário da VPC ou para atribuir a ele um endereço IP estático

Para dar acesso à internet a um canário em uma VPC

1. Crie um gateway NAT em uma sub-rede pública no VPC. Para obter instruções, consulte Create a NAT gateway (Criar um gateway NAT)
2. Adicione uma nova rota à tabela de rotas na sub-rede privada em que o canário é iniciado. Especifique o seguinte:
   - Em Destination (Destinação), insira **`0.0.0.0/0`**
   - Em Destino, escolha Gateway NAT e, em seguida, escolha o ID do gateway NAT que você criou.

- Escolha Save routes (Salvar rotas).

Para obter mais informações sobre a adição da rota na tabela de rotas, consulte Adicionar e remover rotas de uma tabela de rotas.

> **Note**
>
> Certifique-se de que as rotas para seu gateway NAT estejam em um status active (ativo). Se o gateway NAT for excluído e você não tiver atualizado as rotas, elas estarão em um status de buraco negro. Para obter mais informações, consulte Trabalhar com gateways NAT.

Criptografar artefatos do canário

O CloudWatch Synthetics armazena artefatos do canário, como capturas de tela, arquivos HAR e relatórios no bucket do Amazon S3. Por padrão, esses artefatos são criptografados em repouso usando uma chave gerenciada pela AWS. Para obter mais informações, consulte Chaves do cliente e chaves da AWS.

Você pode escolher usar uma opção de criptografia diferente. O CloudWatch Synthetics oferece suporte ao seguinte:

- SSE-S3: criptografia do lado do servidor (SSE) com uma chave gerenciada pelo Amazon S3.
- SSE-KMS: SSE com uma chave do AWS KMS gerenciada pelo cliente.

Se você quiser usar a opção de criptografia padrão com uma chave gerenciada pela AWS, não precisa de permissões adicionais.

Para usar a criptografia SSE-S3, especifique SSE_S3 como o modo de criptografia ao criar ou atualizar seu canário. Não são necessárias permissões adicionais para usar esse modo de criptografia. Para obter mais informações, consulte Proteção de dados usando criptografia no lado do servidor com chaves de criptografia gerenciadas do Amazon S3 (SSE-S3).

Para usar uma chave do AWS KMS gerenciada pelo cliente, especifique SSE-KMS como o modo de criptografia quando você cria ou atualiza seu canário e também forneça o nome do recurso da Amazon (ARN) da sua chave. Você também pode usar uma chave do KMS entre contas.

Para usar uma chave gerenciada pelo cliente, são necessárias as seguintes configurações:

- A função do IAM para seu canário deve ter permissão para criptografar seus artefatos usando sua chave. Se você estiver usando o monitoramento visual, também deverá dar permissão para descriptografar artefatos.

```
{
    "Version": "2012-10-17",
    "Statement": {"Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "Your KMS key ARN"
    }
}
```

- Em vez de adicionar permissões à sua função do IAM, você pode adicionar sua função do IAM à política de chaves. Considere utilizar essa abordagem se você usar a mesma função para vários canaries.

```
{
    "Sid": "Enable IAM User Permissions",
    "Effect": "Allow",
    "Principal": {
        "AWS": "Your synthetics IAM role ARN"
    },
    "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
    ],
    "Resource": "*"
}
```

- Se você estiver usando uma chave do KMS entre contas, consulte Permitir que usuários de outras contas usem uma chave do KMS.

Visualização de artefatos do canário criptografados ao usar uma chave gerenciada pelo cliente

Para visualizar artefatos do canário, atualize sua chave gerenciada pelo cliente para dar ao AWS KMS a permissão de descriptografar para o usuário visualizar os artefatos. Como alternativa, adicione permissões de descriptografia ao usuário ou perfil do IAM que estiver exibindo os artefatos.

A política padrão AWS KMS habilita as políticas do IAM na conta para conceder acesso às chaves do KMS. Se você estiver usando uma chave do KMS entre contas, consulte Por que os usuários entre contas recebem erros de acesso negado quando tentam acessar objetos do Amazon S3 criptografados por uma chave personalizada do AWS KMS?.

Para obter mais informações sobre a solução de problemas de acesso negadas por causa de uma chave do KMS, consulte Solucionar problemas de acesso à chave.

Atualizar a localização e a criptografia do artefato ao usar o monitoramento visual

Para realizar o monitoramento visual, o CloudWatch Synthetics compara suas capturas de tela com capturas de tela de linha de base adquiridas na execução selecionada como linha de base. Se você atualizar a localização do artefato ou a opção de criptografia, faça o seguinte:

- Certifique-se de que sua função do IAM tenha permissão suficiente para o local anterior do Amazon S3 e para o novo local do Amazon S3 para artefatos. Certifique-se também de que ele tenha permissão para os métodos de criptografia anteriores e novos e chaves do KMS.
- Crie uma nova linha de base selecionando a próxima execução do canário como uma nova linha de base. Se você usar essa opção, você só precisa garantir que sua função do IAM tenha permissões suficientes para a nova opção de criptografia e localização do artefato.

Recomendamos a segunda opção de seleção da próxima execução como a nova linha de base. Isso evita ter uma dependência de um local de artefato ou opção de criptografia que você não está mais usando para o canário.

Por exemplo, suponha que seu canário use a localização do artefato A e a chave K do KMS para carregar artefatos. Se você atualizar seu canário para o local de artefato B e a chave L do KMS, você pode garantir que sua função do IAM tenha permissões para ambos os locais de artefatos (A e B) e ambas as chaves do KMS (K e L). Como alternativa, você pode selecionar a próxima execução como a nova linha de base e garantir que seu perfil do IAM do canário tenha permissões para o local do artefato B e a chave L do KMS.

Visualizar estatísticas e detalhes de canaries

É possível visualizar detalhes sobre os canaries e ver estatísticas sobre as execuções deles.

Para poder ver todos os detalhes sobre os resultados da execução do canário, é necessário estar conectado em uma conta com permissões suficientes. Para ter mais informações, consulte Funções e permissões obrigatórias para canaries do CloudWatch.

Como visualizar estatísticas e detalhes de canaries

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals, Canários do Synthetics.

   Nos detalhes sobre os canaries criados:

   - Status mostra visualmente quantos canaries foram aprovados nas execuções mais recentes.
   - Groups (Grupos) exibe os grupos que você criou e mostra quantos deles têm canários com falhas ou com alarmes.
   - Slowest performers (Performances mais lentas) exibe o grupo e a região com os canários com performances mais lentas. Eles são calculados somando a duração média de todos os canários (ao longo do período de tempo selecionado) dentro de um grupo ou região e dividindo-a pelo número de canários no grupo ou região. Se você escolher a métrica para o grupo mais lento, a tabela será filtrada para exibir apenas os grupos mais lentos e seus canários. A tabela é classificada com base na duração média.
   - Perto da parte inferior da página há uma tabela que exibe todos os canaries. Uma coluna exibe os alarmes criados para cada canário. Somente os alarmes que estejam em conformidade com o padrão de nomeação para alarmes do canário serão exibidos. Esse padrão é o `Synthetics-Alarm-`*`canaryName`*`-`*`index`* . Os alarmes do canário criados por você na seção Synthetics do console do CloudWatch usarão automaticamente essa convenção de nomenclatura. Se criar alarmes do canário na seção Alarms (Alarmes) do console do CloudWatch ou usando o AWS CloudFormation e não usar esta convenção de nomenclatura, os alarmes funcionarão, mas eles não serão exibidos nessa lista.
3. Para visualizar mais detalhes sobre um único canário, escolha o nome do canário na tabela Canaries (Canários).

   Nos detalhes sobre o canário em questão:

   - A guia Availability (Disponibilidade) exibe informações sobre as execuções recentes desse canário.

     Em Canary runs (Execuções do canários), você pode escolher uma das linhas para ver detalhes sobre essa execução.

     No gráfico, você pode escolher Steps (Etapas), Screenshot (Captura de tela), Logs ou HAR file (Arquivo HAR) para visualizar esses tipos de detalhes. Se o canário estiver com o

rastreamento ativo habilitado, também é possível escolher Traces (Rastreamentos) para ver informações de rastreamento das execeuções do canário.

Os logs das execuções do canário são armazenados em buckets do S3 e no CloudWatch Logs.

As capturas de tela mostram como seus clientes visualizam suas páginas da Web. É possível usar os arquivos HAR (arquivos HTTP Archive) para visualizar dados de performance detalhados sobre as páginas da Web. Você pode analisar a lista de solicitações da web e detectar problemas de performance, como tempo de carregamento de um item. Os arquivos de log mostram o registro de interações entre a execução do canário e a página da Web e podem ser usados para identificar detalhes de erros.

Se o canário usar o runtime `syn-nodejs-2.0-beta` ou posteriores, você pode classificar os arquivos HAR por código de status, tamanho da solicitação ou duração.

A guia Steps (Etapas) exibe uma lista das etapas do canário, o status de cada etapa, o motivo da falha, o URL após a execução da etapa, as capturas de tela e a duração da execução da etapa. Para canários de API com etapas HTTP, você poderá visulizar etapas e solicitações HTTP correspondentes, se estiver usando o runtime `syn-nodejs-2.2` ou posteriores.

Escolha HTTP Requests (Solicitações de HTTP) para exibir o log de cada solicitação HTTP feita pelo canário. Você pode exibir cabeçalhos de solicitação/resposta, corpo de resposta, código de status, tempos de erro e performance (duração total, tempo de conexão TCP, tempo de handshake TLS, tempo de primeiro byte e tempo de transferência de conteúdo). Todas as solicitações HTTP que usam o módulo HTTP/HTTPS nos bastidores são capturadas aqui.

Por padrão, nos canaries da API, o cabeçalho da solicitação, o cabeçalho da resposta, o corpo da solicitação e o corpo da resposta não são incluídos no relatório por motivos de segurança. Se você escolher incluí-los, os dados serão armazenados somente no bucket do S3. Para obter informações sobre como incluir dados no relatório de sessões, consulte executeHttpStep(stepName, requestOptions, [callback], [stepConfig]).

Os tipos de conteúdo de corpo da resposta de texto, HTML e JSON são compatíveis. Tipos de conteúdo como text/HTML, text/plain, application/JSON e application/x-amz-json-1.0 são compatíveis. Respostas compactadas não são compatíveis.

- A guia Monitoring (Monitoramento) exibe gráficos das métricas do CloudWatch publicadas por esse canário. Para ter mais informações sobre essas métricas, consulte Métricas do CloudWatch publicadas por canaries.

  Abaixo dos gráficos do CloudWatch publicados pelo canário estão gráficos das métricas do Lambda relacionadas ao código do Lambda do canário.
- A guia Configuration (Configuração) exibe informações de configuração e programação sobre o canário.
- A guia Groups (Grupos) exibe os grupos aos quais este canário está associado, se houver.
- A guia Tags exibe as etiquetas associadas ao canário.

Métricas do CloudWatch publicadas por canaries

Os canaries publicam as seguintes métricas no CloudWatch no namespace `CloudWatchSynthetics`. Para obter mais informações sobre como visualizar métricas do CloudWatch, consulte Visualizar métricas disponíveis.

Métrica	Descrição
`SuccessPercent`	A porcentagem das execuções desse canário que foram concluídas e não encontraram falhas. Dimensões válidas: CanaryName Estatística válida: média Unidades: percentual
`Duration`	A duração da execução do canário em milissegundos. Dimensões válidas: CanaryName Estatística válida: média Unidade: milissegundos
`Errors`	O número de vezes que o canário falhou ao executar o script completo. Dimensões válidas: CanaryName

Métrica	Descrição
	Estatística válida: soma
2xx	O número de solicitações de rede executadas pelo canário que retornaram respostas OK, com códigos de resposta entre 200 e 299. Essa métrica é relatada para canaries de interface do usuário que usam versão de runtime `syn-nodejs-2.0` ou posteriores, e é informada para canaries de API que usam a versão de runtime `syn-nodejs-2.2` ou posteriores. Dimensões válidas: CanaryName Estatística válida: soma Unidades: contagem
4xx	O número de solicitações de rede executadas pelo canário que retornaram respostas Error, com códigos de resposta entre 400 e 499. Essa métrica é relatada para canaries de interface do usuário que usam versão de runtime `syn-nodejs-2.0` ou posteriores, e é informada para canaries de API que usam a versão de runtime `syn-nodejs-2.2` ou posteriores. Dimensões válidas: CanaryName Estatística válida: soma Unidades: contagem

Métrica	Descrição
`5xx`	O número de solicitações de rede executadas pelo canário que retornara m respostas Fault, com códigos de resposta entre 500 e 599. Essa métrica é relatada para canaries de interface do usuário que usam versão de runtime `syn-nodejs-2.0` ou posteriores, e é informada para canaries de API que usam a versão de runtime `syn-nodejs-2.2` ou posteriores. Dimensões válidas: CanaryName Estatística válida: soma Unidades: contagem
`Failed`	O número de execuções do canário que falharam na execução. Essas falhas estão relacionadas ao próprio canário. Dimensões válidas: CanaryName Estatística válida: soma Unidades: contagem
`Failed requests`	O número de solicitações HTTP executadas pelo canário no site de destino que falharam sem resposta. Dimensões válidas: CanaryName Estatística válida: soma Unidades: contagem

Métrica	Descrição
`VisualMonitoringSuccessPercent`	A porcentagem de comparações visuais que corresponderam com êxito às capturas de tela da linha de base durante uma execução do canário. Dimensões válidas: CanaryName Estatística válida: média Unidades: percentual
`VisualMonitoringTotalComparisons`	O número total de comparações visuais que ocorreram durante uma execução do canário. Dimensões válidas: CanaryName Unidades: contagem

> **Note**
>
> Canaries que usam os métodos `executeStep()` ou `executeHttpStep()` da biblioteca do Synthetics também publicam métricas `SuccessPercent` e `Duration` com as dimensões `CanaryName` e `StepName` para cada etapa.

Editar ou excluir um canário

É possível editar ou excluir um canário existente.

Edit canary (Editar canário)

Quando você edita um canário, mesmo sem alterar a agendamento, o agendamento é redefinido para corresponder ao momento em que você edita o canário. Por exemplo, se você tiver um canário que é executado a cada hora e editar esse canário, ele será executado imediatamente após a conclusão da edição e a cada hora depois disso.

Para editar ou atualizar um canário

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

2. No painel de navegação, escolha Application Signals, Canários do Synthetics.
3. Selecione o botão ao lado do nome do canário e escolha Actions (Ações), Edit (Editar).
4. (Opcional) Se esse canário executar o monitoramento visual das capturas de tela e você quiser definir a próxima execução do canário como linha de base, selecione Set next run as new baseline (Definir a próxima execução como nova linha de base).
5. (Opcional) Se esse canário executar o monitoramento visual de capturas de tela e você desejar remover uma captura de tela do monitoramento visual ou designar partes da captura de tela a serem ignoradas durante comparações visuais, em Visual Monitoring (Monitoramento visual), escolha Edit Baseline (Editar linha de base).

   A captura de tela será exibida, e você poderá executar uma das seguintes ações:

   - Para remover a captura de tela de ser usada para monitoramento visual, selecione Remove screenshot from visual test baseline (Remover captura de tela da linha de base do teste visual).
   - Para designar partes da captura de tela a serem ignoradas durante comparações visuais, clique e arraste para traçar as áreas da tela a serem ignoradas. Depois de ter feito isso para todas as áreas que deseja ignorar durante as comparações, escolha Save (Salvar).
6. Faça todas as outras alterações que desejar no canário e escolha Save (Salvar).

Delete canary (Excluir canário)

Ao excluir um canário, você pode escolher se também deseja excluir outros recursos usados e criados pelo canário. Ao excluir um canário, também será necessário excluir o seguinte:

- As camadas e as funções do Lambda usadas por esse canário. O prefixo é `cwsyn-`*`MyCanaryName`*.
- Os alarmes do CloudWatch criados para esse canário. Esses alarmes têm um nome que começa com `Synthetics-Alarm-`*`MyCanaryName`*. Para obter mais informações sobre como excluir alarmes, consulte Como editar ou excluir um alarme do CloudWatch.
- Os objetos e os buckets do Amazon S3, como os locais de resultados e os locais de artefatos do canário.
- As funções do IAM criadas para o canário. Elas têm o nome `role/service-role/CloudWatchSyntheticsRole-`*`MyCanaryName`*.
- Grupos de logs do CloudWatch Logs criados para o canário. Esses grupos de logs têm os seguintes nomes: `/aws/lambda/cwsyn-`*`MyCanaryName`*`-`*`randomId`*.

Antes de excluir um canário, convém visualizar os detalhes do canário e anotar essas informações. Dessa forma, é possível excluir os recursos corretos depois de excluir o canário.

Para excluir um canário

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals, Canários do Synthetics.
3. Se atualmente o canário estiver no estado `RUNNING`, você deve interrompê-lo. Só é possível excluir canaries nos estados `STOPPED`, `READY(NOT_STARTED)` ou `ERROR`.

   Para interromper o canário, selecione o botão ao lado do nome do canário e escolha Actions (Ações), Stop (Parar).
4. Selecione o botão ao lado do nome do canário e escolha Actions (Ações), Delete (Excluir).
5. Escolha se também deseja excluir os outros recursos criados para e usados pelo canário. Isso inclui as camadas e a função do Lambda, além da função do IAM e da política do IAM do canário.

   Para excluir a função do IAM e a política do IAM do canário, você precisa ter permissões suficientes. Para ter mais informações, consulte Políticas gerenciadas (predefinidas) pela AWS para o CloudWatch Synthetics.
6. Insira **`Delete`** na caixa e escolha Delete (Excluir).
7. Exclua os outros recursos usados e criados para o canário, conforme listado anteriormente nesta seção.

Iniciar, interromper, excluir ou atualizar o runtime de vários canários

Você pode parar, iniciar, excluir ou atualizar o runtime de até cinco canários com uma única ação. Se você atualizar o runtime de um canário, ele será atualizado para o runtime mais recente disponível para o idioma e a estrutura que o canário usa.

Se você selecionar vários canários e apenas alguns deles estiverem em um estado válido para a ação selecionada, a ação será executada somente nos canários em que a ação for válida. Por exemplo, se você selecionar alguns canários que estão em execução no momento e outros que não estão, e selecionar a opção de iniciar os canários, os canários que ainda não estavam em execução serão iniciados e os canários que já estavam em execução não serão afetados.

Se nenhum dos canários que você selecionar for válido para uma ação, a ação não estará disponível no menu.

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals, Canários do Synthetics.
3. Marque as caixas de seleção ao lado dos canários que você deseja parar, iniciar ou excluir.
4. Escolha Ações e Iniciar, Interromper, Excluir ou Atualizar runtime.

Monitorar eventos do canário com o Amazon EventBridge

As regras de evento do Amazon EventBridge podem notificar você quando os canaries alterarem o status ou concluírem execuções. O Eventbridge oferece um fluxo quase em tempo real dos eventos do sistema que descrevem as alterações nos recursos da AWS. O CloudWatch Synthetics envia esses eventos ao EventBridge na base do melhor esforço. A entrega do melhor esforço significa que o CloudWatch Synthetics tenta enviar todos os eventos ao EventBridge, mas, em alguns casos raros, o evento poderá não ser entregue. O EventBridge processa, pelo menos uma vez,todos os eventos recebidos. Além disso, os listeners do evento poderão não receber os eventos na ordem em que estes ocorreram.

> **Note**
>
> O Amazon EventBridge é um serviço de barramento de eventos que você pode usar para facilitar a conexão de aplicações a dados de diversas origens. Para obter mais informações, consulte O que é o Amazon EventBridge? no Manual do usuário do Amazon EventBridge.

O CloudWatch Synthetics emite um evento quando um canário altera o estado ou conclui uma execução. É possível criar uma regra EventBridge que inclua um padrão de evento para corresponder a todos os tipos de eventos enviados do CloudWatch Synthetics ou que corresponda apenas a tipos de eventos específicos. Quando um canário aciona uma regra, o EventBridge invoca as ações de destino definidas na regra. Isso permite enviar notificações, capturar informações de eventos e executar ações corretivas em resposta a uma alteração de estado ou a conclusão de uma execução do canário. Por exemplo, você pode criar regras para os seguintes casos de uso:

- Investigar quando uma execução do canário falha
- Investigar quando um canário passou para o estado o `ERROR`

- Rastrear o ciclo de vida de um canário
- Monitorar o sucesso ou falha de execução do canário como parte de um fluxo de trabalho

Exemplos de eventos do CloudWatch Synthetics

Esta seção lista exemplos de eventos do CloudWatch Synthetics. Para obter mais informações sobre formato de eventos, consulte Eventos e padrões de eventos no EventBridge.

Alteração do estado do canário

Neste tipo de evento, os valores de `current-state` e `previous-state` podem ser:

`CREATING` | `READY` | `STARTING` | `RUNNING` | `UPDATING` | `STOPPING` | `STOPPED` | `ERROR`


```
{
                "version": "0",
                "id": "8a99ca10-1e97-2302-2d64-316c5dedfd61",
                "detail-type": "Synthetics Canary Status Change",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:19:43Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "EXAMPLE-dc5a-4f5f-96d1-989b75a94226",
                                "canary-name": "events-bb-1",
                                "current-state": "STOPPED",
                                "previous-state": "UPDATING",
                                "source-location": "NULL",
                                "updated-on": 1612909161.767,
                                "changed-config": {
                                                "executionArn": {
                                                                "previous-value":
 "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:1",
                                                                "current-value":
 "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:2"
                                                },
                                                "vpcId": {
                                                                "current-value": "NULL"
```

```
                                        },
                                        "testCodeLayerVersionArn": {
                                                "previous-
value": "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:1",
                                                "current-value":
 "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:2"
                                        }
                                },
                                "message": "Canary status has changed"
                }
}
```

Execução do canário concluída com êxito

```
{
                "version": "0",
                "id": "989EXAMPLE-f4a5-57a7-1a8f-d9cc768a1375",
                "detail-type": "Synthetics Canary TestRun Successful",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:24:01Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "989EXAMPLE-dc5a-4f5f-96d1-989b75a94226",
                                "canary-name": "events-bb-1",
                                "canary-run-id": "c6c39152-8f4a-471c-9810-989EXAMPLE",
                                "artifact-location": "cw-syn-results-123456789012-us-
east-1/canary/us-east-1/events-bb-1-ec3-28ddbe266797/2021/02/09/22/23-41-200",
                                "test-run-status": "PASSED",
                                "state-reason": "null",
                                "canary-run-timeline": {
                                        "started": 1612909421,
                                        "completed": 1612909441
                                },
                                "message": "Test run result is generated successfully"
                }
}
```

Execução do canário concluída com falha

```
{
                "version": "0",
                "id": "2644b18f-3e67-5ebf-cdfd-bf9f91392f41",
                "detail-type": "Synthetics Canary TestRun Failure",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:24:27Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "af3e3a05-dc5a-4f5f-96d1-9989EXAMPLE",
                                "canary-name": "events-bb-1",
                                "canary-run-id": "0df3823e-7e33-4da1-8194-
b04e4d4a2bf6",
                                "artifact-location": "cw-syn-results-123456789012-us-
east-1/canary/us-east-1/events-bb-1-ec3-989EXAMPLE/2021/02/09/22/24-21-275",
                                "test-run-status": "FAILED",
                                "state-reason": "\"Error: net::ERR_NAME_NOT_RESOLVED
 \""
                                "canary-run-timeline": {
                                                "started": 1612909461,
                                                "completed": 1612909467
                                },
                                "message": "Test run result is generated successfully"
                }
}
```

É possível que os eventos estejam duplicados ou fora de ordem. Para determinar a ordem dos eventos, use a propriedade `time`.

Pré-requisitos para criar regras do EventBridge

Antes de criar uma regra do EventBridge para o CloudWatch Synthetics, é necessário:

- Familiarize-se com os eventos, regras e destinos no EventBridge.
- Crie e configure os destinos invocados por suas regras do EventBridge. As regras podem invocar muitos tipos de destinos, incluindo:
  - Tópicos do Amazon SNS
  - Funções do AWS Lambda
  - Streams do Kinesis

- Filas do Amazon SQS

Para obter mais informações, consulte O que é o Amazon EventBridge e Começar a usar o Amazon EventBridge no Manual do usuário do Amazon EventBridge.

Criar uma regra do EventBridge (CLI)

As etapas no exemplo a seguir criam uma regra EventBridge que publica um tópico do Amazon SNS quando o canário chamado `my-canary-name` em `us-east-1` conclui uma execução ou altera o estado.

1. Crie a regra.

```
aws events put-rule \
  --name TestRule \
  --region us-east-1 \
  --event-pattern "{\"source\": [\"aws.synthetics\"], \"detail\": {\"canary-name\":
 [\"my-canary-name\"]}}"
```

   As propriedades que você omitir do padrão serão ignoradas.

2. Adicione o tópico como um destino de regra.

   - Substitua *topic-arn* pelo nome do recurso da Amazon (ARN) do tópico do Amazon SNS.

```
aws events put-targets \
  --rule TestRule \
  --targets "Id"="1","Arn"="topic-arn"
```

> **Note**
>
> Para permitir que o Amazon EventBridge chame o tópico de destino, você deve adicionar uma política baseada em recursos ao tópico. Para obter mais informações, consulte Permissões do Amazon SNS no Manual do usuário do Amazon EventBridge.

Para obter mais informações, consulte Eventos e padrões de eventos no EventBridge no Manual do usuário do Amazon EventBridge.

Execução de lançamentos e experimentos A/B com o CloudWatch Evidently

Você pode usar o Amazon CloudWatch Evidently para validar novos recursos com segurança, oferecendo-os a uma porcentagem especificada de seus usuários enquanto você implementa o recurso. Você pode monitorar a performance do novo recurso para auxiliar na decisão de quando aumentar o tráfego para seus usuários. Isso ajuda você a reduzir riscos e identificar consequências não intencionais antes de iniciar totalmente o recurso.

Você também pode realizar experimentos A/B para decidir sobre design de recursos com base em evidências e dados. Um experimento pode testar até cinco variações ao mesmo tempo. O Evidently coleta dados do experimento e os analisa usando métodos estatísticos. Ele também fornece recomendações claras sobre quais variações têm melhor performance. Você pode testar recursos voltados para o usuário e recursos de backend.

Definição de preço do Evidently

O Evidently cobra sua conta com base em eventos Evidently e unidades de análise Evidently. Os eventos do Evidently, incluem tanto eventos de dados, como cliques e visualizações de página, como eventos de atribuição que determinam a variação de recursos a ser veiculada a um usuário.

As unidades de análise do Evidently são geradas a partir de eventos do Evidently com base nas regras criadas no Evidently. As unidades de análise são o número de correspondências de regras em eventos. Por exemplo, um evento de clique do usuário pode produzir uma única unidade de análise Evidently, uma contagem de cliques. Outro exemplo é um evento de checkout do usuário que pode produzir duas unidades de análise Evidently, valor de checkout e o número de itens no carrinho. Para obter mais informações sobre a definição de preço, consulte Preços do Amazon CloudWatch.

No momento, o CloudWatch Evidently está disponível nas seguintes regiões:

- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)

- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Estocolmo)

Tópicos

Políticas do IAM para usar o Evidently

Para gerenciar totalmente o CloudWatch Evidently, você deve estar conectado como um usuário ou função do IAM que tenha as seguintes permissões:

- A política AmazonCloudWatchEvidentlyFullAccess
- A política ResourceGroupsandTagEditorReadOnlyAccess

Além disso, para poder criar um projeto que armazene eventos de avaliação no Amazon S3 ou no CloudWatch Logs, você precisa das seguintes permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Action": [
                "s3:GetBucketPolicy",
                "s3:PutBucketPolicy",
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:PutResourcePolicy"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Permissões adicionais para integração com o CloudWatch RUM

Além disso, se você pretende gerenciar lançamentos ou experimentos Evidently que se integram ao Amazon CloudWatch RUM e usar métricas de RUM do CloudWatch para monitoramento, você precisa da política AmazonCloudWatchRUMFullAccess. Para criar uma função do IAM para dar ao cliente da Web CloudWatch RUM permissão para enviar dados ao CloudWatch RUM, você precisa das seguintes permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchRUMEvidentlyRole-*",
```

```
                "arn:aws:iam::*:policy/service-role/CloudWatchRUMEvidentlyPolicy-*"
            ]
        }
    ]
}
```

Para dar permissão somente leitura para acesso ao Evidently

Para outros usuários que precisam visualizar dados do Evidently, mas não precisam criar recursos do Evidently, você pode conceder a política AmazonCloudWatchEvidentlyReadOnlyAccess.

Criar projetos, recursos, lançamentos e experimentos

Para começar a usar o CloudWatch Evidently para um lançamento de recursos ou para um experimento A/B, você primeiro cria um projeto. Um projeto é um agrupamento lógico de recursos. Dentro do projeto, você cria recursos com variações que você deseja testar ou iniciar. Você pode criar um recurso antes ou ao mesmo tempo em que cria um lançamento ou experimento.

Tópicos

Criar um novo projeto da

Use estas etapas para configurar um novo projeto do CloudWatch Evidently.

Para criar um novo projeto do CloudWatch Evidently

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha Criar projeto.
4. Em Project name (Nome do projeto), insira um nome a ser usado para identificar esse projeto no console do CloudWatch Evidently.

Você pode adicionar uma descrição opcional do projeto.

5. Em Evaluation event storage (Armazenamento de eventos de avaliação), escolha se você deseja armazenar os eventos de avaliação que você coleta com Evidently. Mesmo que você não armazene esses eventos, o Evidently os agrega para criar métricas e outros dados de experimento que você pode visualizar no painel do Evidently. Para ter mais informações, consulte Armazenamento de dados do projeto.
6. Em Use client-side evaluation (Usar a avaliação do lado do cliente), escolha se deseja habilitar a avaliação do lado do cliente para esse projeto. Com a avaliação do lado do cliente, a aplicação pode atribuir variações às sessões do usuário localmente, em vez de chamar a operação EvaluateFeature. Isso reduz os riscos de latência e disponibilidade ocasionados por uma chamada de API. Para ter mais informações, consulte Usar a avaliação do lado do cliente baseada no AWS AppConfig.

   Para criar um projeto com avaliação do lado do cliente, é necessário ter a permissão `evidently:ExportProjectAsConfiguration`.

   Se você habilitar a avaliação do lado do cliente, faça também o seguinte:

   a. Escolha se deseja usar uma aplicação do AWS AppConfig existente ou crie uma nova.
   b. Escolha se deseja usar um ambiente do AWS AppConfig existente ou crie um novo.

      Para obter mais informações sobre aplicações e ambientes do AWS AppConfig, consulte Como o AWS AppConfig funciona.
7. (Opcional) Para adicionar etiquetas a esse projeto, escolha Tags (Etiquetas), Add new tag (Adicionar nova etiqueta).

   Em seguida, em Key (Chave), insira um nome para a etiqueta. É possível adicionar um valor opcional para a tag em Valor.

   Para adicionar outra etiqueta, escolha novamente Add new tag (Adicionar nova etiqueta).

   Para obter mais informações, consulte Etiquetar recursos da AWS.
8. Escolha Criar projeto.

Usar a avaliação do lado do cliente baseada no AWS AppConfig

Você pode usar avaliação do lado do cliente baseada no AWS AppConfig (avaliação do lado do cliente) em um projeto, o que permite que a aplicação atribua variações às sessões do usuário localmente, em vez de atribuir variações chamando a operação EvaluateFeature. Isso reduz os riscos de latência e disponibilidade ocasionados por uma chamada de API.

Para usar a avaliação do lado do cliente, anexe a extensão do Lambda do AWS AppConfig como uma camada para suas funções Lambda e configure as variáveis de ambiente. A avaliação do lado do cliente é executada como um processo paralelo no host local. Em seguida, você pode chamar as operações EvaluationFeature e PutProjectEvent para `localhost`. O processo de avaliação do lado do cliente lida com atribuição de variações, armazenamento em cache e sincronização de dados. Para obter mais informações sobre o AWS AppConfig, consulte Como o AWS AppConfig funciona.

Ao integrar com o AWS AppConfig, especifique um ID de aplicação do AWS AppConfig e um ID do ambiente AWS AppConfig para o Evidently. É possível usar o mesmo ID de aplicação e ID de ambiente em todos os projetos do Evidently.

Quando você cria um projeto com a avaliação do lado do cliente habilitada, o Evidently cria um perfil de configuração do AWS AppConfig para o projeto. O perfil de configuração de cada projeto será diferente.

Controle de acesso para avaliação do lado do cliente

A avaliação do lado do cliente do Evidently usa um mecanismo de controle de acesso diferente do resto do Evidently. É altamente recomendável entender isso para poder implementar as medidas de segurança adequadas.

Com o Evidently, você pode criar políticas do IAM que limitam as ações que um usuário pode realizar em recursos individuais. Por exemplo, você pode criar um perfil de usuário que impeça que o usuário tenha a ação EvaluateFeature. Para obter mais informações sobre as ações do Evidently que podem ser controladas com políticas do IAM, consulte Ações definidas pelo Amazon CloudWatch Evidently.

O modelo de avaliação do lado do cliente permite avaliações locais de recursos do Evidently que usam metadados do projeto. Um usuário de projeto com a avaliação do lado do cliente habilitada pode chamar a API EvaluateFeature para um endpoint de host local; essa chamada de API não chegará ao Evidently e não será autenticada pelas políticas do IAM do serviço do Evidently. Essa chamada será bem-sucedida mesmo que o usuário não tenha a permissão do IAM para usar a ação EvaluateFeature. No entanto, o usuário ainda precisará da permissão PutProjectEvents para que

o atendente armazene em buffer os eventos de avaliação ou eventos personalizados e transfira os dados para o Evidently de forma assíncrona.

Além disso, o usuário deve ter a permissão `evidently:ExportProjectAsConfiguration` para poder criar um projeto que use avaliação do lado do cliente. Isso ajuda a controlar o acesso às ações EvaluateFeature que são chamadas durante a avaliação do lado do cliente.

Se você não prestar atenção, o modelo de segurança de avaliação do lado do cliente poderá subverter as políticas definidas no restante do Evidently. Um usuário com a permissão `evidently:ExportProjectAsConfiguration` pode criar um projeto com a avaliação do lado do cliente habilitada e usar a ação EvaluateFeature para avaliação do lado do cliente com esse projeto, mesmo que a ação EvaluateFeature seja expressamente negada em uma política do IAM.

Comece a usar o Lambda

Atualmente, o Evidently oferece suporte à avaliação do lado do cliente usando um ambiente do AWS Lambda. Para começar, primeiro decida qual aplicação e ambiente do AWS AppConfig serão usados. Escolha uma aplicação e um ambiente existentes ou crie novos.

Os comandos de exemplo da AWS CLI do AWS AppConfig a seguir criam uma aplicação e um ambiente.

```
aws appconfig create-application --name YOUR_APP_NAME
```

```
aws appconfig create-environment --application-id YOUR_APP_ID --
name YOUR_ENVIRONMENT_NAME
```

Em seguida, crie um projeto do Evidently usando esses recursos do AWS AppConfig. Para ter mais informações, consulte Criar um novo projeto da .

A avaliação do lado do cliente é compatível com o Lambda usando uma camada do Lambda. Essa é uma camada pública que faz parte do `AWS-AppConfig-Extension`, uma extensão pública do AWS AppConfig criada pelo serviço AWS AppConfig. Para obter mais informações sobre camadas do Lambda, consulte Layer (Camada).

Para usar a avaliação do lado do cliente, é necessário adicionar essa camada à função Lambda e configurar as permissões e as variáveis de ambiente.

Como adicionar a camada do Lambda de avaliação do lado do cliente do Evidently à função Lambda e configurá-la

1. Crie uma função Lambda, caso ainda não tenha criado.
2. Adicione a camada de avaliação do lado do cliente à função. Você pode especificar seu ARN ou selecioná-lo na lista de camadas da AWS, caso ainda não tenha feito isso. Para obter mais informações, consulte Configurar funções para usar camadas e Versões disponíveis da extensão do Lambda para o AWS AppConfig.
3. Crie uma política do IAM chamada EvidentlyAppConfigCachingAgentPolicy com o seguinte conteúdo e anexe-a ao perfil de execução da função. Para obter mais informações, consulte Função de execução do Lambda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "appconfig:GetLatestConfiguration",
                "appconfig:StartConfigurationSession",
                "evidently:PutProjectEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Adicione a variável de ambiente necessária `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS` à função Lambda. Essa variável de ambiente especifica o mapeamento entre o projeto do Evidently e os recursos do AWS AppConfig.

   Se estiver usando essa função para um projeto do Evidently, defina o valor da variável de ambiente como: `applications/`*`APP_ID`*`/environments/`*`ENVIRONMENT_ID`*`/configurations/`*`PROJECT_NAME`*

   Se estiver usando essa função para vários projetos do Evidently, use vírgula para separar os valores, como neste exemplo: `applications/`*`APP_ID_1`*`/environments/`*`ENVIRONMENT_ID_1`*`/configurations/`*`PROJECT_NAME_1`*`,`

`applications/APP_ID_2/environments/ENVIRONMENT_ID_2/configurations/PROJECT_NAME_2`

5. (Opcional) Defina outras variáveis de ambiente. Para obter mais informações, consulte Configurar a extensão do Lambda para o AWS AppConfig.
6. Em sua aplicação, obtenha avaliações do Evidently localmente enviando `EvaluateFeature` para `localhost`.

   Exemplo do Python:

```
import boto3
from botocore.config import Config

def lambda_handler(event, context):
    local_client = boto3.client(
        'evidently',
        endpoint_url="http://localhost:2772",
        config=Config(inject_host_prefix=False)
    )
    response = local_client.evaluate_feature(
        project=event['project'],
        feature=event['feature'],
        entityId=event['entityId']
    )
    print(response)
```

   Exemplo de Node.js:

```
const AWS = require('aws-sdk');
const evidently = new AWS.Evidently({
    region: "us-west-2",
    endpoint: "http://localhost:2772",
    hostPrefixEnabled: false
});

exports.handler = async (event) => {

    const evaluation = await evidently.evaluateFeature({
        project: 'John_ETCProject_Aug2022',
        feature: 'Feature_IceCreamFlavors',
        entityId: 'John'
    }).promise()
```

```
    console.log(evaluation)
    const response = {
        statusCode: 200,
        body: evaluation,
    };
    return response;
};
```

Exemplo de Kotlin:

```
String localhostEndpoint = "http://localhost:2772/"
public AmazonCloudWatchEvidentlyClient getEvidentlyLocalClient() {
    return AmazonCloudWatchEvidentlyClientBuilder.standard()

 .withEndpointConfiguration(AwsClientBuilder.EndpointConfiguration(localhostEndpoint,
 region))

 .withClientConfiguration(ClientConfiguration().withDisableHostPrefixInjection(true))
        .withCredentials(credentialsProvider)
        .build();
}

AmazonCloudWatchEvidentlyClient evidently = getEvidentlyLocalClient();

// EvaluateFeature via local client.
EvaluateFeatureRequest evaluateFeatureRequest = new
 EvaluateFeatureRequest().builder()
   .withProject(${YOUR_PROJECT}) //Required.
   .withFeature(${YOUR_FEATURE}) //Required.
   .withEntityId(${YOUR_ENTITY_ID}) //Required.
   .withEvaluationContext(${YOUR_EVAL_CONTEXT}) //Optional: a JSON object of
 attributes that you can optionally pass in as part of the evaluation event sent to
 Evidently.
   .build();

EvaluateFeatureResponse evaluateFeatureResponse =
 evidently.evaluateFeature(evaluateFeatureRequest);

// PutProjectEvents via local client.
PutProjectEventsRequest putProjectEventsRequest = new
 PutProjectEventsRequest().builder()
    .withData(${YOUR_DATA})
```

```
    .withTimeStamp(${YOUR_TIMESTAMP})
    .withType(${YOUR_TYPE})
    .build();

PutProjectEvents putProjectEventsResponse =
 evidently.putProjectEvents(putProjectEventsRequest);
```

Configure a frequência com que o cliente enviará dados ao Evidently

Para especificar com que frequência a avaliação do lado do cliente enviará dados ao Evidently, você pode configurar opcionalmente duas variáveis de ambiente.

- `AWS_APPCONFIG_EXTENSION_EVIDENTLY_EVENT_BATCH_SIZE` especifica o número de eventos por projeto a serem agrupados em lote antes de enviá-los ao Evidently. Os intervalos válidos são valores decimais entre 1 e 50, e o padrão é 40.
- `AWS_APPCONFIG_EXTENSION_EVIDENTLY_BATCH_COLLECTION_DURATION` especifica a duração, em segundos, para aguardar os eventos antes de enviá-los ao Evidently. O padrão é 30.

Solução de problemas

Use as informações a seguir para ajudar a solucionar problemas com o uso do CloudWatch Evidently com avaliação do lado do cliente baseado no AWS AppConfig.

Ocorreu um erro (BadRequestException) ao chamar a operação EvaluateFeature: HTTP method not supported for provided path (Não há suporte para o método HTTP para o caminho fornecido)

As variáveis de ambiente podem estar configuradas incorretamente. Por exemplo, você pode ter usado `EVIDENTLY_CONFIGURATIONS` como nome da variável de ambiente em vez de `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS`.

ResourceNotFoundException: implantação não encontrada

A atualização dos metadados do projeto não foi implantada no AWS AppConfig. Verifique se há uma implantação ativa no ambiente do AWS AppConfig que você usou para avaliação do lado do cliente.

ValidationException: não há configuração do Evidently para o projeto

A variável de ambiente `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS` pode estar configurada com o nome de projeto incorreto.

Adicionar um recurso a um projeto

Um recurso no CloudWatch Evidently representa um recurso que você deseja iniciar ou testar variações.

Para adicionar um recurso, você deve criar um projeto. Para ter mais informações, consulte Criar um novo projeto da .

Para adicionar um recurso a um projeto

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha o nome do projeto.
4. Escolha Add feature (Adicionar recurso).
5. Em Feature name (Nome do recurso), insira um nome a ser usado para identificar esse recurso neste projeto.

   Você pode adicionar uma descrição opcional do recurso.
6. Em Feature variations (Variações de recursos), em Variation type (Tipo de variação) escolha Boolean (Booliano), Long (Longo), Double (Duplo), ou String. Para ter mais informações, consulte Tipos de variação.
7. Adicione até cinco variações para seu recurso. O Value (Valor) de cada variação deve ser válido para o Variation type (Tipo de variação) que você selecionou.

   Especifique uma das variações como padrão. Essa será a linha de base com a qual as outras variações serão comparadas e deve ser a variação que está sendo distribuída aos seus usuários no momento. Essa também é a variação que é veiculada para usuários que não são adicionados a um lançamento ou a um experimento desse recurso.
8. Escolha Sample code (Código de exemplo). O código de exemplo mostra o que você precisa adicionar à aplicação para configurar as variações e atribuir sessões de usuário a elas. Você pode escolher entre JavaScript, Java e Python para o código.

   Você não precisa adicionar o código à sua aplicação imediatamente, mas deve fazê-lo antes de iniciar um lançamento ou um experimento.

   Para ter mais informações, consulte Adicionar código à sua aplicação.
9. (Opcional) Para especificar que determinados usuários sempre vejam uma determinada variação, escolha Overrides (Substituições), Add override (Adicionar substituição). Em seguida,

especifique um usuário inserindo o ID de usuário, o ID da conta ou algum outro identificador no Identifier (Identificador) e especifique qual variação eles devem ver.

Isso é ser útil quando você quer garantir que integrantes da sua própria equipe de testes ou usuários internos vejam uma variação específica. As sessões dos usuários a quem são atribuídas substituições não contribuem para as métricas de início ou experimento.

É possível repetir isso para até 20 usuários escolhendo Adicionar substituição novamente.

10. (Opcional) Para adicionar etiquetas a esse recurso, escolha Tags (Etiquetas), Add new tag (Adicionar nova etiqueta).

    Em seguida, em Key (Chave), insira um nome para a etiqueta. É possível adicionar um valor opcional para a tag em Valor.

    Para adicionar outra etiqueta, escolha novamente Add new tag (Adicionar nova etiqueta).

    Para obter mais informações, consulte Etiquetar recursos da AWS.
11. Escolha Add feature (Adicionar recurso).

Tipos de variação

Ao criar um recurso e definir as variações, você deve selecionar um recurso, você deve selecionar um variation type (tipo de variação). Os tipos possíveis são:

- Booleano
- Inteiro longo
- Número de ponto flutuante de precisão dupla
- String

O tipo de variação define como as distintas variações são diferenciadas em seu código. Você pode usar o tipo de variação para simplificar a implementação do CloudWatch Evidently e também para simplificar o processo de modificação dos recursos em seus lançamentos e experimentos.

Por exemplo, se você definir um recurso com o tipo de variação inteiro longo, os números inteiros especificados para diferenciar as variações poderão ser passados diretamente para o código. Um exemplo é o teste do tamanho do pixel de um botão. Os valores para os tipos de variação podem ser o número de pixels usados em cada variação. O código para cada variação pode ler o valor do tipo

de variação e usá-lo como o tamanho do botão. Para testar um novo tamanho de botão, você pode alterar o número usado para o valor da variação sem fazer nenhuma outra alteração de código.

Ao definir os valores para seus tipos de variação dentro de um recurso, deve-se evitar atribuir os mesmos valores a múltiplas variações, a menos que você queira fazer testes A/A para testar inicialmente o CloudWatch Evidently ou tenha outros motivos para fazê-lo.

O Evidently não tem suporte nativo para JSON como um tipo, mas você pode passar o JSON no tipo de variação String e analisar esse JSON em seu código.

Usar segmentos para delimitar o público

Você pode definir segmentos de público e usá-los em seus lançamentos e experimentos. O segmento é uma parte do público que compartilha uma ou mais características. Exemplos podem ser usuários do navegador Chrome, usuários na Europa ou usuários do navegador Firefox na Europa que também atendam a outro critério que sua aplicação coleta, como idade.

Usar um segmento no experimento limita a avaliação apenas para usuários que correspondam aos critérios do segmento. Ao usar um ou mais segmentos em uma execução, você pode definir diferentes divisões de tráfego para os diferentes segmentos de público.

Sintaxe de padrões de regras de segmento

Para criar um segmento, defina uma regra de segmento pattern. Especifique os atributos que deseja usar para avaliar se uma sessão de usuário será incluída no segmento. O padrão que você cria é comparado ao valor de `evaluationContext` que o Evidently encontra em uma sessão de usuário. Para ter mais informações, consulte Utilizar a operação EvaluateFeature.

Para criar um padrão de regra de segmento, especifique os campos aos quais o padrão deve corresponder. Você também pode usar lógica no padrão, como `And`, `Or`, `Not` e `Exists`.

Para que `evaluationContext` corresponda a um padrão, `evaluationContext` deve corresponder a todas as partes do padrão da regra. O Evidently ignora os campos no `evaluationContext` que não estão incluídos no padrão da regra.

Os valores de correspondência nos padrões de regras seguem regras JSON. Você pode incluir strings entre aspas (“), números e as palavras-chave `true`, `false` e `null`.

No caso de strings, o Evidently usa correspondência exata caractere por caractere sem conversão de caixa alta/baixa ou normalização de strings. Portanto, as correspondências de regras diferenciam

maiúsculas de minúsculas. Por exemplo, se `evaluationContext` incluir um atributo `browser`, mas seu padrão de regra verificar `Browser`, não haverá correspondência.

No caso de números, o Evidently usa representação de string. Por exemplo, 300, 300.0 e 3.0e2 não são considerados iguais.

Quando você grava padrões de regras para corresponder `evaluationContext`, pode usar a API `TestSegmentPattern` ou o comando da CLI `test-segment-pattern` para garantir que o padrão corresponda ao JSON correto. Para obter mais informações, consulte TestSegmentPattern.

O resumo a seguir exibe todos os operadores de comparação que estão disponíveis nos padrões de segmentos do Evidently.

Comparação	Exemplo	Sintaxe da regra
Nulo	UserID é null	`{ "UserID": [ null ] }`
Vazio	LastName está vazio	`{ "LastName": [""] }`
Igual	“Browser” (Navegador) é “Chrome”	`{ "Browser": [ "Chrome" ] }`
E	“Country” (País) é “França” e “Device” (Dispositivo) é “Móvel”	`{ "Country": [ "France" ], "Device": ["Mobile"] }`
Ou (vários valores de um único atributo)	“Browser” (Navegador) é “Chrome” ou “Firefox”	`{ "Browser": ["Chrome", "Firefox"]`

Comparação	Exemplo	Sintaxe da regra
		} }

Ou (atributos diferentes)	“Browser” (Navegador) é “Safari” ou “Device” (Dispositivo) é “Tablet”	{
		"$or": [
		{"Browser": ["Safari"]},
		{"Device": ["Tablet"}]
		]
		}

Não	“Browser” (Navegador) é qualquer valor exceto “Safari”	{
		"Browser": [{ "anything-but": ["Safari"] }]
		}

Numérico (é iguais a)	Price é 100	{
		"Price": [{ "numeric": ["=", 100] }]
		}

Numérico (intervalo)	Price é superior a 10 e menor que ou igual a 20	{
		"Price": [{ "numeric": [">", 10, "<=", 20] }]
		}

Existe	Existe um campo “Age” (Idade)	{
		"Age": [{ "exists": true }]
		}

Comparação	Exemplo	Sintaxe da regra
Não existe	Não existe um campo “Age” (Idade)	`{ "Age": [ { "exists": false } ] }`
Começa com um prefixo	“Regio” (Região) está nos Estados Unidos	`{ "Region": [ {"prefix": "us-" } ] }`
Termina com um sufixo	A localização tem um sufixo “West” (Oeste)	`{ "Region": [ {"suffix": "West" } ] }`

Exemplo de regras de segmentos

Todos os exemplos a seguir pressupõem que você está passando valores para `evaluationContext` com os mesmos rótulos de campo e valores que está usando nos padrões de regras.

O exemplo a seguir encontrará uma correspondência se o `Browser` for Chrome ou Firefox e o `Location` for US-West (Oeste dos EUA).

```
{
  "Browser": ["Chrome", "Firefox"],
  "Location": ["US-West"]
}
```

O exemplo a seguir encontrará uma correspondência se `Browser` for qualquer navegador exceto Chrome, `Location` começar com US e houver um campo `Age` existente.

```
{
  "Browser": [ {"anything-but": ["Chrome"]}],
  "Location": [{"prefix": "US"}],
```

```
  "Age": [{"exists": true}]
}
```

O exemplo a seguir encontrará uma correspondência se `Location` for “Japan” (Japão) e `Browser` for Safari ou `Device` for Tablet.

```
{
 "Location": ["Japan"],
 "$or": [
   {"Browser": ["Safari"]},
   {"Device": ["Tablet"]}
 ]
}
```

Criar um segmento

Após criar um segmento, você poderá usá-lo em qualquer execução ou experimento em qualquer projeto.

Para criar um segmento

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha a guia Segments (Segmentos).
4. Selecione Criar um segmento.
5. Em Segment name (Nome do segmento), insira o nome a ser usado para identificar o segmento.

   Opcionalmente, adicione uma descrição.
6. Em Segment pattern (Padrão de segmento), insira um bloco JSON para definir o padrão de regra. Para obter mais informações sobre a sintaxe do padrão de regras, consulte Sintaxe de padrões de regras de segmento.

Criar um lançamento

Para expor um novo recurso ou alterar para uma porcentagem especificada de seus usuários, crie um lançamento. Em seguida, você pode monitorar métricas importantes, como tempos de carregamento de página e conversões, antes de implantar o recurso para todos os usuários.

Antes de adicionar um lançamento, você deve criar um projeto. Para ter mais informações, consulte Criar um novo projeto da .

Ao adiciona um lançamento, você pode usar um recurso já criado ou criar um novo recurso ao mesmo tempo em que cria o lançamento.

Para adicionar um lançamento a um projeto

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Selecione o botão ao lado do nome do projeto e escolha Project actions (Ações do projeto), Create launch (Criar Lançamento).
4. Em Launch name (Nome do lançamento), insira um nome a ser usado para identificar esse recurso neste projeto.

   Você pode adicionar uma descrição opcional.
5. Escolha entre Select from existing features (Selecionar entre os recursos existentes) ou Add new feature (Adicionar novo recurso).

   Se você estiver usando um recurso existente, selecione-o em Feature name (Nome do recurso).

   Se você escolher Add new feature (Adicionar novo recurso), faça o seguinte:

   a. Em Feature name (Nome do recurso), insira um nome a ser usado para identificar esse recurso neste projeto. Você pode adicionar uma descrição opcional.
   b. Em Feature variations (Variações de recursos), em Variation type (Tipo de variação) escolha Boolean (Booliano), Long (Longo), Double (Duplo), ou String. Para ter mais informações, consulte Tipos de variação.
   c. Adicione até cinco variações para seu recurso. O Value (Valor) de cada variação deve ser válido para o Variation type (Tipo de variação) que você selecionou.

      Especifique uma das variações como padrão. Essa será a linha de base com a qual as outras variações serão comparadas e deve ser a variação que está sendo distribuída aos seus usuários no momento. Se você interromper um experimento, essa variação padrão será veiculada a todos os usuários.
   d. Escolha Sample code (Código de exemplo). O código de exemplo mostra o que você precisa adicionar à aplicação para configurar as variações e atribuir sessões de usuário a elas. Você pode escolher entre JavaScript, Java e Python para o código.

Você não precisa adicionar o código à sua aplicação neste momento, mas deve fazê-lo antes de iniciar o lançamento.

Para ter mais informações, consulte Adicionar código à sua aplicação.

6. Em Launch configuration (Configuração de execução), escolha se deseja iniciar o lançamento imediatamente ou agendá-lo para começar mais tarde.
7. (Opcional) Para especificar outras divisões de tráfego para os segmentos de público que você definiu diferentes da divisão usada para o público geral, escolha Add Segment Overrides (Adicionar substituições de segmento).

   Em Segment Overrides (Substituições de segmento), selecione um segmento e defina a divisão de tráfego a ser usada para esse segmento.

   Você também pode definir mais segmentos para definir divisões de tráfego escolhendo Add Segment Override (Adicionar substituição de segmento). Um lançamento pode ter até seis substituições de segmento.

   Para ter mais informações, consulte Usar segmentos para delimitar o público.
8. Em Traffic configuration (Configuração de tráfego), selecione a porcentagem de tráfego a ser atribuída a cada variação para o público geral que não corresponda às substituições de segmentos. Você também pode optar por excluir a distribuição de variações aos usuários.

   O Traffic summary (Resumo do tráfego) mostra quanto de seu tráfego geral está disponível para este lançamento.
9. Se você optar por agendar o lançamento para começar mais tarde, poderá adicionar várias etapas. Cada etapa pode usar porcentagens diferentes para veicular as variações. Para fazer isso, escolha Add another step (Adicionar outra etapa) e, em seguida, especifique as porcentagens de agendamento e de tráfego para a próxima etapa. É possível incluir até cinco etapas em um lançamento.
10. Se você quiser acompanhar a performance do recurso com métricas durante o lançamento, escolha Metrics (Métricas), Add Metric (Adicionar métricas). Você pode usar métricas do CloudWatch RUM ou métricas personalizadas.

    Para usar uma métrica personalizada, você pode criar a métrica aqui usando uma regra do Amazon EventBridge. Para criar uma métrica personalizada, faça o seguinte:

    - EscolhaCustom metrics (Métricas personalizadas) e insira um nome para a métrica.

- Em Metric rule (Regra de métrica), em Entity ID (ID da entidade), insira a maneira de identificar a entidade. Ela pode ser um usuário ou sessão que executa uma ação que faz com que um valor de métrica seja registrado. Um exemplo é `userDetails.userID`.
- Em Value key (Valor da chave), insira o valor a ser rastreado para produzir a métrica.
- Opcionalmente, insira um nome para as unidades da métrica. Esse nome de unidade é apenas para fins de exibição, para uso em gráficos no console do Evidently.

À medida que você insere esses campos, a caixa mostra exemplos de como codificar a regra EventBridge para criar a métrica. Para obter mais informações sobre o EventBridge, consulte O que é o Amazon EventBridge?

Para usar métricas de RUM, você já deve ter um monitor de aplicação RUM configurado para sua aplicação. Para ter mais informações, consulte Configurar uma aplicação para usar o CloudWatch RUM.

> **Note**
>
> Se você usar métricas do RUM e o monitor de aplicações não estiver configurado para provar 100% das sessões do usuário, nem todas as sessões do usuário que participam do lançamento enviarão métricas para o Evidently. Para garantir que as métricas de inicialização sejam precisas, recomendamos que o monitor de aplicações use 100% das sessões do usuário para amostragem.

11. (Opcional) Se você criar pelo menos uma métrica para o lançamento, poderá associar um alarme do CloudWatch existente a esse lançamento. Para fazer isso, escolhaAssociate CloudWatch alarms (Associar alarmes do CloudWatch).

    Quando você associa um alarme a um lançamento, o CloudWatch Evidently deve adicionar etiquetas com o nome do projeto e o nome de lançamento ao alarme. Isso ocorre para que o CloudWatch Evidently possa exibir os alarmes corretos nas informações de inicialização no console.

    Para reconhecer que o CloudWatch Evidently adicionará essas etiquetas, escolha Allow Evidently to tag the alarm resource identified below with this launch resource. (Permitir que o Evidently marque o recurso de alarme identificado abaixo com este recurso de lançamento). Em seguida, escolha Associate alarm (Alarme associado) e insira o nome do alarme.

Para obter informações sobre como criar alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch.

12. (Opcional) Para adicionar tags a esse lançamento, escolha Tags (Etiquetas), Add new tag (Adicionar nova etiqueta).

    Em seguida, em Key (Chave), insira um nome para a etiqueta. É possível adicionar um valor opcional para a tag em Valor.

    Para adicionar outra etiqueta, escolha novamente Add new tag (Adicionar nova etiqueta).

    Para obter mais informações, consulte Etiquetar recursos da AWS.
13. Escolha Create launch (Criar lançamento).

Criar um experimento

Use experimentos para testar diferentes versões de um recurso ou site e coletar dados de sessões reais do usuário. Dessa forma, é possível fazer escolhas para sua aplicação com base em evidências e dados.

Antes de adicionar um experimento, você deve criar um projeto. Para ter mais informações, consulte Criar um novo projeto da .

Ao adicionar um experimento, você pode usar um recurso que você já criou ou criar um novo recurso ao mesmo tempo em que cria o experimento.

Para adicionar um experimento a um projeto

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Selecione o botão ao lado do nome do projeto e escolha Project actions (Ações do projeto), Create experiment (Criar experimento).
4. Em Experiment name (Nome do experimento), insira um nome a ser usado para identificar esse recurso neste projeto.

   Você pode adicionar uma descrição opcional.
5. Escolha entre Select from existing features (Selecionar entre os recursos existentes) ou Add new feature (Adicionar novo recurso).

Se você estiver usando um recurso existente, selecione-o em Feature name (Nome do recurso).

Se você escolher Add new feature (Adicionar novo recurso), faça o seguinte:

a. Em Feature name (Nome do recurso), insira um nome a ser usado para identificar esse recurso neste projeto. Você também pode inserir uma descrição opcional.

b. Em Feature variations (Variações de recursos), em Variation type (Tipo de variação) escolha Boolean (Booliano), Long (Longo), Double (Duplo), ou String. O tipo define qual tipo de valor é usado para cada variação. Para ter mais informações, consulte Tipos de variação.

c. Adicione até cinco variações para seu recurso. O Value (Valor) de cada variação deve ser válido para o Variation type (Tipo de variação) que você selecionou.

   Especifique uma das variações como padrão. Essa será a linha de base com a qual as outras variações serão comparadas e deve ser a variação que está sendo distribuída aos seus usuários no momento. Se você interromper um experimento que usa esse recurso, a variação padrão será então veiculada para a porcentagem de usuários que estavam no experimento anteriormente.

d. Escolha Sample code (Código de exemplo). O código de exemplo mostra o que você precisa adicionar à aplicação para configurar as variações e atribuir sessões de usuário a elas. Você pode escolher entre JavaScript, Java e Python para o código.

   Não é necessário adicionar o código à sua aplicação neste momento, mas você deve fazê-lo antes de iniciar o experimento. Para ter mais informações, consulte Adicionar código à sua aplicação.

6. Em Audience (Público), você poderá selecionar um segmento que criou se quiser que esse experimento seja aplicado somente aos usuários que corresponderem a esse segmento. Para obter mais informações sobre segmentos, consulte Usar segmentos para delimitar o público.

7. Em Traffic split for the experiment (Divisão de tráfego para o experimento), especifique a porcentagem do público selecionado cujas sessões serão usadas no experimento. Em seguida, aloque o tráfego para as diferentes variações utilizadas no experimento.

   Se um lançamento e um experimento estiverem sendo executados para um mesmo recurso ao mesmo tempo, o público será direcionado primeiro para o lançamento. Em seguida, a porcentagem de tráfego especificada para o lançamento é retirada do público geral. Depois

disso, a porcentagem que você especifica aqui será a porcentagem do público restante usado para o experimento. Qualquer tráfego restante depois disso é recebe a variação padrão.

8. Em Metrics (Métricas) escolha as métricas a serem usadas para avaliar as variações durante o experimento. É necessário usar pelo menos uma métrica para avaliação.

   a. Em Metric source, (Fonte métrica), escolha se deseja usar métricas RUM do CloudWatch ou métricas personalizadas.

   b. Insira um nome para a métrica. Em Goal (Objetivo), escolha Increase (Aumentar) se você quiser que um valor mais alto de métrica indique uma variação melhor. Escolha Decrease (Diminuir) se você quiser que um valor mais baixo de métrica indique uma variação melhor.

   c. Se você estiver usando uma métrica personalizada, poderá criar a métrica aqui usando uma regra do Amazon EventBridge. Para criar uma métrica personalizada, faça o seguinte:

      - Em Metric rule (Regra de métrica), em Entity ID (ID da entidade), insira uma maneira de identificar a entidade. Ela pode ser um usuário ou uma sessão que executa uma ação que faz com que um valor de métrica seja registrado. Um exemplo é `userDetails.userID`.
      - Em Value key (Valor da chave), insira o valor a ser rastreado para produzir a métrica.
      - Opcionalmente, insira um nome para as unidades da métrica. Esse nome de unidade é apenas para fins de exibição, para uso em gráficos no console do Evidently.

      Você só pode usar métricas do RUM se tiver configurado o RUM para monitorar a aplicação. Para ter mais informações, consulte Usar o CloudWatch RUM.

      > **Note**
      >
      > Se você usar métricas do RUM e o monitor de aplicações não estiver configurado para provar 100% das sessões do usuário, nem todas as sessões do usuário no experimento enviarão métricas para o Evidently. Para garantir que as métricas do experimento sejam precisas, recomendamos que o monitor de aplicações use 100% das sessões do usuário para amostragem.

   d. (Opcional) Para adicionar mais métricas para avaliar, escolhaAdd Metric (Adicionar métrica). Você pode avaliar até três métricas durante o experimento.

9. (Opcional) Para criar alarmes do CloudWatch para usar com este experimento, escolha CloudWatch alarms (Alarmes do CloudWatch. Os alarmes podem monitorar se a diferença nos resultados entre cada variação e a variação padrão é maior do que um limite especificado por

você. Se a performance de uma variação for pior que a variação padrão e a diferença for maior que o limite, ela entrará no estado de alarme e o notificará.

A criação de um alarme aqui cria um alarme para cada variação que não seja a variação padrão.

Se você criar um alarme, especifique o seguinte:

- Em Metric name (Nome da métrica), escolha a métrica de experimento a ser usada para o alarme.
- Em Alarm condition (Condição de alarme), escolha qual condição irá fazer com que o alarme entre no estado de alarme, quando os valores da métrica de variação são comparados com os valores de métrica de variação padrão. Por exemplo, escolha Greater (Maior) ouGreater/Equal (Maior/igual) se números mais altos indicados para a variação indicarem que ela está com uma má performance. Isso é apropriado caso a métrica esteja medindo o tempo de carregamento da página, por exemplo.
- Insira um número para o limite, que é a diferença percentual na performance que fará com que o alarme entre no estado `ALARM`.
- Em Average over period (Média ao longo do período), escolha quantos dados de métrica para cada variação serão agregados juntos antes de serem comparados.

É possível escolher novamente Add new alarm (Adicionar novo alarme) para adicionar mais alarmes ao experimento.

Em seguida, escolha Set notifications for the alarm (Definir notificações para o alarme) e selecione ou crie um tópico do Amazon Simple Notification Service para o qual enviar notificações de alarme. Para obter mais informações, consulte Configurar notificações do Amazon SNS.

10. (Opcional) Para adicionar tags a esse experimento, escolha Tags (Etiquetas), Add new tag (Adicionar nova etiqueta).

    Em seguida, em Key (Chave), insira um nome para a etiqueta. É possível adicionar um valor opcional para a tag em Valor.

    Para adicionar outra etiqueta, escolha novamente Add new tag (Adicionar nova etiqueta).

    Para obter mais informações, consulte Etiquetar recursos da AWS.

11. Selecione Create experiment (Criar experimento).

12. Caso ainda não tenha feito, crie as variantes de recursos na aplicação.
13. Selecione Done (Concluído). O experimento não começará até você iniciá-lo.

Depois de concluir as etapas no procedimento a seguir, o experimento começa imediatamente.

Para iniciar um experimento que você criou

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha o nome do projeto.
4. Escolha a guia Experiments (Experimentos).
5. Escolha o botão ao lado do nome do experimento e escolha Actions (Ações), Start experiment.
6. (Opcional) Para exibir ou modificar as configurações definidas quando você criou o experimento, escolha Experiment setup (Configuração do experimento).
7. Escolha um horário de finalização do experimento.
8. Escolha Start experiment (Iniciar experimento).

   O experimento iniciará imediatamente.

Gerenciar recursos, lançamentos e experimentos

Use os procedimentos nessas seções para gerenciar os recursos, lançamentos e experimentos que você criou.

Tópicos

- Visualizar as regras de avaliação atuais e o tráfego de público para um recurso
- Modificar tráfego de lançamento
- Modificar as etapas futuras de um lançamento
- Modificar o tráfego do experimento
- Interromper um lançamento
- Interromper um experimento

Visualizar as regras de avaliação atuais e o tráfego de público para um recurso

Você pode usar o console do CloudWatch Evidently para visualizar como as regras de avaliação do recurso estão alocando o tráfego de público entre os lançamentos, experimentos e variações atuais do recurso.

Para visualizar o tráfego de audiência de um recurso

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha o nome do projeto que contém o recurso.
4. Escolha a guia Features (Recursos).
5. Escolha o nome do recurso.

   Em Evaluation rules (Regras de avaliação), você pode ver o fluxo de tráfego de audiência para seu recurso da seguinte forma:

   - Primeiramente, as substituições são avaliadas. Essas avaliações especificam que determinados usuários sempre recebam uma variação específica. As sessões dos usuários a quem são atribuídas substituições não contribuem para as métricas de início ou experimento.
   - Em seguida, o tráfego restante estará disponível para o lançamento em andamento, se houver um. Se houver um lançamento em andamento, a tabela na seção Launches (Lançamentos) exibe o nome do lançamento e o tráfego do lançamento dividido entre as variações de recursos. No lado direito da seção Launches (Lançamento), um indicador de Traffic exibe o quanto da audiência disponível (após as substituições) é alocada para este lançamento. O restante do tráfego não alocado para o lançamento flui para o experimento (se houver) e, em seguida, para a variação padrão.
   - Em seguida, o tráfego restante estará disponível para o experimento em andamento, se houver um. Se houver um experimento em andamento, a tabela no Experiments (Experimentos) exibe seu nome e progresso. No lado direito da seção Experiments (Experimentos), um indicador de Traffic (Tráfego) exibe quanto da audiência disponível (após substituições e lançamentos) é alocado para este experimento. O restante do tráfego não alocado para o lançamento ou o experimento recebe a variação padrão do recurso.

Modificar tráfego de lançamento

Você pode modificar a alocação do tráfego para um lançamento a qualquer momento, inclusive enquanto o lançamento estiver em andamento.

Se você tiver um lançamento e um experimento em andamento para o mesmo recurso, qualquer alteração no tráfego do recurso causará uma alteração no tráfego do experimento. Isso ocorre porque o público disponível para o experimento é a porção do público total que ainda não está alocada para o lançamento. O aumento do tráfego de lançamento irá diminuir o público disponível para o experimento. Por sua vez, a diminuição do tráfego ou o encerramento do lançamento aumentará o público disponível para o experimento.

Para modificar a alocação de tráfego para um lançamento

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha o nome do projeto que contém o lançamento.
4. Escolha a guia Launches (Lançamentos).
5. Escolha o nome do lançamento.

   Escolha Modify launch traffic (Modificar tráfego de lançamento).
6. Em Serve (Distribuir), selecione a nova porcentagem de tráfego a ser atribuída a cada variação. Você também pode optar por excluir a distribuição de variações aos usuários. À medida que você altera esses valores, é possível ver os efeitos atualizados no tráfego geral do recurso emTraffic summary (Resumo do tráfego).

   O Traffic summary (Resumo do tráfego) mostra quanto de seu tráfego geral está disponível para este lançamento e quanto desse tráfego disponível está alocado para ele.
7. Escolha Modificar.

Modificar as etapas futuras de um lançamento

É possível modificar a configuração das etapas de lançamento que ainda não ocorreram, bem como adicionar mais etapas a um lançamento.

Para modificar as etapas de um lançamento

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

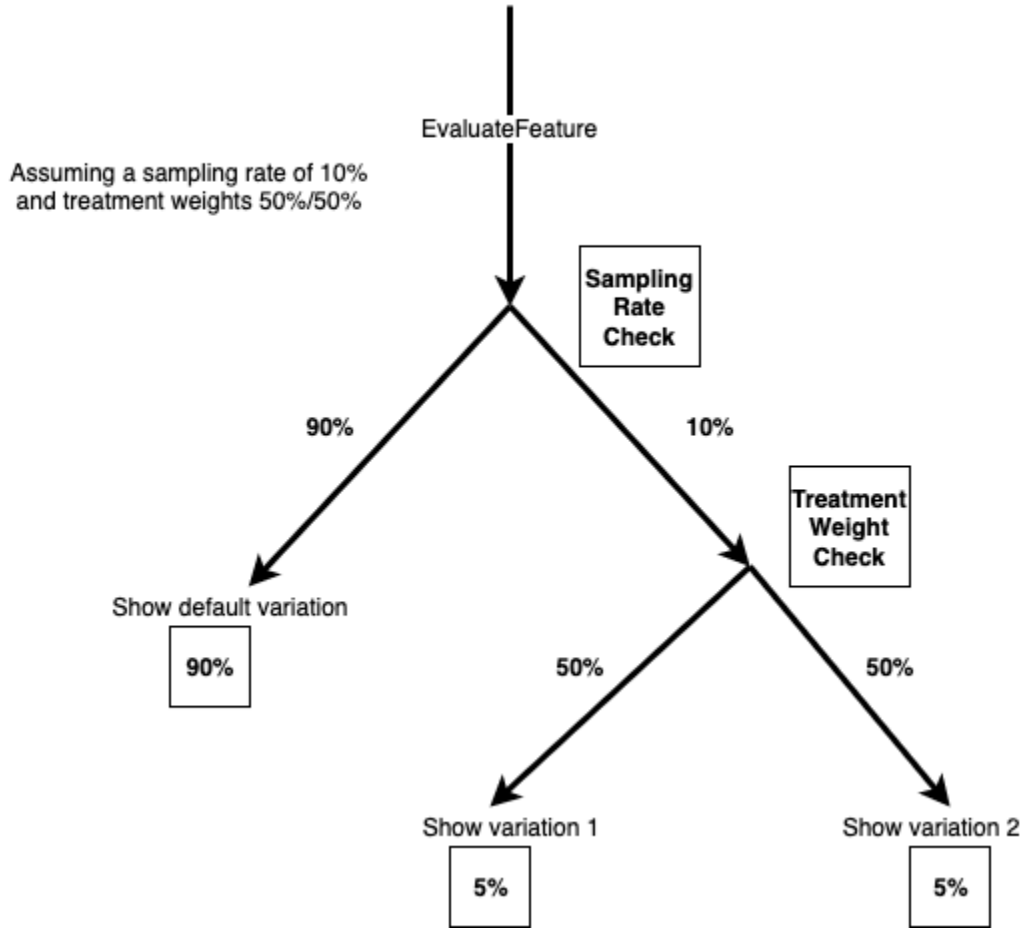
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha o nome do projeto que contém o lançamento.
4. Escolha a guia Launches (Lançamentos).
5. Escolha o nome do lançamento.

   Escolha Modify launch traffic (Modificar tráfego de lançamento).
6. Escolha Schedule launch (Agendar lançamento).
7. Em qualquer etapa que ainda não tenha sido iniciada, é possível modificar a porcentagem do público disponível que será usada no experimento. Também é possível modificar como o tráfego deles é alocado entre as variações.

   Você pode adicionar mais etapas ao lançamento escolhendo Add another step (Adicionar outra etapa). Um lançamento pode ter no máximo cinco etapas.
8. Escolha Modificar.

Modificar o tráfego do experimento

É possível modificar a taxa de amostragem para um experimento a qualquer momento, inclusive enquanto ele estiver em andamento. Entretanto, não é possível atualizar os pesos do tratamento após a execução de um experimento. Portanto, você pode alterar o tráfego total exposto ao experimento após a execução de um experimento, mas não a alocação relativa de cada tratamento. Se você modificar o tráfego de um experimento em andamento, recomendamos que aumente apenas a alocação de tráfego, para evitar a apresentação de vieses.



Para modificar a alocação de tráfego para um experimento

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application monitoring (Monitoramento de aplicações), Evidently.
3. Escolha o nome do projeto que contém o lançamento.
4. Escolha a guia Experiments (Experimentos).
5. Escolha o nome do lançamento.
6. Escolha Modify experiment traffic (Modificar o tráfego do experimento).
7. Insira uma porcentagem ou use o controle deslizante para especificar o quanto do tráfego disponível alocar para esse experimento. O tráfego disponível consiste no público total menos o tráfego alocado para um lançamento atual, se houver algum. O tráfego que não é alocado para o lançamento ou experimento é distribuído na variação padrão.
8. Escolha Modificar.

Interromper um lançamento

Se você interromper um lançamento em andamento, não poderá retomá-lo ou reiniciá-lo. Além disso, ele não será avaliado como regra para alocação de tráfego, e o tráfego alocado para o lançamento ficará disponível para o experimento do recurso, se houver um. Caso contrário, todo o tráfego será distribuído na variação padrão após o lançamento ser interrompido.

Para interromper permanentemente um lançamento

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha o nome do projeto que contém o lançamento.
4. Escolha a guia Launch (Lançamento).
5. Escolha o botão à esquerda do nome do lançamento.
6. Escolha Actions (Ações), Cancel launch (Cancelar o lançamento) ouActions (Ações), Mark as complete (Marcar como concluído).

Interromper um experimento

Se você interromper um experimento em andamento, não poderá retomá-lo ou reiniciá-lo. A parte do tráfego que foi usada anteriormente no experimento será distribuída na variação padrão.

Quando um experimento não é interrompido manualmente e passa da sua data de finalização, o tráfego não muda. A parte do tráfego alocada para o experimento ainda vai para o experimento. Para interromper isso e fazer com que o tráfego do experimento receba a variação padrão, marque o experimento como concluído.

Quando você interrompe um experimento, você pode escolher cancelá-lo ou marcá-lo como completo. Se você cancelar, ele será mostrado como Cancelled (Cancelado) na lista de experimentos. Se você optar por marcá-lo como completo, ele será mostrado como Completed (Concluído).

Para interromper um experimento permanentemente

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha o nome do projeto que contém o experimento.

4. Escolha a guia Experiments (Experimentos).
5. Escolha o botão à esquerda do nome do experimento.
6. Escolha Actions (Ações), Cancel experiment (Cancelar o experimento) ou Actions (Ações), Mark as complete (Marcar como concluído).

Adicionar código à sua aplicação

Para trabalhar com o CloudWatch Evidently, você adiciona código à aplicação para atribuir uma variação a cada sessão do usuário e enviar métricas para o Evidently. Você utiliza a operação `EvaluateFeature` do CloudWatch Evidently para atribuir variações às sessões dos usuários e você utiliza a operação `PutProjectEvents` para enviar eventos ao Evidently a serem utilizados a fim de calcular métricas para seus lançamentos ou experimentos.

Ao criar variações ou métricas personalizadas, o console do CloudWatch Evidently fornece exemplos do código que você precisa adicionar.

Para um exemplo completo, consulte Tutorial: teste de A/B com a aplicação de exemplo do Evidently.

Utilizar a operação EvaluateFeature

Quando as variações de recursos são usadas em um lançamento ou experimento, a aplicação usa a operação EvaluateFeature para atribuir uma variação para cada sessão do usuário. A atribuição de uma variação a um usuário é um evaluation event (evento de avaliação). Quando você chama essa operação, você passa pelo seguinte processo:

- Feature name (Nome do recurso): obrigatório. O Evidently processa a avaliação de acordo com as regras de avaliação de recursos do lançamento ou experimento e seleciona uma variação para a entidade.
- entityId– Obrigatório. Representa um único usuário.
- EvaluationContext– Opcional. Um objeto JSON representando informações adicionais sobre um usuário. Se você tiver criado segmentos, o Evidently usará esse valor para corresponder o usuário a um segmento do seu público durante as avaliações de recursos. Para ter mais informações, consulte Usar segmentos para delimitar o público.

  Veja a seguir um exemplo de um valor de `evaluationContext` que você pode enviar para o Evidently.

```
{
```

```
    "Browser": "Chrome",
    "Location": {
        "Country": "United States",
        "Zipcode": 98007
    }
}
```

Avaliações com aderência

O CloudWatch Evidently utiliza avaliações com “aderência”. Uma única configuração de `entityId`, recurso, configuração de recurso e `evaluationContext` sempre obtém a mesma atribuição de variação. A única vez em que essa atribuição de variação muda é quando uma entidade é adicionada a uma substituição ou o tráfego do experimento é discado.

Uma configuração de recurso inclui o seguinte:

- As variações de recursos
- A configuração de variação (porcentagens atribuídas a cada variação) para um experimento em execução para esse recurso, se houver.
- A configuração de variação para uma inicialização em execução para esse recurso, se houver. A configuração de variação inclui as substituições de segmento definidas, se houver.

Se a alocação de tráfego de um experimento for aumentada, qualquer `entityId` que tenha sido previamente atribuído a um grupo de tratamento experimental continuará recebendo o mesmo tratamento. Qualquer `entityId` que tenha sido previamente atribuído ao grupo de controle pode ser atribuído a um grupo de tratamento experimental de acordo com a configuração de variação especificada para o experimento.

Se a alocação de tráfego de um experimento for reduzida, um `entityId` pode passar de um grupo de tratamento para um grupo de controle, mas não entrará em um grupo de tratamento diferente.

Utilizar a operação PutProjectEvents

Para codificar uma métrica personalizada do Evidently, use a operação PutProjectEvents. Veja a seguir um exemplo de carga útil simples.

```
{
    "events": [
```

```
        {
            "timestamp": {{$timestamp}},
            "type": "aws.evidently.custom",
            "data": "{\"details\": {\"pageLoadTime\": 800.0}, \"userDetails\":
 {\"userId\": \"test-user\"}}"
        }
    ]
}
```

A `entityIdKey` pode ser apenas uma `entityId`, ou você pode renomeá-la para qualquer outra coisa, como`userId`. No evento real, `entityId` pode ser um nome de usuário, um ID de sessão e assim por diante.

```
"metricDefinition":{
    "name": "noFilter",
    "entityIdKey": "userDetails.userId", //should be consistent with jsonValue in
 events "data" fields
    "valueKey": "details.pageLoadTime"
},
```

Para ter certeza de que os eventos estejam associados ao lançamento ou experimento correto, transmita o mesmo `entityId` ao chamar `EvaluateFeature` e `PutProjectEvents`. Certifique-se de chamar `PutProjectEvents` depois de `EvaluateFeature`. Caso contrário, os dados serão descartados e não serão usados pelo CloudWatch Evidently.

A operação `PutProjectEvents` não requer o nome do recurso como parâmetro de entrada. Portanto, você pode usar um evento único em vários experimentos. Por exemplo, suponhamos que você chame `EvaluateFeature` com `entityId` definido como `userDetails.userId`. Se houver dois ou mais experimentos em execução, será possível fazer com que um único evento da sessão desse usuário emita métricas para cada um desses experimentos. Para fazer isso, chame `PutProjectEvents` uma vez para cada experimento, usando o mesmo `entityId`.

Prazo

Depois que a aplicação chama `EvaluateFeature`, há um período de tempo de uma hora no qual os eventos de métricas de `PutProjectEvents` são atribuídos com base nessa avaliação. Se mais eventos ocorrerem após esse período de uma hora, não serão atribuídos.

Porém, se o mesmo `entityId` for usado para uma nova chamada de `EvaluateFeature` durante a janela de uma hora da chamada inicial, o último resultado de `EvaluateFeature` será usado no

lugar, e o timer de uma hora será zerado. Isso pode acontecer apenas em certas circunstâncias, como quando o tráfego do experimento é chamado entre as duas atribuições, conforme explicado na seção anterior, Avaliações com aderência.

Para um exemplo completo, consulte Tutorial: teste de A/B com a aplicação de exemplo do Evidently.

Armazenamento de dados do projeto

O Evidently coleta dois tipos de eventos:

- Eventos de avaliação estão relacionados a qual variação de recurso é atribuída a uma sessão de usuário. O Evidently usa esses eventos para produzir métricas e outros dados de experimento e execução, que você pode visualizar no console do Evidently.

  Você também pode escolher armazenar esses eventos de avaliação no Amazon CloudWatch Logs ou no Amazon S3.
- Eventos personalizados são usados para gerar métricas a partir de ações do usuário, como cliques e checkouts. O Evidently, não fornece um método para você armazenar eventos personalizados. Se você quiser salvá-los, deve modificar o código da aplicação para enviá-los para uma opção de armazenamento fora do Evidently.

Formato de logs de eventos de avaliação

Se você optar por armazenar eventos de avaliação nos logs do CloudWatch ou do Amazon S3, cada evento de avaliação será armazenado como um evento de log com o seguinte formato:

```
{
    "event_timestamp": 1642624900215,
    "event_type": "evaluation",
    "version": "1.0.0",
    "project_arn": "arn:aws:evidently:us-east-1:123456789012:project/petfood",
    "feature": "petfood-upsell-text",
    "variation": "Variation1",
    "entity_id": "7",
    "entity_attributes": {},
    "evaluation_type": "EXPERIMENT_RULE_MATCH",
    "treatment": "Variation1",
    "experiment": "petfood-experiment-2"
}
```

Aqui estão mais detalhes sobre o formato de evento de avaliação anterior:

- O timestamp está no horário do UNIX com milissegundos
- A variação é o nome da variação do recurso que foi atribuído a essa sessão do usuário.
- O ID da entidade é uma string.
- Os atributos da entidade são um hash de valores arbitrários enviados pelo cliente. Por exemplo, se o `entityId` for mapeado para azul ou verde, você poderá, opcionalmente, enviar UserIDs, dados de sessão ou o que mais desejar de uma perspectiva de correlação e data warehouse.

Política do IAM e criptografia para armazenamento de eventos de avaliação no Amazon S3

Se optar por usar o Amazon S3 para armazenar eventos de avaliação, você deverá adicionar uma política do IAM como a que se segue para permitir que o Evidently publique logs no bucket do Amazon S3. Isso ocorre porque os buckets do Amazon S3 e os objetos que eles contêm são privados e não permitem acesso a outros serviços por padrão.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/
*",
            "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-
control"}}
        },
        {
            "Sid": "AWSLogDeliveryCheck",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": ["s3:GetBucketAcl", "s3:ListBucket"],
            "Resource": "arn:aws:s3:::bucket_name"
        }
    ]
}
```

Se você armazenar dados do Evidently no Amazon S3, também poderá optar por criptografá-los com criptografia do lado do servidor com Chaves AWS Key Management Service(SSE-KMS). Para obter mais informações, consulte Proteger dados usando a criptografia no lado do servidor.

Se você usar uma chave gerenciada pelo cliente do AWS KMS, você deve adicionar o seguinte à política do IAM para sua chave. Isso permite que Evidently grave no bucket.

```
{
    "Sid": "AllowEvidentlyToUseCustomerManagedKey",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "delivery.logs.amazonaws.com"
        ]
    },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
       "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Como o Evidently calcula resultados

Você pode usar o teste A/B do Amazon CloudWatch Evidently como ferramenta para a tomada de decisão baseada em dados. Em um teste A/B, os usuários são atribuídos aleatoriamente ao grupo de controle (também chamado de variação padrão) ou a um dos grupos de tratamento (também chamados de variações testadas). Por exemplo, os usuários do grupo de controle podem experimentar o site, o serviço ou a aplicação como faziam antes do início do experimento. Enquanto isso, os usuários do grupo de tratamento podem experimentar a alteração.

O CloudWatch Evidently oferece suporte para até cinco variações diferentes em um experimento. O Evidently atribui tráfego aleatoriamente para essas variações. Dessa forma, é possível acompanhar métricas de negócios (como receita) e métricas de performance (como latência) de cada grupo. O Evidently realiza estas ações:

- Compara o tratamento com o controle. (Por exemplo, comparar se a receita aumenta ou diminui com um novo processo de pagamento.)

- Indica se a diferença observada entre o tratamento e o controle é significativa. Para isso, o Evidently oferece dois métodos: Frequentist significance levels (Níveis de significância frequentista) e Bayesian probabilities (Probabilidades bayesianas).

Por que usar métodos frequentistas e bayesianos?

Considere um caso em que o tratamento não tem efeito comparado ao controle ou um caso em que o tratamento é idêntico ao controle (um teste A/A). Você ainda perceberia uma pequena diferença entre o tratamento e o controle nos dados. Isso ocorre porque os participantes do teste consistem em uma amostra finita de usuários, representando uma pequena porcentagem de todos os usuários do site, serviço ou aplicação. Os níveis de significância frequentista e as probabilidades bayesianas informam se a diferença observada é significativa ou se ocorreu devido ao acaso.

O Evidently considera o seguinte para determinar se a diferença observada é significativa:

- Se a diferença é grande
- Quantas amostras compõem o teste
- Como os dados estão distribuídos

Análise frequentista no Evidently

O Evidently usa testes sequenciais, que evitam os problemas comuns de olhada rápida, uma armadilha comum das estatísticas frequentistas. A olhada rápida é a prática de verificar os resultados de um teste A/B em andamento para interrompê-lo e tomar uma decisão com base nos resultados observados. Para obter mais informações sobre testes sequenciais, consulte Time-uniform, nonparametric, nonasymptotic confidence sequences (Sequências de confiança uniformes, não paramétricas e não assintóticas) por Howard et al. (Ann. Statist. 49 (2) 1055 - 1080, 2021).

Como os resultados do Evidently são válidos a qualquer momento (resultados válidos a qualquer momento), você pode dar uma olhada rápida nos resultados durante o experimento e ainda tirar conclusões sólidas. Isso pode reduzir alguns dos custos da experimentação, pois é possível interromper um experimento antes do horário programado se os resultados já forem significativos.

O Evidently gera níveis de significância válidos a qualquer momento e intervalos de confiança de 95% válidos a qualquer momento da diferença entre a variação testada e a variação padrão na métrica de destino. A coluna Result (Resultado) nos resultados do experimento indica a performance de variação testada, que pode ser:

- Inconclusive (Inconclusiva): o nível de significância é inferior a 95%
- Better (Melhor): o nível de significância é de 95% ou mais e uma das seguintes situações é verdadeira:
  - O limite inferior do intervalo de confiança de 95% é maior que zero, e a métrica deve aumentar
  - O limite superior do intervalo de confiança de 95% é menor que zero, e a métrica deve diminuir
- Worse (Pior): o nível de significância é de 95% ou mais e uma das seguintes situações é verdadeira:
  - O limite superior do intervalo de confiança de 95% é maior que zero, e a métrica deve aumentar
  - O limite inferior do intervalo de confiança de 95% é menor que zero, e a métrica deve diminuir
- Best (A melhor): o experimento tem duas ou mais variações testadas, além da variação padrão, e as seguintes condições são atendidas:
  - A variação se qualifica para a designação Better (A melhor)
  - Um dos valores a seguir é verdadeiro:
    - O limite inferior do intervalo de confiança de 95% é maior do que o limite superior dos intervalos de confiança de 95% de todas as outras variações, e a métrica deve aumentar
    - O limite superior do intervalo de confiança de 95% é menor do que o limite inferior dos intervalos de confiança de 95% de todas as outras variações, e a métrica deve diminuir

Análise bayesiana no Evidently

Com a análise bayesiana, é possível calcular a probabilidade de que a média na variação testada seja maior ou menor do que a média na variação padrão. O Evidently realiza inferência bayesiana para a média da métrica de destino usando prioris conjugadas. Com prioris conjugadas, o Evidently pode inferir com mais eficiência a distribuição posterior necessária para a análise bayesiana.

O Evidently espera até a data final do experimento para computar os resultados da análise bayesiana. A página de resultados exibe o seguinte:

- probabilidade de aumento: a probabilidade de que a média da métrica na variação testada seja pelo menos 3% maior do que a média na variação padrão
- probabilidade de diminuição: a probabilidade de que a média da métrica na variação testada seja pelo menos 3% menor do que a média na variação padrão
- probabilidade de nenhuma alteração: a probabilidade de que a média da métrica na variação testada esteja entre mais ou menos 3% da média na variação padrão

A coluna Result (Resultado) indica a performance de variação e pode ser:

- Better (Melhor): a probabilidade de aumento é de pelo menos 90%, e a métrica deve aumentar, ou a probabilidade de diminuição é de pelo menos 90%, e a métrica deve diminuir
- Worse (Pior): a probabilidade de diminuição é de pelo menos 90%, e a métrica deve aumentar, ou a probabilidade de diminuição é de pelo menos 90%, e a métrica deve diminuir

Visualizar os resultados do lançamento no painel

É possível consultar o progresso e os resultados da métrica de um experimento enquanto ele estiver em andamento e depois de concluído.

Para ver o progresso e os resultados de um lançamento

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha o nome do projeto que contém o lançamento.
4. Escolha a guia Launch (Lançamento).
5. Escolha o nome do lançamento.
6. Para consultar as etapas de lançamento e as alocações de tráfego para cada etapa, escolha a guia Launch (Lançamento).
7. Para ver o número de sessões de usuário atribuídas a cada variação ao longo do tempo e para visualizar as métricas de performance para cada variação no lançamento, escolha a guiaMonitoring (Monitoramento).

   Essa visualização também exibe se algum alarme de lançamento entrou no estado `ALARM` durante o lançamento.
8. Para visualizar as variações, métricas, alarmes e etiquetas para este lançamento, escolha a guia Configuration (Configuração).

Visualizar resultados do experimento no painel

É possível ver os resultados estatísticos de um experimento enquanto ele estiver em andamento e depois que for concluído. Os resultados do experimento estarão disponíveis até 63 dias após o início do experimento. Eles não estarão disponíveis depois disso devido às políticas de retenção de dados do CloudWatch.

Nenhum resultado estatístico é exibido até que cada variação tenha pelo menos 100 eventos.

O Evidently realiza uma análise adicional offline do valor-p ao final do experimento. A análise offline do valor-p pode detectar significância estatística em alguns casos em que os valores-p de validade a qualquer momento usados durante o experimento não encontram significância estatística.

Para obter mais informações sobre como o CloudWatch Evidently calcula os resultados dos experimentos, consulte Como o Evidently calcula resultados.

Para consultar os resultados de um experimento

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Escolha o nome do projeto que contém o experimento.
4. Escolha a guia Experiments (Experimentos).
5. Escolha o nome do experimento e, em seguida, escolha a guia Results (Resultados).
6. Em Variation performance (Performance da variação), há um controle onde você pode selecionar quais estatísticas do experimento exibir. Se você selecionar mais de uma estatística, o Evidently exibirá um gráfico e uma tabela para cada estatística.

   Cada gráfico e tabela exibe os resultados do experimento até o momento.

   Cada gráfico pode exibir os resultados a seguir. Você pode usar o controle à direita do gráfico para determinar qual dos seguintes itens será exibido:

   - O número de eventos de sessão do usuário registrados para cada variação.
   - O valor médio da métrica que foi selecionada na parte superior do gráfico, para cada variação.
   - A significância estatística dos experimentos. Isso compara a diferença da métrica selecionada na parte superior do gráfico com a variação padrão e cada uma das outras variações.
   - Os limites de confiança 95% superiores e inferiores sobre a diferença da métrica selecionada, entre cada uma das variações e a variação padrão.

   A tabela exibe uma linha para cada variação. O Evidently mostra se cada variação que não é padrão recebeu dados suficientes para que se declare os resultados estatisticamente significativos. Também mostra se a o aperfeiçoamento da variação no valor estatístico atingiu um nível de confiança de 95%.

Por fim, na coluna Result (Resultado), o Evidently fornece uma recomendação sobre qual variação tem melhor performance com base nessa estatística, ou se os resultados são inconclusivos.

Como o CloudWatch Evidently coleta e armazena dados

O Amazon CloudWatch Evidently coleta e armazena dados relacionados às configurações do projeto para que os clientes possam executar experimentos e lançamentos. Os dados incluem o seguinte:

- Metadados sobre projetos, recursos, lançamentos e experimentos
- Eventos de métrica
- Dados de avaliação

Os metadados de recursos são armazenados no Amazon DynamoDB. Por padrão, os dados são criptografados em repouso, usando as Chaves pertencentes à AWS. Essas chaves são uma coleção de chaves do AWS KMS de propriedade e gerenciamento da AWS service (Serviço da AWS) para uso em várias Contas da AWS. Os clientes não podem visualizar, gerenciar ou auditar o uso dessas chaves. Os clientes também não precisam executar medidas nem alterar programas para proteger as chaves que criptografam seus dados.

Para obter mais informações, consulte Chaves pertencentes à AWS no Guia do desenvolvedor do AWS Key Management Service.

Os eventos de métrica e de avaliação do Evidently são entregues diretamente em locais de propriedade do cliente.

Os dados em trânsito são criptografados automaticamente com HTTPS. Eles serão entregues para locais de propriedade do cliente.

Você também pode escolher armazenar eventos de avaliação no Amazon Simple Storage Service ou no Amazon CloudWatch Logs. Para obter mais informações sobre como proteger seus dados nesses serviços, consulte Habilitar a criptografia de bucket padrão do Amazon S3 e Criptografar dados de log no CloudWatch Logs usando AWS KMS.

Recuperação de dados

É possível recuperar seus dados usando as APIs do CloudWatch Evidently. Para recuperar dados do projeto, useGetProject ou ListProjects.

Para recuperar dados do recurso, use GetFeature ou ListFeatures.

Para recuperar dados do lançamento, use GetLaunch ou ListLaunches.

Para recuperar dados do experimento, use GetExperiment, ListExperiments ouGetExperimentResults.

Modificar e excluir dados

É possível modificar e excluir seus dados usando as APIs do CloudWatch Evidently. Para dados do projeto, useUpdateProject ou DeleteProject.

Para dados de recursos, use UpdateFeature ouDeleteFeature.

Para dados de lançamento, use UpdateLaunch ou DeleteLaunch.

Para dados do experimento, use UpdateExperiment ou DeleteExperiment.

Usar perfis vinculados ao serviço do Evidently

O CloudWatch Evidently usa perfis vinculados ao serviço do AWS Identity and Access Management (IAM). O perfil vinculado ao serviço é um tipo exclusivo de perfil do IAM vinculado diretamente ao Evidently. Os perfis vinculados ao serviço são predefinidos pelo Evidently e incluem todas as permissões que o serviço exige para chamar outros produtos da AWS em seu nome.

Um perfil vinculado ao serviço facilita a configuração do Evidently porque você não precisa adicionar as permissões necessárias manualmente. O Evidently define as permissões de perfis vinculados ao serviço e, exceto se definido de outra forma, somente o Evidently pode assumir os perfis. As permissões definidas incluem as políticas de confiança e de permissões, e essa política de permissões não pode ser anexada a nenhuma outra entidade do IAM.

Você pode excluir uma função vinculada ao serviço somente depois de primeiro excluir seus recursos relacionados. Isso protege seus recursos do Evidently, pois não é possível remover por engano as permissões para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com perfis vinculados aos serviços, consulte Serviços da AWS que funcionam com o IAM e procure os serviços que apresentam Sim na coluna Funções vinculadas aos serviços. Escolha um Sim com um link para visualizar a documentação da função vinculada a esse serviço.

Permissões de perfil vinculado ao serviço para o Evidently

O Evidently usa o perfil vinculado ao serviço chamado AWSServiceRoleForCloudWatchEvidently: permite que o CloudWatch Evidently gerencie recursos da AWS associados em nome do cliente.

O perfil vinculado ao serviço AWSServiceRoleForCloudWatchEvidently confia nos seguintes serviços para assumir o perfil:

- `CloudWatch Evidently`

A política de permissões do perfil chamada AmazonCloudWatchEvidentlyServiceRolePolicy permite que o Evidently conclua as seguintes ações nos recursos especificados:

- Ações: `appconfig:StartDeployment`, `appconfig:StopDeployment`, `appconfig:ListDeployments` e `appconfig:TagResource` em thick clients do Evidently.

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada ao serviço. Para obter mais informações, consulte Service-linked role permissions (Permissões de função vinculada a serviços) no Guia do usuário do IAM.

Criar um perfil vinculado ao serviço do Evidently

Não é necessário criar manualmente uma função vinculada ao serviço. Quando você começa a usar um thick client do Evidently no AWS Management Console, na AWS CLI ou na API da AWS, o Evidently cria o perfil vinculado ao serviço para você.

Se excluir essa função vinculada ao serviço e precisar criá-la novamente, você poderá usar esse mesmo processo para recriar a função em sua conta. Quando você começa a usar um thick client do Evidently, o Evidently cria o perfil vinculado ao serviço para você.

Editar um perfil vinculado ao serviço para o Evidently

O Evidently não permite editar o perfil vinculado ao serviço AWSServiceRoleForCloudWatchEvidently. Depois que criar uma função vinculada ao serviço, você não poderá alterar o nome da função, pois várias entidades podem fazer referência a ela. No entanto, será possível editar a descrição da função usando o IAM. Para ter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Excluir um perfil vinculado ao serviço do Evidently

Se você não precisar mais usar um recurso ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de excluí-la manualmente. É necessário excluir todos os projetos do Evidently que estão usando thick clients.

> **Note**
>
> Se o serviço do Evidently estiver usando o perfil quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir os recursos do Evidently usados por AWSServiceRoleForCloudWatchEvidently

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application monitoring (Monitoramento de aplicações), Evidently.
3. Na lista de projetos, marque a caixa de seleção ao lado dos projetos que usaram thick clients.
4. Escolha Project actions (Ações do projeto), Delete project (Excluir projeto).

Como excluir manualmente a função vinculada ao serviço usando o IAM

Use o console do IAM, a AWS CLI ou a API da AWS para excluir o perfil vinculado ao serviço AWSServiceRoleForCloudWatchEvidently service-linked. Para ter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Regiões compatíveis com perfis vinculados ao serviço do Evidently

O Evidently oferece suporte a perfis vinculados ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte Regiões e endpoints da AWS.

Cotas do CloudWatch Evidently

O CloudWatch Evidently tem as cotas a seguir.

Recurso	Cota padrão
Projetos	50 por região por conta É possível solicitar um aumento da cota.
Segmentos	500 por região por conta É possível solicitar um aumento da cota.
Cotas por projeto	• 100 recursos no total • 500 lançamentos no total • 50 lançamentos em execução • 500 experimentos no total • 50 experimentos em execução É possível solicitar um aumento de cota para todas essas cotas.
Cotas de API (todas as cotas são por região)	• PutProjectEvents: 1.000 transações por segundo (TPS) nas regiões Leste dos EUA (N. da Virgínia), Oeste dos EUA (Oregon) e Europa (Irlanda). 200 TPS em todas as outras regiões. • EvaluateFeature: 1.000 TPS nas regiões Leste dos EUA (N. da Virgínia), Oeste dos EUA (Oregon) e Europa (Irlanda). 200 TPS em todas as outras regiões. • BatchEvaluateFeature: 50 TPS • APIs Create, Read, Update, Delete (CRUD): 10 TPS combinados em todas as APIs CRUD É possível solicitar um aumento de cota para todas essas cotas.

Tutorial: teste de A/B com a aplicação de exemplo do Evidently

Esta seção fornece um tutorial para usar o Amazon CloudWatch Evidently para testes A/B. Este tutorial é a aplicação de exemplo do Evidently, uma aplicação de reação simples. O aplicativo de exemplo será configurado para exibir um recurso `showDiscount` ou não. Quando o recurso é exibido para um usuário, o preço exibido no site de compras é exibido com um desconto de 20%.

Além de mostrar o desconto para alguns usuários e não para outros, neste tutorial você configurou o Evidently para coletar métricas de tempo de carregamento de página de ambas as variações.

> **Warning**
>
> Este cenário precisa de usuários do IAM com acesso programático e credenciais de longo prazo, o que representa um risco de segurança. Para ajudar a reduzir esse risco, recomendamos que você forneça a esses usuários somente as permissões necessárias para realizar a tarefa e que você os remova quando não forem mais necessários. As chaves de acesso podem ser atualizadas, se necessário. Para obter mais informações, consulte Atualização de chaves de acesso no Guia de usuário do IAM.

Etapa 1: baixar a aplicação de exemplo

Comece baixando a aplicação de exemplo do Evidently.

Para baixar a aplicação de exemplo

1. Baixe a aplicação de exemplo do seguinte bucket do Amazon S3:

   ```
   https://evidently-sample-application.s3.us-west-2.amazonaws.com/evidently-sample-shopping-app.zip
   ```

2. Descompacte o pacote.

Etapa 2: adicionar o endpoint do Evidently e configurar credenciais

Em seguida, adicione a região e o endpoint para o Evidently ao arquivo `config.js` no diretório `src` no pacote da aplicação de exemplo, conforme mostrado a seguir:

```
evidently: {
```

```
    REGION: "us-west-2",
    ENDPOINT: "https://evidently.us-west-2.amazonaws.com (https://evidently.us-
west-2.amazonaws.com/)",
},
```

Você também deve garantir que a aplicação tenha permissão para chamar o CloudWatch Evidently.

Para conceder à aplicação de exemplo permissões para chamar o Evidently

1. Federe-a à sua conta da AWS.
2. Crie um usuário do IAM e anexe a política AmazonCloudWatchEvidentlyFullAccess a esse usuário.
3. Anote o ID da chave de acesso do usuário do IAM e a chave de acesso secreta, pois você precisará deles na próxima etapa.
4. No mesmo arquivo `config.js` modificado anteriormente nesta seção, insira os valores do ID da chave de acesso e da chave de acesso secreta, como no exemplo a seguir:

```
credential: {
   accessKeyId: "Access key ID",
   secretAccessKey: "Secret key"
}
```

> **Important**
>
> Usamos esta etapa para tornar a aplicação de exemplo a mais simples possível para você experimentar. Não recomendamos colocar a credencial de usuário do IAM em sua aplicação de produção real. Em vez disso, recomendamos usar o Amazon Cognito para autenticação. Para obter mais informações, consulte Como integrar o Amazon Cognito a aplicações Web e móveis.

Etapa 3: configurar código para a avaliação do recurso

Quando você usa o CloudWatch Evidently para avaliar um recurso, você deve usar a operação EvaluateFeature para selecionar aleatoriamente uma variação de recursos para cada sessão do usuário. Esta operação atribui sessões do usuário a cada variação do recurso, de acordo com as porcentagens especificadas no experimento.

Para configurar o código de avaliação de recursos para a aplicação de demonstração da livraria

1. Adicione o construtor do cliente ao arquivo `src/App.jsx` para que a aplicação de exemplo possa chamar o Evidently.

```
import Evidently from 'aws-sdk/clients/evidently';
import config from './config';

const defaultClientBuilder = (
    endpoint,
    region,
) => {
    const credentials = {
        accessKeyId: config.credential.accessKeyId,
        secretAccessKey: config.credential.secretAccessKey
    }
    return new Evidently({
        endpoint,
        region,
        credentials,
    });
};
```

2. Adicione o seguinte à seção do código `const App` para iniciar o cliente.

```
  if (client == null) {
      client = defaultClientBuilder(
        config.evidently.ENDPOINT,
        config.evidently.REGION,
      );
```

3. Construa `evaluateFeatureRequest` adicionando o código a seguir. Esse código preenche previamente o nome do projeto e o nome do recurso que recomendaremos mais adiante neste tutorial. Você poderá substituí-los por seus próprios nomes de projeto e recurso, desde que também especifique esses nomes de projeto e recurso no console do Evidently.

```
const evaluateFeatureRequest = {
              entityId: id,
              // Input Your feature name
              feature: 'showDiscount',
              // Input Your project name'
              project: 'EvidentlySampleApp',
```

```
            };
```

4. Adicione o código para chamar o Evidently para avaliação de recursos. Quando a solicitação é enviada, o Evidently atribui aleatoriamente a sessão do usuário para ver o recurso `showDiscount` ou não.

```
client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
    if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
    }
    getPageLoadTime()
 })
```

Etapa 4: configurar código para as métricas do experimento

Para a métrica personalizada, use a API `PutProjectEvents` do Evidently para enviar resultados de métricas para o Evidently. Os exemplos a seguir mostram como configurar a métrica personalizada e enviar dados do experimento para Evidently.

Adicione a seguinte função para calcular o tempo de carregamento da página e use `PutProjectEvents` para enviar os valores de métrica para o Evidently. Adicione a seguinte função em `Home.tsx` e chame essa função dentro da API `EvaluateFeature`:

```
const getPageLoadTime = () => {
    const timeSpent = (new Date().getTime() - startTime.getTime()) * 1.000001;
    const pageLoadTimeData = `{
      "details": {
        "pageLoadTime": ${timeSpent}
      },
      "UserDetails": { "userId": "${id}", "sessionId": "${id}"}
    }`;
    const putProjectEventsRequest = {
      project: 'EvidentlySampleApp',
      events: [
        {
          timestamp: new Date(),
          type: 'aws.evidently.custom',
          data: JSON.parse(pageLoadTimeData)
        },
      ],
    };
```

```
    client.putProjectEvents(putProjectEventsRequest).promise();
  }
```

É assim que o arquivo `App.js` deveria ser após toda a edição que você fez desde o download.

```
import React, { useEffect, useState } from "react";
import { BrowserRouter as Router, Switch } from "react-router-dom";
import AuthProvider from "contexts/auth";
import CommonProvider from "contexts/common";
import ProductsProvider from "contexts/products";
import CartProvider from "contexts/cart";
import CheckoutProvider from "contexts/checkout";
import RouteWrapper from "layouts/RouteWrapper";
import AuthLayout from "layouts/AuthLayout";
import CommonLayout from "layouts/CommonLayout";
import AuthPage from "pages/auth";
import HomePage from "pages/home";
import CheckoutPage from "pages/checkout";
import "assets/scss/style.scss";
import { Spinner } from 'react-bootstrap';

import Evidently from 'aws-sdk/clients/evidently';
import config from './config';

const defaultClientBuilder = (
  endpoint,
  region,
) => {
  const credentials = {
    accessKeyId: config.credential.accessKeyId,
    secretAccessKey: config.credential.secretAccessKey
  }
  return new Evidently({
    endpoint,
    region,
    credentials,
  });
};

const App = () => {
  const [isLoading, setIsLoading] = useState(true);
  const [startTime, setStartTime] = useState(new Date());
  const [showDiscount, setShowDiscount] = useState(false);
```

```
  let client = null;
  let id = null;

  useEffect(() => {
    id = new Date().getTime().toString();
    setStartTime(new Date());
    if (client == null) {
      client = defaultClientBuilder(
        config.evidently.ENDPOINT,
        config.evidently.REGION,
      );
    }
    const evaluateFeatureRequest = {
      entityId: id,
      // Input Your feature name
      feature: 'showDiscount',
      // Input Your project name'
      project: 'EvidentlySampleApp',
    };

    // Launch
    client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
      if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
      }
    });

    // Experiment
    client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
      if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
      }
      getPageLoadTime()
    })

    setIsLoading(false);
  },[]);

  const getPageLoadTime = () => {
    const timeSpent = (new Date().getTime() - startTime.getTime()) * 1.000001;
    const pageLoadTimeData = `{
      "details": {
        "pageLoadTime": ${timeSpent}
      },
```

```
      "UserDetails": { "userId": "${id}", "sessionId": "${id}"}
    }`;
    const putProjectEventsRequest = {
      project: 'EvidentlySampleApp',
      events: [
        {
          timestamp: new Date(),
          type: 'aws.evidently.custom',
          data: JSON.parse(pageLoadTimeData)
        },
      ],
    };
    client.putProjectEvents(putProjectEventsRequest).promise();
  }
  return (
    !isLoading? (
    <AuthProvider>
      <CommonProvider>
        <ProductsProvider>
          <CartProvider>
            <CheckoutProvider>
              <Router>
                <Switch>
                  <RouteWrapper
                    path="/"
                    exact
                    component={() => <HomePage showDiscount={showDiscount}/>}
                    layout={CommonLayout}
                  />
                  <RouteWrapper
                    path="/checkout"
                    component={CheckoutPage}
                    layout={CommonLayout}
                  />
                  <RouteWrapper
                    path="/auth"
                    component={AuthPage}
                    layout={AuthLayout}
                  />
                </Switch>
              </Router>
            </CheckoutProvider>
          </CartProvider>
        </ProductsProvider>
```

```
      </CommonProvider>
    </AuthProvider> ) : (
      <Spinner animation="border" />
    )
  );
};

export default App;
```

Cada vez que um usuário visita a aplicação de exemplo, uma métrica personalizada é enviada para análise do Evidently. O Evidently analisa cada métrica e exibe resultados em tempo real no painel do Evidently. O exemplo a seguir mostra uma carga útil de métrica:

```
[ {"timestamp": 1637368646.468, "type": "aws.evidently.custom", "data": "{\"details
\":{\"pageLoadTime\":2058.002058},\"userDetails\":{\"userId\":\"1637368644430\",
\"sessionId\":\"1637368644430\"}}" } ]
```

Etapa 5: criar o projeto, o recurso e o experimento

Em seguida, você cria o projeto, o recurso e o experimento no console do CloudWatch Evidently.

Para criar o projeto, o recurso e o experimento para este tutorial

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. EscolhaCreate project (Criar projeto) e preencha os campos. Você deve usar **`EvidentlySampleApp`** para o nome do projeto para que o exemplo funcione corretamente. Em Evaluation event storage (Armazenamento de eventos de avaliação), escolha Don't store Evaluation events (Não armazenar eventos de avaliação).

   Após preencher os campos, escolha Create Project (Criar projeto).

   Para obter mais detalhes, consulte Criar um novo projeto da .
4. Depois que o projeto for criado, crie um recurso nele. Nomeie o recurso como **`showDiscount`**. Nesse recurso, crie duas variações do tipo **`Boolean`**. Nomeie a primeira variação como **`disable`** com um valor de**`False`** e nomeie a segunda variação como **`enable`** com um valor de **`True`**.

   Para obter mais informações sobre como criar um recurso, consulte Adicionar um recurso a um projeto.

5. Depois de terminar de criar o recurso, crie um experimento no projeto. Nomeie o experimento como **`pageLoadTime`**.

   Este experimento usará uma métrica personalizada chamada `pageLoadTime`, que mede o tempo de carregamento da página que está sendo testada. Métricas personalizadas para experimentos são criadas usando o Amazon EventBridge. Para obter mais informações sobre o EventBridge, consulte O que é o Amazon EventBridge?.

   Para criar essa métrica personalizada, faça o seguinte ao criar o experimento:

   - Em Metrics (Métricas), em Metric source (Fonte da métrica), escolha Custom metrics (Métricas personalizadas).
   - Em Metric name (Nome da métrica), insira **`pageLoadTime`**.
   - Em Goal (Objetivo), escolha Decrease (Diminuir). Isso indica que um valor menor dessa métrica indicará a melhor variação do recurso.
   - Em Metric rule (Regra de métrica), insira o seguinte:
     - Em Entity ID (ID da entidade), insira **`UserDetails.userId`**.
     - Em Value key (Valor da chave), insira **`details.pageLoadTime`**.
     - Em Units (Unidade), insira **`ms`**.
   - Escolha Add metric (Adicionar métrica).

   Em Audiences (Públicos), selecione 100% para que todos os usuários sejam inseridos no experimento. Configure a divisão de tráfego entre as variações para 50% cada uma.

   Em seguida, escolha Create Experiment (Criar experimento). Depois dessa criação, o experimento não começa até que você diga Evidently para iniciá-lo.

Etapa 6: iniciar o experimento e testar o CloudWatch Evidently

As etapas finais são iniciar o experimento e iniciar a aplicação de exemplo.

Para iniciar o experimento tutorial

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, Evidently.
3. Selecione o projeto EvidentlySampleApp.

4. Escolha a guia Experiments (Experimentos).
5. Escolha o botão ao lado de pageLoadTime e escolha Actions (Ações), Start experiment (Iniciar experimento).
6. Escolha um horário de finalização do experimento.
7. Escolha Start experiment (Iniciar experimento).

   O experimento iniciará imediatamente.

Em seguida, inicie a aplicação de exemplo do Evidently com o seguinte comando:

```
npm install -f && npm start
```

Assim que a aplicação for iniciada, você será atribuído a uma das duas variações de recursos que estão sendo testadas. Uma variação exibe "20% de desconto" e a outra não. Continue atualizando a página para ver as diferentes variações.

> **Note**
>
> O Evidently tem avaliações persistentes. As avaliações de recursos são determinísticas, o que significa para os mesmos `entityId` e recurso, um usuário sempre receberá a mesma atribuição de variação. A única vez que as atribuições de variação mudam é quando uma entidade é adicionada a uma substituição ou o tráfego do experimento é discado.
> No entanto, para facilitar o uso do tutorial da aplicação de exemplo para você, o Evidently reatribui a avaliação do recurso da aplicação de exemplo toda vez que você atualiza a página para que você possa experimentar ambas as variações sem precisar adicionar substituições.

Solução de problemas

Recomendamos usar o `npm` versão 6.14.14. Se você observar algum erro durante a compilação ou ao iniciar a aplicação de exemplo e estiver usando uma versão diferente do npm, faça o seguinte.

Como instalar o **`npm`** versão 6.14.14

1. Use um navegador para se conectar a https://nodejs.org/download/release/v14.17.5/.
2. Baixe o arquivo node-v14.17.5.pkg e execute esse pkg para instalar o npm.

Se um erro `webpack not found` surgir, navegue para a pasta `evidently-sample-shopping-app` e tente o seguinte:

a. Exclua `package-lock.json`
b. Exclua `yarn-lock.json`
c. Exclua `node_modules`
d. Exclua a dependência do webpack de `package.json`
e. Execute o seguinte:

```
npm install -f && npm
```

Usar o CloudWatch RUM

Com o CloudWatch RUM, você pode realizar o monitoramento real do usuário para coletar e visualizar dados do lado do cliente sobre a performance da aplicação Web a partir de sessões reais do usuário praticamente em tempo real. Os dados que você pode visualizar e analisar incluem tempos de carregamento de página, erros no lado do cliente e comportamento do usuário. Ao visualizar esses dados, você pode vê-los todos agregados e também detalhados por navegadores e dispositivos que seus clientes usam.

Você pode usar os dados coletados para identificar e depurar rapidamente problemas de performance do lado do cliente. O CloudWatch RUM ajuda a visualizar anomalias na performance da aplicação e encontrar dados de depuração relevantes, como mensagens de erro, rastreamentos de pilha e sessões de usuário. Você também pode usar o RUM para entender a amplitude do impacto no usuário final, incluindo o número de usuários, áreas geográficas e navegadores usados.

Os dados do usuário final coletados para o CloudWatch RUM são retidos por 30 dias e, em seguida, excluídos automaticamente. Se você quiser manter os eventos RUM por mais tempo, você pode optar por fazer com que o monitor de aplicações envie cópias dos eventos para o CloudWatch Logs em sua conta. Em seguida, você pode ajustar o período de retenção para esse grupo de logs.

Para usar o RUM, crie um app monitor (monitor de aplicações) e insira algumas informações. O RUM gera um snippet do JavaScript para você colar na sua aplicação. O snippet extrai o código do cliente da Web RUM. O cliente da Web do RUM captura dados de uma porcentagem das sessões de usuário da aplicação, que é exibida em um painel pré-pronto. Você pode especificar a porcentagem de sessões de usuário da qual coletar dados.

O CloudWatch RUM é integrado ao Application Signals, que pode descobrir e monitorar serviços das suas aplicações, clientes, canários do Synthetics e dependências de serviços. Use o Application Signals para ver uma lista ou um mapa visual dos seus serviços, visualizar métricas de integridade com base nos seus objetivos de nível de serviço (SLOs) e fazer uma busca profunda para ver rastreamentos do X-Ray correlacionados para uma solução de problemas mais detalhada. Para ver as solicitações da página do cliente do RUM no Application Signals, ative o rastreamento ativo do X-Ray criando um monitor de aplicações ou configurando manualmente o cliente Web do RUM. Os clientes do RUM são exibidos no Mapa de serviços conectado aos serviços e na página Detalhes do serviço dos serviços que eles chamam.

O cliente da Web do RUM é de código aberto. Para obter mais informações, consulte Cliente da Web do CloudWatch RUM.

Considerações sobre a performance

Esta seção discute as considerações de performance no uso do CloudWatch RUM.

- Impacto na performance da carga: o cliente da Web do CloudWatch RUM pode ser instalado na aplicação Web como um módulo JavaScript ou carregado nessa aplicação de forma assíncrona por meio de uma rede de entrega de conteúdo (CDN). Ele não bloqueia o processo de carregamento da aplicação. O CloudWatch RUM foi projetado para que não causar impacto perceptível no tempo de carregamento da aplicação.
- Impacto do runtime— O cliente da Web do RUM executa o processamento para registrar e despachar dados de RUM para o serviço CloudWatch RUM. Como os eventos são pouco frequentes e a quantidade de processamento é pequena, o CloudWatch RUM foi projetado para que não haja impacto detectável na performance da aplicação.
- Impacto na rede— O cliente da Web do RUM envia dados periodicamente para o serviço CloudWatch RUM. Os dados são despachados em intervalos regulares enquanto a aplicação está em execução e também imediatamente antes do navegador descarregá-la. Os dados enviados imediatamente antes do navegador descarregar a aplicação são enviados como beacons, que são projetados para não ter impacto detectável no tempo de descarga da aplicação.

Definição de preço do RUM

Com o CloudWatch RUM, a cobrança é feita por cada evento do RUM que o CloudWatch RUM recebe. Cada item de dados coletado usando o cliente da Web do RUM é considerado um evento do RUM. Entre os exemplos de eventos RUM estão uma visualização de página, um erro de JavaScript e um erro HTTP. Há opções para definir quais tipos de eventos serão coletados por cada monitor de

aplicações. Você pode ativar ou desativar opções para coletar eventos de telemetria de performance, erros de JavaScript, erros HTTP e rastreamentos de X-Ray. Para obter mais informações sobre a escolha dessas opções, consulte Etapa 2: Criar um monitor de aplicações e Informações coletadas pelo cliente da Web CloudWatch RUM. Para obter mais informações sobre a definição de preço, consulte Preços do Amazon CloudWatch.

Disponibilidade de regiões

No momento, o CloudWatch RUM está disponível nas seguintes regiões:

- Leste dos EUA (Norte da Virgínia)
- Leste dos EUA (Ohio)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)
- África (Cidade do Cabo)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Melbourne)
- Ásia-Pacífico (Osaka)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milão)
- Europe (Paris)
- Europa (Espanha)
- Europa (Estocolmo)
- Europa (Zurique)

- Oriente Médio (Barém)
- Oriente Médio (Emirados Árabes Unidos)
- América do Sul (São Paulo)

Tópicos

Políticas do IAM para uso do CloudWatch RUM

Para poder gerenciar totalmente o CloudWatch RUM, você deve estar conectado como um usuário do IAM ou função que tenha uma Política do IAM AmazonCloudWatchRUMFullAccess. Além disso, podem ser necessárias outras políticas ou permissões:

- Para criar um monitor de aplicações que gere um novo grupo de identidades do Amazon Cognito para autorização, você precisa ter a função do IAM Admin ou a política do IAM AdministratorAccess.
- Para criar um monitor de aplicações que envie dados para o CloudWatch Logs, você deve estar conectado a uma função ou política do IAM que tenha a seguinte permissão:

```
{
```

```
    "Effect": "Allow",
    "Action": [
        "logs:PutResourcePolicy"
    ],
    "Resource": [
        "*"
    ]
}
```

Outros usuários que precisam visualizar os dados, mas não necessitam criar recursos do CloudWatch RUM podem receber a política AmazonCloudWatchRUMReadOnlyAccess.

Configurar uma aplicação para usar o CloudWatch RUM

Siga as etapas nessas seções para configurar sua aplicação para começar a usar o CloudWatch RUM na coleta de dados de performance de sessões reais do usuário.

Tópicos

- Etapa 1: Autorizar sua aplicação a enviar dados para a AWS
- Etapa 2: Criar um monitor de aplicações
- (Opcional) Etapa 3: Modificar manualmente o snippet de código para configurar o cliente da Web do CloudWatch RUM
- Etapa 4: Inserir o snippet de código na sua aplicação
- Etapa 5: Testar a configuração do monitor de aplicações gerando eventos do usuário

Etapa 1: Autorizar sua aplicação a enviar dados para a AWS

Para usar o CloudWatch RUM, sua aplicação deve ter autorização.

Você tem três opções para configurar a autorização:

- Deixe o CloudWatch RUM criar um grupo de identidades do Amazon Cognito para a aplicação. Esse método requer o menor esforço de configuração. Essa é a opção padrão.

  O grupo de identidades conterá uma identidade não autenticada. Isso permite que o cliente da Web do CloudWatch RUM envie dados para o CloudWatch RUM sem autenticar o usuário da aplicação.

O grupo de identidades do Amazon Cognito tem uma função do IAM anexada. A identidade não autenticada do Amazon Cognito permite que o cliente da Web assuma a função do IAM autorizada a enviar dados para o CloudWatch RUM.

- Usar um grupo de identidades do Amazon Cognito já existente. Nesse caso, você também deve modificar a função do IAM anexada ao grupo de identidades. Use esta opção para bancos de identidades que oferecem suporte para usuários não autenticados. Você pode usar somente bancos de identidades da mesma região.
- Use a autenticação de um provedor de identidade existente que você já configurou. Nesse caso, você deve obter credenciais do provedor de identidade e sua aplicação deve encaminhá-las para o cliente da Web do RUM.

  Use esta opção para bancos de identidades que oferecem suporte somente para usuários autenticados.

As seções a seguir apresentam mais detalhes sobre essas opções.

O CloudWatch RUM cria um novo grupo de identidades do Amazon Cognito

Essa é a opção mais simples de configurar e, se você a escolher, não serão necessárias outras etapas de configuração. Você deve ter permissões administrativas para usar essa opção. Para ter mais informações, consulte Políticas do IAM para uso do CloudWatch RUM.

Com essa opção, o CloudWatch RUM cria os seguintes recursos:

- Um novo grupo de identidades do Amazon Cognito
- Uma identidade não autenticada do Amazon Cognito. Isso permite que o cliente da Web do RUM assuma uma função do IAM sem autenticar o usuário da aplicação.
- A função do IAM que o cliente da Web do RUM assumirá. A política do IAM anexada a essa função permite que ela use a API `PutRumEvents` com o recurso de monitor de aplicações. Em outras palavras, permite que o cliente da Web do RUM envie dados para o RUM.

O cliente da Web do RUM usa a identidade do Amazon Cognito para obter credenciais da AWS. As credenciais AWS são associadas à função do IAM. A função do IAM está autorizada a usar a `PutRumEvents` com o recurso AppMonitor.

O Amazon Cognito envia o token de segurança necessário para permitir que sua aplicação envie dados para o CloudWatch RUM. O snippet de código do JavaScript gerado pelo CloudWatch RUM inclui as linhas a seguir para habilitar a autenticação.

```
   {
    identityPoolId: [identity pool id],  // e.g., 'us-west-2:EXAMPLE4a-66f6-4114-902a-
EXAMPLEbad7'
   }
);
```

Usar um grupo existente de identidades do Amazon Cognito

Se você optar por usar um grupo de identidades do Amazon Cognito já existente, especifique o grupo de identidades ao adicionar a aplicação ao CloudWatch RUM. O grupo deve ser compatível com a habilitação do acesso a identidades não autenticadas. Você pode usar somente bancos de identidades da mesma região.

Você também deve adicionar as seguintes permissões à política do IAM anexada à função do IAM associada a esse grupo de identidades.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rum:PutRumEvents",
            "Resource": "arn:aws:rum:[region]:[accountid]:appmonitor/[app monitor
 name]"
        }
    ]
}
```

O Amazon Cognito enviará o token de segurança necessário para permitir que sua aplicação acesse o CloudWatch RUM.

Provedor de terceiros

Se você decidir usar a autenticação privada de um provedor terceirizado, deverá obter credenciais do provedor de identidade e encaminhá-las para a AWS. A melhor maneira de fazer isso é usar um

fornecedor de token de segurança. Você pode usar qualquer fornecedor de token de segurança, incluindo o Amazon Cognito com o AWS Security Token Service. Para obter mais informações sobre o AWS STS, consulte Welcome to the AWS Security Token Service API Reference.

Se você quiser usar o Amazon Cognito como fornecedor de token nesse cenário, você pode configurá-lo para trabalhar com um provedor de autenticação. Para obter mais informações, consulte Conceitos básicos dos grupos de identidades do Amazon Cognito (identidades federadas).

Depois de configurar o Amazon Cognito para trabalhar com seu provedor de identidade, é necessário também fazer o seguinte:

- Criar uma função do IAM com as permissões a seguir. Sua aplicação usará essa função para acessar a AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rum:PutRumEvents",
            "Resource": "arn:aws:rum:[region]:[accountID]:appmonitor/[app monitor
 name]"
        }
    ]
}
```

- Adicione o seguinte à sua aplicação para que ela passe as credenciais do provedor para o CloudWatch RUM. Insira a linha para que ela seja executada depois que um usuário fizer login na aplicação e a aplicação receber as credenciais a serem usadas para acessar a AWS.

```
cwr('setAwsCredentials', {/* Credentials or CredentialProvider */});
```

Para obter mais informações sobre provedores de credenciais no SDK JavaScript da AWS, consulte Definir credenciais em um navegador da Web no guia do desenvolvedor v3 para SDK JavaScript, Configurar credenciais em um navegador da Web no guia do desenvolvedor v2 para SDK para JavaScript, e @aws -sdk/credencial-providers.

Você também pode usar o SDK para o cliente da Web do CloudWatch RUM configurar os métodos de autenticação do cliente da Web. Para obter mais informações sobre o cliente da Web, do SDK, consulte CloudWatch RUM web client SDK.

Etapa 2: Criar um monitor de aplicações

Para começar a usar o CloudWatch RUM com sua aplicação, você cria um app monitor (monitor de aplicações). Quando o monitor da aplicação é criado, o RUM gera um snippet do JavaScript para você colar em sua aplicação. O snippet extrai o código do cliente da Web RUM. O cliente da Web do RUM captura dados de uma porcentagem das sessões de usuário da aplicação e os envia para o RUM.

Para criar um monitor de aplicações

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, RUM.
3. Escolha Add app monitor (Adicionar monitor de aplicações).
4. Inserir as informações e as configurações da aplicação:

   - Em App monitor name (Nome do monitor de aplicações), insira um nome a ser usado para identificar esse monitor de aplicações no console do CloudWatch RUM.
   - Em Application domain (Domínio da aplicação), insira o nome do domínio de nível superior em que sua aplicação tem autoridade administrativa. Ele deve estar em um formato de domínio de URL.

     Escolha Include sub domains (Incluir subdomínios) para que o monitor de aplicações também colete dados de todos os subdomínios sob o domínio de nível superior.
5. Em Configure RUM data collection (Configurar a coleta de dados do RUM), especifique se você deseja que o monitor de aplicações colete cada um dos seguintes dados:

   - Performance telemetry (Telemetria de performance) – Coleta informações sobre o carregamento da página e os tempos de carregamento de recursos
   - Erros de JavaScript (Erros de JavaScript): coleta informações sobre erros de JavaScript não tratados gerados pela sua aplicação
   - HTTP Errors (Erros HTTP): coleta informações sobre erros de HTTP lançados pela aplicação

   A seleção dessas opções fornece mais informações sobre sua aplicação, mas também gera mais eventos RUM do CloudWatch e, portanto, incorre em mais cobranças.

   Se você não selecionar nenhuma delas, o monitor de aplicações ainda coletará eventos de início de sessão e IDs de página para que você possa ver quantos usuários estão usando sua

aplicação, incluindo falhas por tipo e versão do sistema operacional, tipo e versão do navegador, tipo de dispositivo e localização.

6. Selecione Check this option to allow the CloudWatch RUM Web Client to set cookies (Marque esta opção para permitir que o CloudWatch RUM Web Client defina cookies) se quiser coletar IDs de usuário e IDs de sessão de sessões de usuário amostradas. As IDs de usuário são geradas aleatoriamente pelo RUM. Para ter mais informações, consulte Cookies do cliente da Web do CloudWatch RUM (ou tecnologias semelhantes).
7. Em Sessions sample (Exemplos de sessão), insira a porcentagem de sessões do usuário que serão usadas para coletar dados do RUM. O padrão é 100%. A redução desse número fornece menos dados, mas diminui suas cobranças. Para obter mais informações sobre a definição de preços do RUM, consulte Definição de preços do RUM.
8. Os dados do usuário final coletados para o CloudWatch RUM são retidos por 30 dias e depois excluídos. Se você quiser manter cópias de eventos RUM no CloudWatch Logs e configurar por quanto tempo reter essas cópias, escolha Check this option to store your application telemetry data in your CloudWatch Logs account (Marque esta opção para armazenar os dados de telemetria da aplicação em sua conta do CloudWatch Logs) abaixo de Data storage (Armazenamento de dados). Por padrão, o grupo de logs do CloudWatch Logs retém os dados por 30 dias. Você pode ajustar seu período de retenção de logs no console de Logs do CloudWatch.
9. Em Autorization (Autorização), especifique se deve ser usado um grupo de identidades do Amazon Cognito novo ou existente ou se deve se usar um provedor de identidade diferente. A criação de um novo grupo de identidades é a opção mais simples, que não requer outras etapas de configuração. Para ter mais informações, consulte Etapa 1: Autorizar sua aplicação a enviar dados para a AWS.

   A criação de um novo grupo de identidades do Amazon Cognito requer permissões administrativas. Para ter mais informações, consulte Políticas do IAM para uso do CloudWatch RUM.
10. (Opcional) Por padrão, quando você adiciona o snippet de código RUM à aplicação, o cliente da Web injeta a etiqueta JavaScript para monitorar o uso no código HTML de todas as páginas da aplicação. Para alterar isso, escolha Configure pages (Configurar páginas) e, em seguida, escolha Include only these pages (Incluir somente essas páginas) ou Exclude these pages (Excluir essas páginas). Em seguida, especifique as páginas a serem incluídas ou excluídas. Para especificar uma página a ser incluída ou excluída, insira as URLs completas. Para especificar páginas adicionais, escolha Add URL (Adicionar URL).

11. Para habilitar o rastreamento do AWS X-Ray para as sessões de usuário incluídas na amostra obtida pelo monitor de aplicações, escolha Rastreamento ativo e selecione Rastrear meu serviço com o AWS X-Ray.

    Se você fizer essa seleção, as solicitações `XMLHttpRequest` e `fetch` feitas durante as sessões do usuário amostradas pelo monitor de aplicações serão rastreadas. Você pode visualizar os rastreamentos e segmentos dessas sessões de usuário no painel do RUM e no mapa de rastreamento e nas páginas de detalhes do rastreamento do X-Ray. Essas sessões de usuário também aparecerão como páginas de clientes no Application Signals depois que você as tiver habilitado para a aplicação.

    Ao fazer alterações de configuração adicionais no cliente da Web do CloudWatch RUM, você pode adicionar um cabeçalho de rastreamento do X-Ray às solicitações HTTP para permitir o rastreamento de ponta a ponta das sessões do usuário até o downstream de serviços gerenciados pela AWS. Para ter mais informações, consulte Habilitar o rastreamento do X-Ray de ponta a ponta.

12. (Opcional) Para adicionar etiquetas ao monitor de aplicações, escolha Tags (Etiquetas), Add new tag (Adicionar nova etiqueta).

    Em seguida, em Key (Chave), insira um nome para a etiqueta. É possível adicionar um valor opcional para a tag em Valor.

    Para adicionar outra etiqueta, escolha novamente Add new tag (Adicionar nova etiqueta).

    Para obter mais informações, consulte Etiquetar recursos daAWS.

13. Escolha Add app monitor (Adicionar monitor de aplicações).

14. Na seção Código de exemplo, é possível copiar o snippet de código a ser usado para adicioná-lo à aplicação. Recomendamos escolher JavaScript ou TypeScript e usar o NPM para instalar o cliente web CloudWatch RUM como um módulo JavaScript.

    Como alternativa, é possível escolher HTML para usar uma rede de entrega de conteúdo (CDN) para instalar o cliente da Web do CloudWatch RUM. A desvantagem de usar uma CDN é que o cliente da Web muitas vezes é bloqueado por bloqueadores de anúncios.

15. Escolha Copy (Copiar) ou Download (Fazer download) e, em seguida, escolha Done (Pronto).

(Opcional) Etapa 3: Modificar manualmente o snippet de código para configurar o cliente da Web do CloudWatch RUM

Você pode modificar o snippet de código antes de inseri-lo em sua aplicação, para ativar ou desativar várias opções. Para obter mais informações, consulte a Documentação do cliente da Web do CloudWatch RUM.

Existem três opções de configuração que você definitivamente deve conhecer, conforme discutido nessas seções.

Impedir a coleta de URLs de recursos que possam conter informações pessoais

Por padrão, o cliente da Web do CloudWatch RUM é configurado para registrar as URLs de recursos baixados pela aplicação. Esses recursos incluem arquivos HTML, imagens, arquivos CSS, arquivos JavaScript e assim por diante. Em algumas aplicações, os URLs podem conter informações de identificação pessoal (PII).

Se esse for o caso da sua aplicação, recomendamos que você desabilite a coleção de URLs de recursos definindo `recordResourceUrl: false` na configuração do snippet de código antes de inseri-lo em sua aplicação.

Registrar visualizações de página manualmente

Por padrão, o cliente da Web registra as visualizações de página quando ela é carregada pela primeira vez e quando a API de histórico do navegador é chamada. O ID da página padrão é `window.location.pathname`. No entanto, em alguns casos, talvez você queira substituir esse comportamento e instrumentar a aplicação para registrar visualizações de página de forma programática. Isso permite que você controle o ID da página e quando ele é gravado. Por exemplo, considere uma aplicação da Web que tenha um URI com um identificador de variável, como `/entity/123` ou `/entity/456`. Por padrão, o CloudWatch RUM gera um evento de visualização de página para cada URI com um ID de página distinto correspondente ao nome do caminho, mas talvez você queira agrupá-los pelo mesmo ID de página. Para fazer isso, desative a automação de visualização de página do cliente Web usando a configuração `disableAutoPageView` e use o comando `recordPageView` para definir o ID da página desejada. Para obter mais informações, consulte Configurações específicas da aplicação no GitHub.

Exemplo de script incorporado:

```
cwr('recordPageView', { pageId: 'entityPageId' });
```

Exemplo de módulo JavaScript:

```
awsRum.recordPageView({ pageId: 'entityPageId' });
```

Habilitar o rastreamento do X-Ray de ponta a ponta

Quando você cria o monitor de aplicações, selecionar Rastrear meu serviço com o AWS X-Ray habilita o rastreamento de solicitações de `XMLHttpRequest` e `fetch` feitas durante as sessões de usuário incluídas na amostra obtida pelo monitor de aplicações. Você pode visualizar os rastreamentos dessas solicitações HTTP no painel do RUM e no mapa de rastreamento e nas páginas de detalhes de rastreamento do X-Ray.

Por padrão, esses rastreamentos do lado do cliente não estão conectados a rastreamentos do lado do servidor downstream. Para conectar rastreamentos do lado do cliente a rastreamentos do lado do servidor e habilitar o rastreamento de ponta a ponta, defina a opção `addXRayTraceIdHeader` para `true` no cliente da Web. Isso faz com que o cliente da Web do CloudWatch RUM adicione um cabeçalho de rastreamento do X-Ray às solicitações HTTP.

O bloco de código a seguir mostra um exemplo de adição de rastreamentos do lado do cliente. Algumas opções de configuração são omitidas dessa amostra para oferecer melhor leitura.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            enableXRay: true,
            telemetries: [
                'errors',
                'performance',
                [ 'http', { addXRayTraceIdHeader: true } ]
            ]
        }
    );
</script>
```

> **Warning**
>
> Configurar o cliente da Web do CloudWatch RUM para adicionar um cabeçalho de rastreamento do X-Ray às solicitações HTTP pode fazer com que o compartilhamento de recursos de origem cruzada (CORS) falhe ou invalide a assinatura da solicitação, se essa solicitação for assinada com Sigv4. Para obter mais informações, consulte a Documentação do cliente da Web do CloudWatch RUM. É altamente recomendável que você teste sua aplicação antes de adicionar um cabeçalho de rastreamento do X-Ray do lado do cliente em um ambiente de produção.

Para obter mais informações consulte a Documentação do cliente da Web do CloudWatch RUM

Etapa 4: Inserir o snippet de código na sua aplicação

Em seguida, insira o snippet de código criado na seção anterior na aplicação.

> **Warning**
>
> O cliente da Web, baixado e configurado pelo snippet de código, usa cookies (ou tecnologias semelhantes) para ajudar a coletar dados do usuário final. Antes de inserir o snippet do código, consulte Filtrar por atributos de metadados no console.

Se você não tiver o snippet de código gerado anteriormente, poderá encontrá-lo seguindo as instruções no Como encontro um snippet de código que já gerei?.

Para inserir o snippet de código RUM do CloudWatch em sua aplicação

1. Insira o snippet de código que você copiou ou baixou na seção anterior dentro do elemento <head> da aplicação. Insira-o antes do elemento <body> ou de qualquer outra etiqueta <script>.

   O seguinte exemplo mostra um de um snippet de código gerado:

```
<script>
(function (n, i, v, r, s, c, x, z) {
    x = window.AwsRumClient = {q: [], n: n, i: i, v: v, r: r, c: c};
    window[n] = function (c, p) {
```

```
        x.q.push({c: c, p: p});
    };
    z = document.createElement('script');
    z.async = true;
    z.src = s;
    document.head.insertBefore(z, document.getElementsByTagName('script')[0]);
})('cwr',
    '194a1c89-87d8-41a3-9d1b-5c5cd3dafbd0',
    '1.0.0',
    'us-east-2',
    'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
    {
        sessionSampleRate: 1,
        identityPoolId: "us-east-2:c90ef0ac-e3b8-4d1a-b313-7e73cfd21443",
        endpoint: "https://dataplane.rum.us-east-2.amazonaws.com",
        telemetries: ["performance", "errors", "http"],
        allowCookies: true,
        enableXRay: false
    });
</script>
```

2. Se a aplicação for uma aplicação Web de várias páginas, você deverá repetir a etapa 1 para cada página HTML que deseja incluir na coleta de dados.

Etapa 5: Testar a configuração do monitor de aplicações gerando eventos do usuário

Depois que você inserir o snippet de código e que a aplicação atualizada estiver em execução, você poderá testá-la gerando eventos do usuário manualmente. Para esse teste, recomendamos que faça o seguinte. Esse teste gera cobranças padrão do CloudWatch RUM.

- Navegar entre páginas em sua aplicação Web.
- Criar várias sessões de usuário, usando navegadores e dispositivos diferentes.
- Fazer solicitações.
- Causar erros do JavaScript.

Depois de gerar alguns eventos, visualize-os no painel do CloudWatch RUM. Para ter mais informações, consulte Visualizar o painel do CloudWatch RUM.

Os dados das sessões do usuário podem levar até 15 minutos para serem exibidos no painel.

Se você não vir dados 15 minutos após gerar eventos na aplicação, consulte Solucionar problemas do CloudWatch RUM.

Configurar o cliente da Web do CloudWatch RUM

As aplicações podem usar um dos snippets de código gerados pelo CloudWatch RUM para instalar o cliente da Web do CloudWatch RUM. Os snippets gerados oferecem suporte a dois métodos de instalação: como um módulo JavaScript via NPM ou por meio de uma rede de entrega de conteúdo (CDN). Para proporcionar o melhor desempenho, recomendamos usar o método de instalação via NPM. Para obter mais informações sobre o uso desse método, consulte Instalar como um módulo JavaScript.

Se você usar a opção de instalação por CDN, os bloqueadores de anúncios poderão bloquear a CDN padrão fornecida pelo CloudWatch RUM. Isso desabilita o monitoramento de aplicações para usuários com bloqueadores de anúncios instalados. Por isso, recomendamos usar a CDN padrão somente para integração inicial com o CloudWatch RUM. Para obter mais informações sobre as formas de mitigar esse problema, consulte Instrumentar a aplicação.

O snippet de código permanece na etiqueta <head> de um arquivo HTML e instala o cliente da Web fazendo o download o cliente da Web e, em seguida, configurando o cliente da Web para a aplicação monitorada. O snippet é uma função autoexecutável que se parece com a seguinte. Neste exemplo, o corpo da função do snippet foi omitido para legibilidade.

```
<script>
(function(n,i,v,r,s,c,u,x,z){...})(
'cwr',
'00000000-0000-0000-0000-000000000000',
'1.0.0',
'us-west-2',
'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
{ /* Configuration Options Here */ }
);
<script>
```

Argumentos

O snippet de código aceita seis argumentos:

- Um namespace para executar comandos no cliente da Web, como `'cwr'`
- O ID do monitor de aplicações, como `'00000000-0000-0000-0000-000000000000'`

- A versão da aplicação, como `'1.0.0'`
- A região da AWS do monitor de aplicações, como `'us-west-2'`
- O URL do cliente da Web, como `'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js'`
- Opções de configuração específicas da aplicação. Para obter mais informações, consulte a seção a seguir.

Ignorar erros

O cliente Web do CloudWatch RUM escuta todos os tipos de erros que ocorrem nas suas aplicações. Se a aplicação emitir erros de JavaScript que você não deseja visualizar no painel do CloudWatch RUM, o cliente Web do CloudWatch RUM poderá ser configurado para filtrar esses erros. Assim, você verá apenas os eventos de erro relevantes no painel do CloudWatch RUM. Por exemplo, você pode optar por não visualizar alguns erros de JavaScript no painel porque já identificou uma correção para eles e o volume desses erros está mascarando outros. Você também pode ignorar erros que não podem ser corrigidos porque pertencem a uma biblioteca de terceiros.

Para obter mais informações sobre como instrumentar o cliente Web para filtrar erros específicos de JavaScript, consulte o exemplo em Errors (Erros) na documentação do cliente Web do Github.

Opções de configuração

Para obter informações sobre as opções de configuração disponíveis para o cliente da Web CloudWatch RUM, consulte aDocumentação do cliente da Web CloudWatch RUM

Regionalização

Esta seção ilustra estratégias para usar o CloudWatch RUM com aplicações em diferentes regiões.

Minha aplicação da Web é implantada em várias regiões da AWS

Se a sua aplicação da Web for implantada em várias regiões da AWS, você tem três opções:

- Implantar um monitor de aplicações em uma região, em uma conta, atendendo a todas as regiões.
- Implantar monitores de aplicações separados para cada região, em contas exclusivas.
- Implantar monitores de aplicações separados para cada região, tudo em uma conta.

A vantagem de usar um monitor de aplicações é que todos os dados serão centralizados em uma visualização e todos os logs serão gravados no mesmo grupo de logs no CloudWatch Logs. Com um único monitor de aplicações há uma pequena latência adicional para solicitações e um único ponto de falha.

O uso de vários monitores de aplicações remove o único ponto de falha, mas impede que todos os dados sejam combinados em uma visualização.

O CloudWatch RUM não foi lançado em algumas regiões nas quais minha aplicação está implantada

O CloudWatch RUM é lançado em várias regiões e tem ampla cobertura geográfica. Ao configurar o CloudWatch RUM nas regiões em que ele está disponível, é possível obter os benefícios. Os usuários finais podem estar em qualquer lugar e ainda ter suas sessões incluídas se você tiver configurado um monitor de aplicações na região à qual eles estão se conectando.

No entanto, o CloudWatch RUM ainda não foi lançado em AWS GovCloud (Leste dos EUA), AWS GovCloud (Oeste dos EUA) ou em qualquer região da China. Você não pode enviar dados para o CloudWatch RUM a partir dessas regiões.

Usar grupos de páginas

Use grupos de páginas para associar diferentes páginas na aplicação e poder ver análises agregadas para esses grupos. Por exemplo, você pode ver o tempo de carregamento agregado de todas as suas páginas de destino.

Para criar um grupo de páginas, adicione uma ou mais etiquetas aos eventos de visualização de página no cliente Web do CloudWatch RUM. Os exemplos a seguir colocam a página `/home` nos grupos de páginas `en` e `landing`.

Exemplo de script incorporado

```
cwr('recordPageView', { pageId: '/home', pageTags: ['en', 'landing']});
```

Exemplo de módulo JavaScript

```
awsRum.recordPageView({ pageId: '/home', pageTags: ['en', 'landing']});
```

> **Note**
>
> Os grupos de páginas têm como objetivo facilitar a agregação de análises em diferentes páginas. Para obter informações sobre como definir e manipular `pageIds` para sua aplicação, consulte a seção Registro manual de visualizações de páginas em (Opcional) Etapa 3: Modificar manualmente o snippet de código para configurar o cliente da Web do CloudWatch RUM.

Especificar metadados personalizados

O CloudWatch RUM anexa dados adicionais a cada evento como metadados. Metadados de eventos consistem em atributos sob a forma de pares chave-valor. Você pode usar esses atributos para pesquisar ou filtrar eventos no console do CloudWatch RUM. Por padrão, o CloudWatch RUM cria alguns metadados para você. Para obter mais informações sobre metadados padrão, consulte Metadados de evento do RUM.

Você também pode usar o cliente da Web do CloudWatch RUM para adicionar metadados personalizados aos eventos do CloudWatch RUM. Os metadados personalizados podem incluir atributos de sessão e atributos de página.

Para adicionar metadados personalizados, você deve usar a versão 1.10.0 ou posterior do cliente da Web do CloudWatch RUM.

Requisitos e sintaxe

Cada evento pode incluir até 10 atributos personalizados nos metadados. Os requisitos de sintaxe para atributos personalizados são os seguintes:

- Chaves
  - Máximo de 128 caracteres
  - Podem incluir caracteres alfanuméricos, dois pontos (:) e sublinhas (_)
  - Não podem começar com `aws:`.
  - Não pode consistir inteiramente em nenhuma das palavras-chave reservadas listadas na seção a seguir. É permitido usar essas palavras-chave como parte de um nome de chave mais longo.
- Valores
  - Máximo de 256 caracteres

- Devem ser strings, números ou valores boolianos

Palavras-chave reservadas

Você não pode usar as palavras-chave reservadas a seguir como nomes de chave inteiros. Você pode usar as palavras-chave a seguir como parte de um nome de chave mais longo, como `applicationVersion`.

- `browserLanguage`
- `browserName`
- `browserVersion`
- `countryCode`
- `deviceType`
- `domain`
- `interaction`
- `osName`
- `osVersion`
- `pageId`
- `pageTags`
- `pageTitle`
- `pageUrl`
- `parentPageId`
- `platformType`
- `referrerUrl`
- `subdivisionCode`
- `title`
- `url`
- `version`

> **Note**
>
> O CloudWatch RUM remove os atributos personalizados dos eventos do RUM se um atributo incluir uma chave ou valor que não seja válido, ou se o limite de 10 atributos personalizados por evento já tiver sido atingido.

Adicionar atributos de sessão

Se você configurar atributos de sessão personalizados, eles serão adicionados a todos os eventos de uma sessão. Você configura os atributos de sessão durante a inicialização do cliente da Web do CloudWatch RUM ou no runtime usando o comando `addSessionAttributes`.

Por exemplo, você pode adicionar a versão da aplicação como um atributo de sessão. Em seguida, no console do CloudWatch RUM, você pode filtrar os erros por versão para descobrir se uma taxa de erro maior está associada a uma determinada versão da aplicação.

Adicionar um atributo de sessão na inicialização, exemplo de NPM

A seção de código em negrito adiciona o atributo de sessão.

```
import { AwsRum, AwsRumConfig } from 'aws-rum-web';

try {
  const config: AwsRumConfig = {
    allowCookies: true,
    endpoint: "https://dataplane.rum.us-west-2.amazonaws.com",
    guestRoleArn: "arn:aws:iam::000000000000:role/RUM-Monitor-us-
west-2-000000000000-00xx-Unauth",
    identityPoolId: "us-west-2:00000000-0000-0000-0000-000000000000",
    sessionSampleRate: 1,
    telemetries: ['errors', 'performance'],
    sessionAttributes: {
        applicationVersion: "1.3.8"
    }
  };

  const APPLICATION_ID: string = '00000000-0000-0000-0000-000000000000';
  const APPLICATION_VERSION: string = '1.0.0';
  const APPLICATION_REGION: string = 'us-west-2';

  const awsRum: AwsRum = new AwsRum(
```

```
    APPLICATION_ID,
    APPLICATION_VERSION,
    APPLICATION_REGION,
    config
  );
} catch (error) {
  // Ignore errors thrown during CloudWatch RUM web client initialization
}
```

Adicionar um atributo de sessão no runtime, exemplo de NPM

```
awsRum.addSessionAttributes({
    applicationVersion: "1.3.8"
})
```

Adicionar um atributo de sessão na inicialização, exemplo de script integrado

A seção de código em negrito adiciona o atributo de sessão.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            sessionSampleRate:1,
            guestRoleArn:'arn:aws:iam::000000000000:role/RUM-Monitor-us-
west-2-000000000000-00xx-Unauth',
            identityPoolId:'us-west-2:00000000-0000-0000-0000-000000000000',
            endpoint:'https://dataplane.rum.us-west-2.amazonaws.com',
            telemetries:['errors','http','performance'],
            allowCookies:true,
            sessionAttributes: {
                applicationVersion: "1.3.8"
            }
        }
    );
</script>
```

Adicionar um atributo de sessão no runtime, exemplo de script integrado

```
<script>
    function addSessionAttribute() {
        cwr('addSessionAttributes', {
            applicationVersion: "1.3.8"
        })
    }

</script>
```

Adicionar atributos de página

Se você configurar atributos de página personalizados, eles serão adicionados a todos os eventos na página atual. Você configura os atributos de página durante a inicialização do cliente da Web do CloudWatch RUM ou no runtime usando o comando `recordPageView`.

Por exemplo, você pode adicionar seu modelo de página como um atributo de página. Em seguida, no console do CloudWatch RUM, você pode filtrar os erros por modelos de página para descobrir se uma taxa de erro maior está associada a um determinado modelo de página da aplicação.

Adicionar um atributo de página na inicialização, exemplo de NPM

A seção de código em negrito adiciona o atributo de página.

```
const awsRum: AwsRum = new AwsRum(
    APPLICATION_ID,
    APPLICATION_VERSION,
    APPLICATION_REGION,
    { disableAutoPageView:  true // optional }
);
awsRum.recordPageView({
    pageId:'/home',
    pageAttributes: {
      template: 'artStudio'
    }
});
const credentialProvider = new CustomCredentialProvider();
if(awsCreds) awsRum.setAwsCredentials(credentialProvider);
```

Adicionar um atributo de página no runtime, exemplo de NPM

```
awsRum.recordPageView({
    pageId: '/home',
```

```
    pageAttributes: {
        template: 'artStudio'
    }
});
```

Adicionar um atributo de página na inicialização, exemplo de script integrado

A seção de código em negrito adiciona o atributo de página.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            disableAutoPageView: true //optional
        }
    );
    cwr('recordPageView', {
       pageId: '/home',
       pageAttributes: {
           template: 'artStudio'
       }
    });
    const awsCreds = localStorage.getItem('customAwsCreds');
    if(awsCreds) cwr('setAwsCredentials', awsCreds)
</script>
```

Adicionar um atributo de página no runtime, exemplo de script integrado

```
<script>
    function recordPageView() {
        cwr('recordPageView', {
            pageId: '/home',
            pageAttributes: {
                template: 'artStudio'
            }
        });
    }
</script>
```

Filtrar por atributos de metadados no console

Para filtrar as visualizações no console do CloudWatch RUM com qualquer atributo de metadados integrado ou personalizado, use a barra de pesquisa. Na barra de pesquisa, você pode especificar até 20 termos de filtro sob a forma de chave=valor para aplicar às visualizações. Por exemplo, para filtrar dados somente para o navegador Chrome, você pode adicionar o termo de filtro browserName=Chrome.

Por padrão, o console do CloudWatch RUM recupera as 100 chaves de atributos e valores mais comuns para serem exibidos no menu suspenso na barra de pesquisa. Para adicionar mais atributos de metadados como termos de filtro, insira a chave e o valor do atributo inteiros na barra de pesquisa.

Um filtro pode incluir até 20 termos de filtro e você pode salvar até 20 filtros por monitor de aplicações. Quando você salva um filtro, ele é salvo na lista suspensa Saved filters (Filtros salvos). Também é possível excluir um filtro salvo.

Enviar eventos personalizados

O CloudWatch RUM registra e ingere os eventos listados em Informações coletadas pelo cliente da Web CloudWatch RUM. Se você usar a versão 1.12.0 ou posterior do cliente da Web do CloudWatch RUM, poderá definir, registrar e enviar eventos personalizados adicionais. Você define o nome do tipo de evento e os dados a serem enviados para cada tipo de evento definido. Cada carga útil de evento personalizado pode ter até 6 KB.

Os eventos personalizados só serão ingeridos se o monitor de aplicações tiver eventos personalizados habilitados. Para atualizar as configurações do monitor de aplicações, use o console do CloudWatch RUM ou a API UpdateAppMonitor.

Após habilitar eventos personalizados, e definir e enviar os eventos personalizados, você poderá pesquisá-los. Para isso, use a guia Events (Eventos) no console do CloudWatch RUM. Pesquisar usando o tipo de evento.

Requisitos e sintaxe

Os eventos personalizados consistem em um tipo de evento e detalhes do evento. Os requisitos para cada um deles são os seguintes:

- Tipo de evento

  - Isso pode ser o tipo ou o nome do evento. Por exemplo, o tipo de evento integrado do CloudWatch RUM denominadoJsError tem um tipo de evento de `com.amazon.rum.js_error_event`.
  - Deve ter de 1 a 256 caracteres.
  - Pode ser uma combinação de caracteres alfanuméricos, sublinhas, hifens e pontos.
- Detalhes do evento
  - Contém os dados reais que você deseja registrar no CloudWatch RUM.
  - Deve ser um objeto que consiste em campos e valores.

Exemplos de registro de eventos personalizados

Há duas maneiras de registrar eventos personalizados no cliente da Web do CloudWatch RUM.

- Usar a API `recordEvent` do cliente da Web do CloudWatch RUM.
- Usar um plug-in personalizado.

Enviar um evento personalizado usando a API **`recordEvent`**, exemplo de NPM

```
awsRum.recordEvent('my_custom_event', {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
)
```

Enviar um evento personalizado usando a API **`recordEvent`**, exemplo de script integrado

```
cwr('recordEvent', {
    type: 'my_custom_event',
    data: {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
```

```
            interaction_2 : "scroll"
        },
        visit_count:10
    }
})
```

Exemplo de envio de um evento personalizado usando um plug-in personalizado

```
// Example of a plugin that listens to a scroll event, and
// records a 'custom_scroll_event' that contains the timestamp of the event.
class MyCustomPlugin implements Plugin {
    // Initialize MyCustomPlugin.
    constructor() {
        this.enabled;
        this.context;
        this.id = 'custom_event_plugin';
    }
    // Load MyCustomPlugin.
    load(context) {
        this.context = context;
        this.enable();
    }
    // Turn on MyCustomPlugin.
    enable() {
        this.enabled = true;
        this.addEventHandler();
    }
    // Turn off MyCustomPlugin.
    disable() {
        this.enabled = false;
        this.removeEventHandler();
    }
    // Return MyCustomPlugin Id.
    getPluginId() {
        return this.id;
    }
    // Record custom event.
    record(data) {
        this.context.record('custom_scroll_event', data);
    }
    // EventHandler.
    private eventHandler = (scrollEvent: Event) => {
        this.record({timestamp: Date.now()})
```

```
    }
    // Attach an eventHandler to scroll event.
    private addEventHandler(): void {
        window.addEventListener('scroll', this.eventHandler);
    }
    // Detach eventHandler from scroll event.
    private removeEventHandler(): void {
        window.removeEventListender('scroll', this.eventHandler);
    }
}
```

Visualizar o painel do CloudWatch RUM

O CloudWatch RUM ajuda a coletar dados de sessões do usuário sobre a performance da aplicação, incluindo tempos de carregamento da página, pontuação do Apdex, navegadores e dispositivos usados, geolocalização de sessões de usuário e sessões com erros. Todas essas informações são exibidas em um painel.

Para visualizar o painel do RUM

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, RUM.

   A guia Overview (Visão geral) exibe informações coletadas por um dos monitores de aplicações que você criou.

   A fileira superior de painéis exibe as seguintes informações para este monitor de aplicações:

   - Número de carregamentos de página
   - Velocidade média de carregamento da página
   - Pontuação do Apdex
   - Status de todos os alarmes associado ao monitor de aplicações

   A pontuação do índice de performance da aplicação (Apdex) indica o nível de satisfação dos usuários finais. As pontuações variam de 0 (menos satisfeitos) a 1 (mais satisfeitos). As pontuações são baseadas apenas na performance da aplicação. Os usuários não são solicitados a classificar a aplicação. Para obter mais informações sobre as pontuações do Apdex, consulte Como o CloudWatch RUM define pontuações do Apdex.

Vários desses painéis incluem links que você pode usar para examinar mais os dados. A escolha de qualquer um desses links exibe uma visualização detalhada com as guias Performance, Erros, Solicitações de HTTP, Sessões, Navegadores e dispositivos de eventos e Jornada do usuário na parte superior da tela.

3. Para delimitar ainda mais as informações, escolha a guia List view (Visualização em lista) e, em seguida, escolha o nome do monitor de aplicações no qual deseja se concentrar. As guias a seguir do monitor de aplicações escolhido serão exibidas.

   - A guia Performance exibe informações de performance da página, incluindo tempos de carregamento, informações de sessão, informações de solicitação, sinais vitais da Web e número de carregamentos ao longo do tempo. Essa visualização inclui controles para alternar o foco da visualização ente Page loads (Carregamentos de página), Requests (Solicitações) e Location (Localização).
   - A guia Erros exibe informações de erro de Javascript, incluindo a mensagem de erro mais vista por usuários e os dispositivos e navegadores com a maioria dos erros. Esta visualização inclui um histograma dos erros e uma exibição em lista dos erros. É possível filtrar a lista de erros por detalhes de usuário e de evento. Escolha uma mensagem de erro para ver mais detalhes.
   - A guia Solicitações de HTTP exibe informações da solicitação de HTTP, incluindo o URL da solicitação com mais erros e os dispositivos e navegadores com mais erros. Esta guia inclui um histograma das solicitações, uma exibição de lista de solicitações e uma exibição de lista de erros de rede. É possível filtrar as listas por detalhes de usuário e de evento. Escolha um código de resposta ou uma mensagem de erro para ver mais detalhes sobre a solicitação ou o erro de rede, respectivamente.
   - A guia Sessões exibe as métricas da sessão. Esta guia inclui um histograma dos eventos de início da sessão e uma exibição em lista das sessões. É possível filtrar a lista de sessões por tipo de evento, detalhes de usuário e detalhes de evento. Escolha um sessionId para ver mais detalhes sobre uma sessão.
   - A guia Eventos exibe um histograma dos eventos do RUM e uma exibição em lista dos eventos. É possível filtrar a lista de eventos por tipo de evento, detalhes de usuário e detalhes de evento. Escolha um evento RUM para ver o evento bruto.
   - A guia Browser & Devices (Navegadores e dispositivos) exibe informações como a performance e o uso de diferentes navegadores e dispositivos para acessar sua aplicação. Esta visualização inclui controles para alternar o foco da visualização entre Navegadores e Dispositivos.

Se você restringir o escopo a um único navegador, verá os dados detalhados pela versão do navegador.

- A guia User Journey (Jornada do usuário) exibe os caminhos que seus clientes usam para navegar pela aplicação. É possível ver de onde seus clientes entram na aplicação e para qual página eles saem página eles saem da aplicação. Também é possível ver os caminhos que eles seguem e a porcentagem de clientes que seguem esses caminhos. Você pode pausar em um nó para obter mais detalhes sobre essa página. Você pode escolher um único caminho para destacar as conexões para facilitar a visualização.

4. (Opcional) Em qualquer uma das seis primeiras guias, é possível escolher o botão Páginas e selecionar uma página ou um grupo de páginas da lista. Isso reduz os dados exibidos para uma única página ou grupo de páginas da aplicação. Você também pode marcar páginas e grupos de páginas na lista como favoritos.

Como o CloudWatch RUM define pontuações do Apdex

O Apdex (Application Performance Index) é um padrão aberto que define um método para relatar, comparar e classificar o tempo de resposta da aplicação. Uma pontuação do Apdex ajuda você a entender e identificar o impacto na performance da aplicação ao longo do tempo.

A pontuação do Apdex indica que as pontuações do nível de satisfação dos usuários finais variam de 0 (menos satisfeitos) a 1 (mais satisfeitos). As pontuações são baseadas apenas na performance da aplicação. Os usuários não são solicitados a classificar a aplicação.

Cada pontuação individual do Apdex cai em um dos três limites. Com base no limite do Apdex e no tempo real de resposta da aplicação, há três tipos de performance, da seguinte forma:

- Satisfied (Satisfeito): o tempo real de resposta da aplicação é menor ou igual ao limite do Apdex. No CloudWatch RUM, esse limite é de 2000 ms ou menos.
- Tolerable (Tolerável): o tempo real de resposta da aplicação é maior que o limite do Apdex, mas menor ou igual a quatro vezes o limite do Apdex. No CloudWatch RUM, esse intervalo é de 2000 a 8000 ms.
- Frustrating (Frustrante): o tempo real de resposta da aplicação é maior que quatro vezes o limite do Apdex. No CloudWatch RUM, esse intervalo é superior a 8000 ms.

A pontuação total do Apdex 0-1 é calculada usando a seguinte fórmula:

```
(positive scores + tolerable scores/2)/total scores * 100
```

Métricas do CloudWatch que você pode coletar com o CloudWatch RUM

A tabela nesta seção lista as métricas que você coleta automaticamente com o CloudWatch RUM. Você também pode ver essas métricas no console do CloudWatch. Para ter mais informações, consulte Visualizar métricas disponíveis.

Você também pode, opcionalmente, enviar métricas estendidas para o CloudWatch ou o CloudWatch Evidently. Para ter mais informações, consulte Métricas estendidas.

Essas métricas estão publicadas no namespace da métrica chamado `AWS/RUM`. Todas as métricas a seguir são publicadas com uma dimensão `application_name`. O valor dessa dimensão é o nome do monitor de aplicações. Algumas métricas também são publicadas com dimensões adicionais, conforme listado na tabela.

Métrica	Unidade	Descrição
`HttpStatusCodeCount`	Contagem	A contagem de respostas HTTP na aplicação por seu código de status de resposta. Dimensões adicionais: • `event_details.response.status` é o código de status da resposta, como 200, 400, 404 e assim por diante. • `event_type` é o tipo de evento. No momento, o único valor possível para

Métrica	Unidade	Descrição
		esta dimensão é `http`.
`Http4xxCount`	Contagem	A contagem de respostas HTTP na aplicação, com o código 4xx de status da resposta. Elas são calculadas com base em eventos `http_event` do RUM que resultam em códigos 4xx.
`Http5xxCount`	Contagem	A contagem de respostas HTTP na aplicação, com o código 5xx de status da resposta. Elas são calculadas com base em eventos `http_event` do RUM que resultam em códigos 5xx.
`JsErrorCount`	Contagem	A contagem de eventos de erro do JavaScript ingeridos.

Métrica	Unidade	Descrição
`NavigationFrustratedCount`	Contagem	A contagem de eventos de navegação com um `duration` maior que o limite frustrante, que é 8000 ms. A duração dos eventos de navegação é rastreada na métrica `PerformanceNavigationDuration`.
`NavigationSatisfiedCount`	Contagem	A contagem de eventos de navegação com um `duration` menor que o objetivo Apdex, que é 2000 ms. A duração dos eventos de navegação é rastreada na métrica `PerformanceNavigationDuration`.

Métrica	Unidade	Descrição
`NavigationToleratedCount`	Contagem	A contagem de eventos de navegação com um `duration` entre 2000 ms e 8000 ms. A duração dos eventos de navegação é rastreada na métrica `PerformanceNavigationDuration` .
`PageViewCount`	Contagem	A contagem de eventos de visualização de página ingeridos pelo monitor de aplicações. Esse cálculo é feito contando os eventos `page_view_event` do RUM.

Métrica	Unidade	Descrição
`PerformanceResourceDuration`	Milissegundos	O `duration` de um evento de recurso. Dimensões adicionais: • `event_details.file.type` é o tipo de arquivo do evento de recurso, como uma folha de estilo, documento, imagem, script ou fonte. • `event_type` é o tipo de evento. No momento, o único valor possível para esta dimensão é `resource`.
`PerformanceNavigationDuration`	Milissegundos	O `duration` de um evento de navegação.

Métrica	Unidade	Descrição
`RumEventPayloadSize`	Bytes	O tamanho de cada evento ingerido pelo CloudWatch RUM. Você também pode usar a estatística `SampleCount` para essa métrica para monitorar o número de eventos que um monitor de aplicações está ingerindo.
`SessionCount`	Contagem	A contagem de eventos de início de sessão ingeridos pelo monitor de aplicaçõe s. Em outras palavras, o número de novas sessões iniciadas.
`WebVitalsCumulativeLayoutShift`	Nenhum	Rastreia o valor dos eventos de deslocamento de layout cumulativos.
`WebVitalsFirstInputDelay`	Milissegundos	Rastreia o valor dos primeiros eventos de atraso de entrada.
`WebVitalsLargestContentfulPaint`	Milissegundos	Acompanha o valor dos maiores eventos de contentful paint.

Métricas personalizadas e métricas estendidas que podem ser enviadas para o CloudWatch ou o CloudWatch Evidently

Por padrão, os monitores de aplicações do RUM enviam métricas para o CloudWatch. Essas métricas e dimensões padrão estão listadas em Métricas do CloudWatch que você pode coletar com o CloudWatch RUM.

Além disso, é possível configurar um monitor de aplicações para exportar métricas. O monitor de aplicações pode enviar métricas estendidas, métricas personalizadas ou ambas. Ele pode enviar as métricas para o CloudWatch, para o CloudWatch Evidently, ou para ambos.

- Métricas personalizadas: métricas personalizadas são métricas que você define. Com métricas personalizadas, é possível usar qualquer nome e namespace de métrica. Para derivar as métricas, é possível usar quaisquer eventos personalizados, eventos incorporados, atributos personalizados ou atributos padrão.

  É possível enviar métricas personalizadas para o CloudWatch ou o CloudWatch Evidently
- Métricas estendidas: permite enviar as métricas padrão do CloudWatch RUM para o CloudWatch Evidently para serem usadas em experimentos do Evidently. Também é possível enviar qualquer uma das métricas padrão do CloudWatch RUM para o CloudWatch com dimensões adicionais. Dessa forma, essas métricas podem oferecer uma visão mais minuciosa.

Tópicos

Métricas personalizadas

Para enviar métricas personalizadas, você deve usar as APIs da AWS ou a AWS CLI em vez do console. Para obter informações sobre como usar as APIs da AWS, consulte PutRumMetricsDestination e BatchCreateRumMetricDefinitions.

O número máximo de definições de métricas estendidas e métricas personalizadas que um destino pode conter é 2000. Para cada métrica personalizada ou métrica estendida que você envia a cada destino, cada combinação de nome de dimensão e valor de dimensão conta para esse limite. Isso também conta como uma métrica personalizada do CloudWatch para o cálculo de preços.

O exemplo a seguir mostra como criar uma métrica personalizada derivada de um evento personalizado. Aqui está o exemplo de evento personalizado usado:

```
cwr('recordEvent', {
    type: 'my_custom_event',
    data: {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
})
```

Com este evento personalizado, é possível criar uma métrica personalizada que contabilize o número de visitas ao URL `amazonaws.com` a partir de navegadores Chrome. A definição a seguir cria uma métrica chamada `AmazonVisitsCount` em sua conta, no namespace `RUM/CustomMetrics/PageVisits`.

```
{
    "AppMonitorName":"customer-appMonitor-name",
    "Destination":"CloudWatch",
    "MetricDefinitions":[
        {
            "Name":"AmazonVisitsCount",
            "Namespace":"PageVisit",
            "ValueKey":"event_details.visit_count",
            "UnitLabel":"Count",
            "DimensionKeys":{
                "event_details.current_url": "URL"
            },
            "EventPattern":"{\"metadata\":{\"browserName\":[\"Chrome\"]},\"event_type
\":[\"my_custom_event\"],\"event_details\": {\"current_url\": [\"amazonaws.com\"]}}"
        }
    ]
}
```

Métricas estendidas

Se configurar métricas estendidas, você poderá fazer um dos seguintes procedimentos ou ambos:

- Envie as métricas padrão do CloudWatch RUM para o CloudWatch Evidently para serem usadas em experimentos do Evidently. Somente as métricas PerformanceNavigationDuration, PerformanceResourceDuration, WebVitalsCumulativeLayoutShift, WebVitalsFirstInputDelay e WebVitalsLargestContentfulPaint podem ser enviadas para o Evidently.
- Envie qualquer métrica padrão do CloudWatch RUM para o CloudWatch com dimensões adicionais para que as métricas ofereçam uma visão mais minuciosa. Por exemplo, você pode ver as métricas específicas de um determinado navegador usado pelos usuários ou as métricas para os usuários em uma determinada geolocalização.

Para obter mais informações sobre as métricas padrão do CloudWatch RUM, consulte Métricas do CloudWatch que você pode coletar com o CloudWatch RUM.

O número máximo de definições de métricas estendidas e métricas personalizadas que um destino pode conter é 2000. Para cada métrica estendida ou métrica personalizada que você envia a cada destino, cada combinação de nome de dimensão e valor de dimensão conta como uma métrica estendida para esse limite. Isso também conta como uma métrica personalizada do CloudWatch para o cálculo de preços.

Quando você envia métricas estendidas para o CloudWatch, pode usar o console do CloudWatch RUM para criar alarmes do CloudWatch para essas métricas.

As métricas estendidas são cobradas como métricas personalizadas do CloudWatch. Para obter mais informações, consulte Preços do Amazon CloudWatch.

As dimensões a seguir são compatíveis com as métricas estendidas para todos os nomes de métricas que os monitores de aplicações podem enviar. Esses nomes de métricas estão listados em Métricas do CloudWatch que você pode coletar com o CloudWatch RUM.

- `BrowserName`

  Exemplo de valores de dimensão: `Chrome`, `Firefox`, `Chrome Headless`
- `CountryCode` O formato ISO-3166, com códigos de duas letras, é usado.

  Exemplo de valores de dimensão: `US`, `JP`, `DE`
- `DeviceType`

  Exemplo de valores de dimensão: `desktop`, `mobile`, `tablet`, `embedded`

- `FileType`

  Exemplo de valores de dimensão: `Image`, `Stylesheet`

- `OSName`

  Exemplo de valores de dimensão: `Linux`, `Windows, iOS`, `Android`

- `PageId`

Configurar métricas estendidas usando o console

Para usar o console para enviar métricas estendidas ao CloudWatch, use as etapas a seguir.

Para enviar métricas estendidas para o CloudWatch Evidently, você deve usar as APIs da AWS ou a AWS CLI em vez do console. Para obter informações sobre como usar as APIs da AWS para enviar métricas estendidas para o CloudWatch ou para o Evidently, consulte PutrumMetricsDestination e BatchCreateRumMetricDefinitions.

Para usar o console para configurar um monitor de aplicações e enviar métricas estendidas do RUM para o CloudWatch

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, RUM.
3. Escolha List view (Visualização em lista) e depois o nome do monitor de aplicações que deverá enviar as métricas.
4. Escolha a guia Configuration (Configuração) e escolha RUM extended metrics (Métricas estendidas do RUM).
5. Escolha Send metrics (Enviar métricas).
6. Selecione um ou mais nomes de métricas para enviar com dimensões adicionais.
7. Selecione um ou mais fatores para usar como dimensões para essas métricas. À medida que você faz suas escolhas, o número de métricas estendidas que são criadas é exibido em Number of extended metrics (Número de métricas estendidas).

   Esse número é calculado multiplicando o número de nomes de métricas escolhidos pelo número de dimensões diferentes que você cria. Esse número representa por quantas métricas personalizadas a cobrança é efetuada. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

a. Para enviar uma métrica com o ID da página como dimensão, escolha Browse for page ID (Procurar ID de página) e selecione os IDs de página a serem usados.

b. Para enviar uma métrica com o tipo de dispositivo como dimensão, escolha Desktop devices (Dispositivos desktop) ou Mobile and tablets (Celulares e tablets).

c. Para enviar uma métrica com o sistema operacional como dimensão, selecione um ou mais sistemas operacionais em Operating system (Sistema operacional).

d. Para enviar uma métrica com o tipo de navegador como dimensão, selecione um ou mais navegadores em Browsers (Navegadores).

e. Para enviar uma métrica com geolocalização como dimensão, selecione um ou mais locais em Locations (Locais).

   Somente os locais de onde esse monitor de aplicações relatou métricas aparecerão na lista para você escolher.

8. Quando você terminar de fazer suas seleções, escolha Send metrics (Enviar métricas).
9. (Opcional) Na lista Extended metrics (Métricas estendidas), para criar um alarme que monitore uma das métricas, escolha Create alarm (Criar alarme) na linha dessa métrica.

   Para obter informações gerais sobre alarmes do CloudWatch, consulte Usar alarmes do Amazon CloudWatch. Para ver um tutorial sobre como configurar um alarme em uma métrica estendida do CloudWatch RUM, consulte Tutorial: criar uma métrica estendida e definir um alarme para ela.

Parar de enviar métricas estendidas

Para usar o console para parar de enviar métricas estendidas

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, RUM.
3. Escolha List view (Visualização em lista) e depois o nome do monitor de aplicações que deverá enviar as métricas.
4. Escolha a guia Configuration (Configuração) e escolha RUM extended metrics (Métricas estendidas do RUM).
5. Selecione uma ou mais combinações de nome de métrica e dimensão da métrica para parar de enviar. Em seguida, escolha Actions (Ações), Delete (Excluir).

Tutorial: criar uma métrica estendida e definir um alarme para ela

Este tutorial demonstra como configurar uma métrica estendida para ser enviada ao CloudWatch e, em seguida, como definir um alarme para essa métrica. Neste tutorial, você cria uma métrica que rastreia erros de JavaScript no navegador Chrome.

Para configurar essa métrica estendida e definir um alarme para ela

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, RUM.
3. Escolha List view (Visualização em lista) e escolha o nome do monitor de aplicações que deverá enviar a métrica.
4. Escolha a guia Configuration (Configuração) e escolha RUM extended metrics (Métricas estendidas do RUM).
5. Escolha Send metrics (Enviar métricas).
6. Selecione JSErrorCount.
7. Em Browsers (Navegadores), selecione Chrome.

   Essa combinação de JSErrorCount e Chrome enviará uma métrica estendida para o CloudWatch. A métrica conta os erros de JavaScript apenas para as sessões de usuário que usam o navegador Chrome. O nome da métrica será JSErrorCount e o nome da dimensão será Browser (Navegador).
8. Escolha Send metrics (Enviar métricas).
9. Na lista Extended metrics (Métricas estendidas), escolha Create alarm (Criar alarme) na linha que exibe JSErrorCount em Name (Nome) e Chrome em BrowserName.
10. Em Specify metric and conditions (Especificar métrica e condições), confirme se os campos Metric name (Nome da métrica) e BrowserName estão preenchidos com os valores corretos.
11. Em Statistic (Estatística), selecione a estatística que você deseja usar para o alarme. Average (Média) é uma boa escolha para esse tipo de métrica de contagem.
12. Em Period (Período), selecione 5 minutes (5 minutos).
13. Em Condições, faça o seguinte:

    - Escolha Static (Estático).
    - Escolha Greater (Maior) para especificar que o alarme entre no estado ALARM (ALARME) quando o número de erros for maior do que o limite que você está prestes a especificar.

- Em than... (de...), insira o número para o limite do alarme. O alarme entra no estado ALARM (ALARME) quando o número de erros em um período de 5 minutos excede esse número.

14. (Opcional) Por padrão, o alarme entra no estado ALARM (ALARME) assim que o número de erros excede o número limite que você define durante um período de 5 minutos. Opcionalmente, você pode alterar esse valor para que o alarme só entre no estado ALARM (ALARME) se esse número for excedido por um período maior que 5 minutos.

    Para fazer isso, escolha Additional configuration(Configuração adicional) e, em Datapoints to alarm (Pontos de dados para alarme), especifique quantos períodos de 5 minutos precisam ter o número de erros acima do limite para disparar o alarme. Por exemplo, você pode selecionar 2 de 2 para que o alarme só seja disparado quando dois períodos consecutivos de 5 minutos ficarem acima do limite, ou 2 de 3 para que o alarme seja disparado se dois dos três períodos consecutivos de 5 minutos ficarem acima do limite.

    Para obter mais informações sobre esse tipo de avaliação de alarme, consulte Avaliar um alarme.
15. Escolha Próximo.
16. Em Configure actions (Configurar ações), especifique o que deve acontecer quando o alarme entrar em estado de alarme. Para receber uma notificação com o Amazon SNS, faça o seguinte:

    - Escolha Adicionar notificação.
    - Escolha Em alarme.
    - Selecione um tópico existente do SNS ou crie um novo. Se você criar um novo, especifique um nome para o tópico e adicione pelo menos um endereço de e-mail a ele.
17. Escolha Próximo.
18. Insira um nome e uma descrição opcional para o alarme e escolha Next (Avançar).
19. Revise os detalhes e escolha Create alarm (Criar alarme).

Proteção de dados e privacidade de dados com o CloudWatch RUM

Saiba como o modelo de responsabilidade compartilhada da AWS se aplica à proteção e à privacidade de dados no Amazon CloudWatch RUM. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem da AWS. Você é responsável por manter o controle sobre o conteúdo que hospeda nessa infraestrutura. Para obter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre

privacidade de dados. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem The AWS Shared Responsibility Model and GDPR no blog de segurança da AWS. Para obter mais recursos sobre o cumprimento dos requisitos do GDPR, consulte o Centro Geral de Regulamentação de Proteção de Dados (GDPR).

O Amazon CloudWatch RUM gera um snippet de código para ser incorporado em seu site ou código de aplicação Web, com base na entrada dos dados do usuário final que você deseja coletar. O cliente da Web, baixado e configurado pelo snippet de código, usa cookies (ou tecnologias semelhantes) para ajudar a coletar dados do usuário final. O uso de cookies (ou tecnologias semelhantes) está sujeito a regulamentos de privacidade de dados em determinadas jurisdições. Antes de usar o Amazon CloudWatch RUM, recomendamos enfaticamente que você avalie suas obrigações de conformidade de acordo com a lei aplicável, incluindo quaisquer requisitos legais aplicáveis para fornecer avisos de privacidade legalmente adequados e obter quaisquer consentimentos necessários para o uso de cookies e o processamento (incluindo coleta) de dados do usuário final. Para obter mais informações sobre como o cliente da Web usa cookies (ou tecnologias semelhantes) e quais dados do usuário final o cliente da Web coleta, consulte Informações coletadas pelo cliente da Web CloudWatch RUM e Cookies do cliente da Web do CloudWatch RUM (ou tecnologias semelhantes).

É altamente recomendável que você nunca coloque informações de identificação confidenciais de clientes finais, como números de conta, endereços de e-mail ou outras informações pessoais de clientes finais, em campos de formato livre. Todos os dados inseridos no Amazon CloudWatch RUM ou em outros serviços podem ser incluídos em logs de diagnóstico.

Cookies do cliente da Web do CloudWatch RUM (ou tecnologias semelhantes)

O cliente da Web do CloudWatch RUM coleta determinados dados sobre sessões do usuário por padrão. Você pode optar por habilitar os cookies para que o cliente da Web colete um ID de usuário e um ID de sessão que persistam nos carregamento da página. O ID do usuário é gerado aleatoriamente pelo RUM.

Se esses cookies estiverem habilitados, o RUM poderá exibir os seguintes tipos de dados quando você visualizar o painel do RUM desse monitor de aplicações.

- Dados agregados com base em IDs de usuário, como número de usuários exclusivos e o número de usuários diferentes que receberam um erro.
- Dados agregados com base em IDs de sessão, como o número de sessões e o número de sessões que sofreram um erro.

- A jornada do usuário, que é a sequência de páginas que cada sessão de usuário amostrada inclui.

> **Important**
>
> Se você não habilitar esses cookies (ou tecnologias semelhantes), o cliente da Web ainda registra determinadas informações sobre sessões do usuário final, como tipo ou versão do navegador, tipo ou versão do sistema operacional, tipo do dispositivo, e assim por diante. Eles são coletados para fornecer agregados insights específicos de página, como sinais vitais da Web, visualizações de página e páginas que sofreram erros. Para obter mais informações sobre os dados registrados, consulte Informações coletadas pelo cliente da Web CloudWatch RUM.

Informações coletadas pelo cliente da Web CloudWatch RUM

Esta seção documenta o esquema PutRumEvents, que define a estrutura dos dados que você pode coletar de sessões do usuário usando o CloudWatch RUM.

Uma solicitação PutrumEvents envia uma estrutura de dados com os campos a seguir para o CloudWatch RUM.

- O ID deste lote de eventos do RUM
- Detalhes do monitor de aplicações, que incluem o seguinte:
  - ID do monitor de aplicações
  - Versão da aplicação monitorado
- Detalhes do usuário, que incluem o seguinte. Isso é coletado somente se o monitor de aplicações tiver cookies habilitados.
  - Um ID de usuário gerado pelo cliente da Web
  - ID de sessão
- A matriz de RUM eventsneste lote.

Esquema de evento do RUM

A estrutura de cada evento do RUM inclui os campos a seguir.

- O ID do evento

- Um timestamp
- O tipo de evento
- O atendente do usuário
- Metadados
- Detalhes do evento do RUM

Metadados de evento do RUM

Os metadados incluem metadados de página, metadados do atendente do usuário, metadados de geolocalização e metadados de domínio.

Metadados da página

Os metadados da página incluem o seguinte:

- ID da página
- Título da página
- ID da página pai. Isso será coletado somente se o monitor de aplicações tiver cookies habilitados.
- Profundidade da interação: Isso será coletado somente se o monitor de aplicações tiver cookies habilitados.
- Tags de página: é possível adicionar tags aos eventos de página para agrupar as páginas. Para ter mais informações, consulte Usar grupos de páginas.

Metadados do atendente do usuário

Os metadados do atendente do usuário incluem o seguinte:

- Idioma do navegador
- Nome do navegador
- Versão do navegador
- Nome do sistema operacional
- Versão do sistema operacional
- Tipo de dispositivo
- Tipos de plataforma

Metadados de localização geográfica

Os metadados de geolocalização incluem o seguinte:

- Código do país
- Código de subdivisão

Metadados do domínio

Os metadados do domínio incluem o domínio da URL.

Detalhes do evento do RUM

Os detalhes de um evento seguem um dos tipos de esquemas a seguir, dependendo do tipo de evento.

Evento de início da sessão

Esse evento não contém campos. Isso é coletado somente se o monitor de aplicações tiver cookies habilitados.

Esquema de visualização de página

Um evento de Page View (Exibição de página) contém as seguintes propriedades. Você pode desativar a coleção de exibição de página configurando o cliente da Web. Para obter mais informações, consulte a Documentação do cliente da Web do CloudWatch RUM.

Nome	Tipo	Descrição
ID da página	String	Um ID que representa exclusivamente esta página dentro da aplicação. Por padrão, esse é o caminho do URL.
ID da página pai	String	O ID da página em que o usuário estava quando navegou para a página atual. Isso é coletado somente se o monitor de aplicações tiver cookies habilitados.
Profundidade de interação	String	Isso é coletado somente se o monitor de aplicações tiver cookies habilitados.

Esquema de erro do JavaScript

Os eventos de erro JavaScript gerados pelo atendente contêm as propriedades a seguir. O cliente da Web coleta esses eventos somente se você selecionou a coleta de telemetria de erros.

Nome	Tipo	Descrição
Tipo de erro	String	O nome do erro, se existir algum erro. Para obter mais informações, consulte Error.prototype.Name. Alguns navegadores podem não ser compatíveis com tipos de erro.
Mensagem de erro	String	A mensagem do erro. Para obter mais informações, consulte Error.prototype.message. Se o campo de erro não existir, esta é a mensagem do evento de erro. Para obter mais informações, consulte ErrorEvent. As mensagens de erro podem não ser consistentes em navegadores diferentes.
Rastreamento	String	O rastreamento de pilha do erro, se houver, reduzido a 150 caracteres. Para obter mais informações, consulte Error.prototype.Stack. Alguns navegadores podem não suportar rastreamentos de pilha.

Esquema de eventos DOM

Os eventos do modelo de objeto de documento (DOM) gerados pelo atendente contêm as propriedades a seguir. Esses eventos não são coletados por padrão. Eles serão coletados apenas se você ativar a telemetria das interações. Para obter mais informações, consulte a Documentação do cliente da Web do CloudWatch RUM.

Nome	Tipo	Descrição
Evento	String	O tipo de evento DOM, como clicar, rolar ou passar o mouse. Para obter mais informações, consulte Event reference.
Elemento	String	O tipo de elemento DOM
Element ID (ID de elemento)	String	Se o elemento que gerou o evento tiver uma ID, essa propriedade armazenará essa ID. Para obter mais informações, consulteElement.id.
CSSLocator	String	O localizador CSS usado para identificar o elemento DOM.
InteractionId	String	Um ID exclusivo para a interação entre o usuário e a interface do usuário.

Esquema de eventos de navegação

Os eventos de navegação só serão coletados se o monitor de aplicações tiver telemetria de performance ativada.

Os eventos de navegação usam as APIs Navigation timing Level 1 e Navigation timing Level 2. As APIs de nível 2 não são compatíveis com todos os navegadores, portanto, esses campos mais novos são opcionais.

> **Note**
>
> As Métricas de timestamp são baseadas em DomHighrestimestamp. Com APIs de Nível 2, todos os horários são, por padrão, relativos ao `startTime`. No entanto, para as de Nível 1, a métrica `navigationStart` é subtraída das métricas de timestamp para obter valores relativos. Todos os valores de timestamp estão em milissegundos.

Os eventos de navegação contêm as propriedades a seguir.

Nome	Tipo	Descrição	Observações
initiatorType	String	Representa o tipo de recurso que iniciou o evento de performance.	Value: "navigation" Level 1 (Nível 1): "navigation" Level 2 (Nível 2): entryData.initiatorType
navigationType	String	Representa o tipo de navegação. Esse atributo não é obrigatório.	Value: Esse valor deve ser um dos seguintes: • `navigate` é uma navegação que se inicia com a escolha de um link, a inserção de um URL na barra de endereços de um navegador, o envio de formulários ou uma operação de script diferente

Nome	Tipo	Descrição	Observações
			de `reload` ou `back_forward` . • `reload` é uma navegação através da operação de recarga do navegador ou `location.reload()` . • `back_forward` é uma navegação através da operação de passagem do histórico do navegador. • `prerender` é uma navegação iniciada por uma dica de pré-rende

Nome	Tipo	Descrição	Observações
			rização. Para obter mais informaçõ es, consulte Pré-rende rização.
startTime	Número	Indica quando o evento é acionado.	Value: 0 Level 1 (Nível 1): entryData .navigati onStart - entryData .navigati onStart Level 2 (Nível 2): entryData .startTime

Nome	Tipo	Descrição	Observações
unloadEventStart	Número	Indica o momento em que o documento anterior na janela começou a descarregar após o lançamento do evento`unload`.	Value: (Valor). Se não houver documento anterior ou se o documento anterior ou um dos redirecionamentos necessários não for da mesma origem, o valor retornado será 0. Level 1 (Nível 1): `entryData.unloadEventStart > 0 ? entryData.unloadEventStart - entryData.navigationStart : 0` Level 2 (Nível 2): entryData

Nome	Tipo	Descrição	Observações
			.unloadEvent Start

Nome	Tipo	Descrição	Observações
PromptForUnload	Número	O tempo necessário para descarregar o documento. Em outras palavras, o tempo entre `unloadEventStart` e `unloadEventEnd`. `UnloadEventEnd` representa o momento em milissegundos quando o manipulador de eventos de descarga termina.	Value: (Valor). Se não houver documento anterior ou se o documento anterior ou um dos redirecionamentos necessários não for da mesma origem, o valor retornado será 0. Level 1 (Nível 1): entryData.unloadEventEnd - entryData.unloadEventStart Level 2 (Nível 2): entryData.unloadEventEnd - entryData.unloadEventStart

Nome	Tipo	Descrição	Observações
redirectCount	Número	Um número que representa o número de redirecionamentos desde a última navegação sem redirecionamento no contexto de navegação atual. Esse atributo não é obrigatório.	Value: (Valor) Se não houver redirecionamento ou se houver algum redirecionamento que não seja da mesma origem que o documento de destino, o valor retornado será 0. Level 1 (Nível 1): não disponível Level 2 (Nível 2): entryData.redirectCount

Nome	Tipo	Descrição	Observações
redirectStart	Número	O momento em que o primeiro redirecio namento HTTP é iniciado.	Value: (Valor) Se não houver redirecio namento ou se houver algum redirecio namento que não seja da mesma origem que o documento de destino, o valor retornado será 0. Level 1 (Nível 1): `entryData .redirect Start > 0 ? entryData .redirect Start - entryData .navigati onStart : 0`

Nome	Tipo	Descrição	Observações
			Level 2 (Nível 2): entryData .redirectStart
redirectTime	Número	O tempo necessário para o redirecionamento HTTP. Essa é a diferença entre `redirectStart` e `redirectEnd` .	Level 1 (Nível 1): entryData .redirectEnd - entryData .redirectStart Level 2 (Nível 2): entryData .redirectEnd - entryData .redirectStart

Nome	Tipo	Descrição	Observações
workerStart	Número	Esta é uma propriedade da interface `PerformanceResourceTiming` . Ela marca o início da operação do encadeamento do trabalhador. Esse atributo não é obrigatório.	Value: (Valor) Se um thread do Service Worker já estiver em execução ou imediatamente antes de seu início, essa propriedade retornará a hora imediatamente antes do envio `FetchEvent` . Ela retornará 0 se o recurso não for interceptado por um Service Worker. Level 1 (Nível 1): não disponível Level 2 (Nível 2): entryData.workerStart

Nome	Tipo	Descrição	Observações
workerTime	Número	Se o recurso for interceptado por um Service Worker, isso retornará o tempo necessário para a operação de thread do trabalhador. Esse atributo não é obrigatório.	Level 1 (Nível 1): não disponível Level 2 (Nível 2): `entryData.workerStart > 0 ? entryData.fetchStart - entryData.workerStart : 0`
fetchStart	Número	O horário em que o navegador está pronto para buscar o documento usando uma solicitação HTTP. Isso ocorre antes de verificar qualquer cache de aplicação.	Level 1 (Nível 1): `: entryData.fetchStart > 0 ? entryData.fetchStart - entryData.navigationStart : 0` Level 2 (Nível 2): entryData .fetchStart

Nome	Tipo	Descrição	Observações
domainLookupStart	Número	O momento em que a pesquisa do domínio é iniciada.	Value: (Valor) Se uma conexão persistente for usada ou se as informações forem armazenadas em um cache ou recurso local, o valor será o mesmo que `fetchStart`. Level 1 (Nível 1): `entryData.domainLookupStart > 0 ? entryData.domainLookupStart - entryData.navigationStart : 0` Level 2 (Nível 2): entryData

Nome	Tipo	Descrição	Observações
			.domainLookupStart
DNS	Número	O tempo necessário para a pesquisa de domínio.	Value: (Valor) Se os recursos e os registros DNS forem armazenados em cache, o valor esperado será 0. Level 1 (Nível 1): entryData.domainLookupEnd - entryData.domainLookupStart Level 2 (Nível 2): entryData.domainLookupEnd - entryData.domainLookupStart

Nome	Tipo	Descrição	Observações
nextHopProtocol	String	Uma string que representa o protocolo de rede usado para buscar o recurso. Esse atributo não é obrigatório.	Level 1 (Nível 1): não disponível Level 2 (Nível 2): entryData.nextHopProtocol

Nome	Tipo	Descrição	Observações
connectStart	Número	O momento imediatamente anterior ao atendente do usuário começar a estabelecer a conexão com o servidor para recuperar o documento.	Value (Valor): se uma conexão persistente RFC2616 for usada ou se o documento atual for recuperado de caches de aplicações relevantes ou recursos locais, esse atributo retornará o valor de `domainLookupEnd`. Level 1 (Nível 1): `entryData.connectStart > 0 ? entryData.connectStart - entryData.navigationStart : 0`

Nome	Tipo	Descrição	Observações
			Level 2 (Nível 2): entryData .connectStart
connect	Número	Mede o tempo necessário para estabelecer as conexões de transporte ou para executar a autenticação SSL. Também inclui o tempo bloqueado que é levado quando há muitas solicitações simultâneas emitidas pelo navegador.	Level 1 (Nível 1): entryData .connectEnd - entryData .connectStart Level 2 (Nível 2): entryData .connectEnd - entryData .connectStart
SecureConnectionStart	Número	Se o esquema de URL da página atual for “https”, esse atributo retornará o momento imediatamente anterior ao atendente do usuário iniciar o processo de handshake para proteger a conexão atual. Ele retorna 0 se HTTPS não for usado. Para obter mais informações sobre esquemas de URL, consulteRepresentação de URL.	Formula: entryData .secureConnectionStart

Nome	Tipo	Descrição	Observações
tlsTime	Número	O tempo necessário para completar um Handshake do SSL.	Level 1 (Nível 1): `entryData.secureConnectionStart > 0 ? entryData.connectEnd - entryData.secureConnectionStart : 0` Level 2 (Nível 2): `entryData.secureConnectionStart > 0 ? entryData.connectEnd - entryData.secureConnectionStart : 0`

Nome	Tipo	Descrição	Observações
requestStart	Número	O momento imediatamente anterior ao atendente do usuário começar a solicitar o recurso do servidor ou de caches de aplicações relevantes ou de recursos locais.	Level 1 (Nível 1): `: entryData.requestStart > 0 ? entryData.requestStart - entryData.navigationStart : 0` Level 2 (Nível 2): entryData.requestStart
TimeToFirstByte	Número	O tempo necessário para receber o primeiro byte de informações após a solicitação ser feita. Esse tempo é relativo ao `startTime` .	Level 1 (Nível 1): entryData.responseStart - entryData.requestStart Level 2 (Nível 2): entryData.responseStart - EntryData.requestStart

Nome	Tipo	Descrição	Observações
ResponseStart	Número	O tempo imediatamente posterior ao atendente analisador de HTTP do usuário receber o primeiro byte da resposta dos caches de aplicações relevantes, de recursos locais ou do servidor.	Level 1 (Nível 1): `entryData.responseStart > 0 ? entryData.responseStart - entryData.navigationStart : 0` Level 2: entryData .response Start

Nome	Tipo	Descrição	Observações
ResponseTime	String	O tempo necessário para receber uma resposta completa na forma de bytes dos caches de aplicações relevantes, de recursos locais ou do servidor.	Level 1 (Nível 1): `entryData.responseStart > 0 ? entryData.responseEnd - entryData.responseStart : 0` Level 2 (Nível 2): `entryData.responseStart > 0 ? entryData.responseEnd - entryData.responseStart : 0`

Nome	Tipo	Descrição	Observações
domIntera ctive	Número	O momento em que o analisador terminou seu trabalho no documento principal e o HTML DOM é construído. Neste momento, seu `Document.readyState` se altera para "interativo" e o evento correspondente `readystatechange` lançado.	Level 1 (Nível 1): `entryData .domInter active > 0 ? entryData .domInter active - entryData .navigati onStart : 0` Level 2 (Nível 2): entryData .domInter active

Nome	Tipo	Descrição	Observações
domContentLoadedEventStart	Número	Representa o valor de tempo igual ao tempo imediatamente anterior ao atendente do usuário acionar o evento DOMContentLoaded no documento atual. O evento DOMContentLoaded é acionado quando o documento HTML inicial estiver completamente carregado e analisado. Neste momento, o documento HTML principal terminou a análise, o navegador começa a construir a árvore de renderização e os sub-recursos ainda precisam ser carregados. Ele não espera que folhas de estilo, imagens e subquadros terminem o carregamento.	Level 1 (Nível 1): `entryData.domContentLoadedEventStart > 0 ? entryData.domContentLoadedEventStart - entryData.navigationStart : 0` Level 2 (Nível 2): entryData.domContentLoadedEventStart

Nome	Tipo	Descrição	Observações
domContentLoaded	Número	Esse horário de início e término da construção da árvore de renderização é marcado pelos `domContentLoadedEventStart` e `domContentLoadedEventEnd`. Ele permite que o CloudWatch RUM rastreie a execução. Essa propriedade é a diferença entre `domContentLoadedStart` e `domContentLoadedEnd`. Durante esse período, o DOM e o CSSOM estão prontos. Essa propriedade aguarda a execução do script, exceto scripts assíncronos e criados dinamicamente. Se os scripts dependem de folhas de estilo, o `domContentLoaded` aguarda nas folhas de estilo, também. Ele não aguarda nas imagens. **Note** Os valores reais de `domContentLoadedStart` e `domContentLoadedEnd` se aproximam de `domContentLoaded` no painel Rede do Google Chrome. Ele indica o tempo de construção da árvore de renderização HTML DOM + CSSOM desde o início do processo de carregamento da página. No caso das métricas de navegação, o valor de `domContentLoaded` representa a diferença entre os valores inicial e final, que é o tempo necessário somente para baixar sub-recursos e construção de árvore de renderização.	Level 2 (Nível 2): entryData.domContentLoadedEventEnd - entryData.domContentLoadedEventStart Level 2 (Nível 2): entryData.domContentLoadedEventEnd - entryData.domContentLoadedEventStart

Nome	Tipo	Descrição	Observações
domComplete	Número	O momento imediatamente anterior ao navegador definir a prontidão do documento atual para a sua conclusão. Nesse ponto, o carregamento de sub-recursos, como imagens, está completo. Isso inclui o tempo necessário para baixar conteúdo de bloqueio, como CSS e JavaScript síncrono. Isso se aproxima de `loadTime` no painel Rede do Google Chrome.	Level 1 (Nível 1): `entryData.domComplete > 0 ? entryData.domComplete - entryData.navigationStart : 0` Level 2 (Nível 2): entryData.domComplete
domProcessingTime	Número	O tempo total entre a resposta e o início do evento de carga.	Level 1 (Nível 1): entryData.loadEventStart - entryData.responseEnd Level 2 (Nível 2): entryData.loadEventStart - entryData.responseEnd

Nome	Tipo	Descrição	Observações
loadEvent Start	Número	O tempo imediatamente anterior ao disparo do evento `load` do documento atual.	Level 1 (Nível 1): `entryData .loadEven tStart > 0 ? entryData .loadEven tStart - entryData .navigati onStart : 0` Level 2 (Nível 2): entryData .loadEven tStart
loadEvent Time	Número	A diferença entre `loadEventStart` e `loadEventEnd` . Funções adicionais ou lógicas aguardando esse evento de carregamento serão disparadas durante esse período.	Level 1 (Nível 1): entryData .loadEven tEnd - entryData .loadEven tStart Level 2 (Nível 2): entryData .loadEven tEnd - entryData .loadEven tStart

Nome	Tipo	Descrição	Observações
duration	String	A duração é o tempo total de carregamento da página. Ela registra o tempo para baixar a página principal e todos os seus sub-recursos síncronos, e também para renderizar a página. Recursos assíncronos, como scripts, continuam sendo baixados posteriormente. Essa é a diferença entre as propriedades `loadEventEnd` e `startTime` propriedades.	Level 1 (Nível 1): entryData.loadEventEnd - entryData.navigationStart Level 2 (Nível 2): entryData.duration
headerSize	Número	Devolve a diferença entre `transferSize` e `encodedBodySize` . Esse atributo não é obrigatório.	Level 1 (Nível 1): não disponível Level 2 (Nível 2): entryData.transferSize - entryData.encodedBodySize Level 2 (Nível 2): entryData.transferSize - entryData.encodedBodySize

Nome	Tipo	Descrição	Observações
compressionRatio	Número	A proporção de `encodedBodySize` e `decodedBodySize`. O valor de `encodedBodySize` é o tamanho compactado do recurso, excluindo os cabeçalhos HTTP. O valor de `decodedBodySize` é o tamanho descompactado do recurso, excluindo os cabeçalhos HTTP. Esse atributo não é obrigatório.	Level 1 (Nível 1): não disponível Level 2 (Nível 2): `entryData.encodedBodySize > 0 ? entryData.decodedBodySize / entryData.encodedBodySize : 0`
navigationTimingLevel	Número	A versão da API de tempo de navegação.	Value: (Valor) 1 ou 2

Esquema de eventos de recurso

Os eventos de recurso são coletados somente se o monitor de aplicações tiver a telemetria de performance ativada.

Métricas de timestamp são baseadas em The DOMHighResTimeStamp typedef. Com APIs de Nível 2, por padrão, todos os horários são relativos ao `startTime`. Mas para APIs de Nível 1, a métrica `navigationStart` é subtraída das métricas de timestamp para obter valores relativos. Todos os valores de timestamp estão em milissegundos.

Os eventos de recursos gerados pelo atendente contêm as propriedades a seguir.

Nome	Tipo	Descrição	Observações
TargetUrl	String	Retorna a URL do recurso.	Fórmula: entryData .name
InitiatorType	String	Representa o tipo de recurso que iniciou o evento de recurso de performance.	Value: "resource" Fórmula: entryData .InitiatorType
duration	String	Retorna a diferença entre as propriedades `responseEnd` e `startTime` . Esse atributo não é obrigatório.	Formula: entryData .duration
transferSize	Número	Retorna o tamanho (em octetos) do recurso buscado, incluindo os campos do cabeçalho de resposta e o corpo da carga útil da resposta. Esse atributo não é obrigatório.	Formula: entryData .transferSize
fileType	String	Extensões derivadas do padrão de URL de destino.	

Esquema de evento largest contentful paint (maior exibição de conteúdos)

Os eventos de largest contentful paint event contêm as propriedades a seguir.

Esses eventos só são coletados se a telemetria de performance do monitor de aplicações estiver ativada.

Nome	Descrição		
Valor	Para obter mais informações,		

Nome	Descrição		
	consulte Web Vitals.		

Evento de first input delay (atraso da primeira entrada)

Os primeiros eventos first input delay contêm as seguintes propriedades.

Esses eventos só são coletados se a telemetria de performance do monitor de aplicações estiver ativada.

Nome	Descrição		
Valor	Para obter mais informações, consulte Web Vitals.		

Evento de cumulative layout shift (deslocamento cumulativo de layout)

Os eventos de cumulative layout shift contêm as propriedades a seguir.

Esses eventos só são coletados se a telemetria de performance do monitor de aplicações estiver ativada.

Nome	Descrição		
Valor	Para obter mais informações, consulte Web Vitals.		

Evento HTTP

Os eventos HTTP podem conter as propriedades a seguir. Ele conterá um campo `Response` ou um campo `Error`, mas não ambos.

Esses eventos só são coletados se a telemetria HTTP do monitor de aplicações estiver ativada.

Nome	Descrição
Solicitação	O campo de solicitação inclui o seguinte: • O campo `Method`, que pode ter valores como GET, POST e assim por diante. • O URL
Resposta	O campo de resposta inclui o seguinte: • Status, como 2xx, 4xx ou 5xx • Texto de status
Erro	O campo de erros inclui o seguinte: • Tipo • Message • Nome do arquivo • Line number • Número da coluna • Rastreamento

Esquema de eventos de rastreamento do X-Ray

Esses eventos são coletados somente se o monitor de aplicações tiver o rastreamento de X-Ray ativado.

Para obter informações sobre esquemas de eventos de rastreamento de X-Ray, consulte Documentos de segmento do AWS X-Ray.

Tempo de mudança de rota para aplicações de página única

Em uma aplicação tradicional de várias páginas, ao solicitar o carregamento de um novo conteúdo, o usuário está, na verdade, solicitando uma nova página HTML do servidor. Por isso, o cliente Web do CloudWatch RUM registra os tempos de carregamento usando as métricas regulares da API de performance.

No entanto, aplicações Web de página única usam JavaScript e Ajax para atualizar a interface sem carregar uma nova página do servidor. As atualizações de página única não são registradas pela API de tempo do navegador e, em vez disso, usam o tempo de mudança de rota.

O CloudWatch RUM é compatível tanto com o monitoramento de carregamentos de páginas inteiras do servidor como de atualizações de página única, com as seguintes diferenças:

- No tempo de mudança de rota, não há métricas fornecidas pelo navegador, como `tlsTime`, `timeToFirstByte` e outras.
- No tempo de mudança de rota, o campo `initiatorType` será `route_change`.

O cliente Web do CloudWatch RUM escuta as interações do usuário que podem levar a uma mudança de rota e, quando essa interação do usuário é registrada, o cliente Web registra um carimbo de data/hora. Em seguida, o tempo de mudança de rota começará a ser contado se estas duas condições forem verdadeiras:

- Uma API de histórico do navegador (exceto os botões avançar e voltar do navegador) foi usada para realizar a mudança de rota.
- A diferença entre o tempo de detecção da mudança de rota e o carimbo de data/hora da interação mais recente do usuário é inferior a 1.000 ms. Isso evita a distorção de dados.

Após ser iniciado, o tempo de mudança de rota será concluído se não houver solicitações AJAX e mutações DOM em andamento. Em seguida, o carimbo de data/hora da última atividade concluída será usado como carimbo de data/hora de conclusão.

O tempo da mudança de rota expirará se houver solicitações AJAX ou mutações DOM em andamento por mais de 10 segundos (por padrão). Nesse caso, o cliente Web do CloudWatch RUM não registrará mais o tempo dessa mudança de rota.

Como resultado, a duração de um evento de mudança de rota é calculada da seguinte forma:

```
(time of latest completed activity) - (latest user interaction timestamp)
```

Gerenciar suas aplicações que usam o CloudWatch RUM

Siga as etapas nessas seções para gerenciar o uso do CloudWatch RUM pelas aplicações.

Como encontro um snippet de código que já gerei?

Para encontrar um snippet de código RUM do CloudWatch que você já gerou para uma aplicação, siga estas etapas.

Para encontrar um snippet de código que você já gerou

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, RUM.
3. Escolha List view (Visualização em lista).
4. Ao lado do nome do monitor de aplicações, escolha View JavaScript (Visualizar JavaScript).
5. No painel do Snippet JavaScript, escolha Copy to clipboard. (Copiar para a área de transferência).

Editar sua aplicação

Para alterar as configurações de um monitor de aplicações, siga estas etapas. Você pode alterar todas as configurações, exceto o nome do monitor de aplicações.

Para editar como sua aplicação usa o CloudWatch RUM

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, RUM.
3. Escolha List view (Visualização em lista).
4. Escolha o botão ao lado do nome da aplicação e escolha Actions (Ações), Edit (Editar).
5. Altere todas as configurações, exceto o nome da aplicação. Para obter mais informações sobre as configurações, consulte Etapa 2: Criar um monitor de aplicações.
6. Quando terminar, escolha Save (Salvar).

   Alterar as configurações altera o snippet de código. Agora você deve colar o snippet de código atualizado em sua aplicação.
7. Depois que o snippet de código JavaScript for criado, escolha Copy to clipboard (Copiar para a área de transferência) ou Download (Baixar) e depois escolha Done (Concluído).

   Para iniciar o monitoramento com as novas configurações, insira o snippet de código em sua aplicação. Insira o snippet de código dentro do elemento `<head>` da aplicação, antes do elemento `<body>` ou qualquer outra etiqueta `<script>`.

Pare de usar o CloudWatch RUM ou exclua um monitor de aplicações

Para parar de usar o CloudWatch RUM com uma aplicação, remova o snippet de código gerado pelo RUM do código da aplicação.

Para excluir um monitor de aplicações do RUM, siga estas etapas.

Para excluir um monitor de aplicações

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Application Signals e, em seguida, RUM.
3. Escolha List View (Visualização em lista).
4. Escolha o botão ao lado do nome da aplicação e escolha Actions (Ações), Delete (Excluir).
5. Na caixa de confirmação, insira **`Delete`** e escolha Delete (Excluir).
6. Se você ainda não tiver feito isso, exclua o snippet de código RUM do CloudWatch do código da aplicação.

Cotas do RUM do CloudWatch

O CloudWatch RUM tem as cotas a seguir.

Recurso	Cota padrão
Monitores de aplicações	20 por conta É possível solicitar um aumento da cota.
Taxa de ingestão do RUM	50 solicitações PutRumEvents por segundo (TPS). É possível solicitar um aumento da cota.

Solucionar problemas do CloudWatch RUM

Esta seção contém dicas para ajudar na solução de problemas do CloudWatch RUM.

Não há dados da minha aplicação

Primeiro, verifique se o snippet de código foi inserido corretamente em sua aplicação. Para ter mais informações, consulte Etapa 4: Inserir o snippet de código na sua aplicação.

Se esse não for o problema, talvez ainda não tenha havido tráfego para sua aplicação. Gere algum tráfego acessando sua aplicação da mesma forma que um usuário faria.

Os dados deixaram de ser gravados para a minha aplicação

Sua aplicação pode ter sido atualizada e agora não contém mais um snippet de código RUM do CloudWatch. Confira o código da sua aplicação.

Também há a possibilidade de que alguém tenha atualizado o snippet de código, mas não o tenha inserido na aplicação. Encontre o snippet de código correto atual seguindo as instruções em Como encontro um snippet de código que já gerei? e compare-o com o snippet de código colado em sua aplicação.

Monitoramento de rede

Os tópicos desta seção descrevem os recursos de monitoramento de rede e Internet do CloudWatch fornecidos pelo Monitor de Internet do Amazon CloudWatch e pelo Amazon CloudWatch Network Monitor. Esses serviços ajudam você a obter visibilidade operacional do desempenho e da disponibilidade de rede e Internet das aplicações hospedadas na AWS.

- O Monitor de Internet usa os dados de conectividade que a AWS captura de sua rede global para calcular uma linha de base de performance e de disponibilidade para tráfego voltado para a Internet. Você pode ter uma visão geral dos padrões de tráfego e eventos de integridade, além de detalhar facilmente as informações sobre os eventos. Você também pode receber alertas sobre eventos de integridade da Internet que afetam os clientes das aplicações. Além disso, você pode usar os insights fornecidos pelo Monitor de Internet para explorar possíveis melhorias na experiência do cliente, usando o Amazon CloudFront ou roteando em outras Regiões da AWS.
- O Network Monitor usa uma abordagem de agente totalmente gerenciada para permitir que você rastreie e visualize a latência e a perda de pacotes para conexões de rede híbrida. Para coletar medições e permitir que o Network Monitor crie alertas de eventos de integridade para a aplicação, crie testes que são enviados dos seus recursos hospedados na AWS para endereços IP de destino on-premises. Não é necessário instalar agentes adicionais para monitorar a performance da rede. Assim como no Monitor de Internet, é possível definir alertas e limites, obter informações que ajudam a solucionar problemas rapidamente e tomar medidas para melhorar a experiência do usuário final.

Tópicos

Uso do Monitor de Internet do Amazon CloudWatch

O Monitor de Internet do Amazon CloudWatch fornece visibilidade de como os problemas de Internet afetam a performance e a disponibilidade entre suas aplicações hospedadas na AWS e seus usuários finais. Isso pode reduzir o tempo necessário para diagnosticar problemas na Internet de dias para minutos. O Monitor de Internet usa os dados de conectividade que a AWS captura de sua rede global para calcular uma linha de base de performance e de disponibilidade para tráfego voltado para a Internet. Esses são os mesmos dados que a AWS usa para monitorar o tempo de atividade

e a disponibilidade de Internet. Com essas medições como base, o Monitor de Internet chama a sua atenção quando há problemas significativos para seus usuários finais (clientes) nas diferentes localizações geográficas em que a sua aplicação for executada.

No console do Amazon CloudWatch, é possível ter uma visão global dos padrões de tráfego e dos eventos de integridade, e detalhar facilmente as informações sobre os eventos em diferentes granularidades geográficas (locais). É possível visualizar claramente o impacto e identificar os locais e redes dos clientes (ASNs, geralmente provedores de serviços de Internet ou ISPs) afetados. Se o Monitor de Internet determinar que um problema de disponibilidade ou de performance da Internet é causado por um ASN específico ou pela rede da AWS, ele fornecerá essas informações.

Principais recursos do Monitor de Internet

- O Monitor de Internet sugere ideias e recomendações que podem ajudar você a melhorar a experiência dos usuários finais. É possível explorar, quase em tempo real, como aprimorar a latência prevista da sua aplicação alternando para o uso de outros serviços ou redirecionando o tráfego para sua workload por diferentes Regiões da AWS.
- Com o Monitor de Internet, você pode identificar rapidamente o que está afetando a performance e a disponibilidade da aplicação, para poder rastrear e resolver os problemas.
- O Monitor de Internet publica medições da Internet para o CloudWatch Logs e o CloudWatch, para oferecer suporte ao uso de ferramentas do CloudWatch com informações de integridade para locais e ASNs (provedores de serviços de Internet) específicos da sua aplicação. Opcionalmente, também é possível publicar medições de Internet no Amazon S3.
- O Monitor de Internet envia os eventos de integridade para o Amazon EventBridge para que você possa configurar as notificações. Se um problema for causado pela rede da AWS, você também receberá automaticamente uma notificação do AWS Health Dashboard informando com as etapas que a AWS está tomando para resolver o problema.

Como utilizar o Monitor de Internet

Para usar o Monitor de Internet, você cria um monitor e associa os recursos da sua aplicação a ele (VPCs, Network Load Balancers, distribuições do CloudFront ou diretórios do WorkSpaces), para permitir que o Monitor de Internet saiba onde está o tráfego voltado para a Internet da sua aplicação. Em seguida, o Monitor de Internet publica medições da Internet de AWS específicas para as cidades-redes, ou seja, as localizações dos clientes e os ASNs (geralmente provedores de serviços de Internet ou ISPs), onde os clientes acessam sua aplicação. Para ter mais informações, consulte

Como o Monitor de Internet do Amazon CloudWatch funciona. Para iniciar o trabalho com o Monitor de Internet, consulte Introdução ao Monitor de Internet do Amazon CloudWatch usando o console.

Conteúdo

Regiões da AWS com suporte para o Monitor de Internet do Amazon CloudWatch

As Regiões da AWS nas quais o Monitor de Internet do Amazon CloudWatch tem suporte estão descritas nesta seção. Para obter a lista atualizada das regiões com suporte para o Monitor de Internet, incluindo as regiões de adesão, consulte Endpoints e cotas do Monitor de Internet do Amazon CloudWatch na Referência geral da Amazon Web Services.

Observe que o Monitor de Internet armazena dados de um monitor somente na Região da AWS em que você cria o monitor, embora um monitor possa incluir recursos em várias regiões.

Nome da região (compatibilidade opcional)	Região
Africa (Cape Town)	af-south-1
Ásia-Pacífico (Hong Kong)	ap-east-1
Ásia-Pacífico (Hyderabad)	ap-south-2
Ásia-Pacífico (Jacarta)	ap-southeast-3
Ásia-Pacífico (Melbourne)	ap-southeast-4
Europa (Milão)	eu-south-1
Europa (Espanha)	eu-south-2
Europa (Zurique)	eu-central-2
Oriente Médio (Barém)	me-south-1
Oriente Médio (Emirados Árabes Unidos)	me-central-1

Nome da região (compatibilidade padrão)	Região
Leste dos EUA (Ohio)	us-east-2
Leste dos EUA (N. da Virgínia)	us-east-1
Oeste dos EUA (N. da Califórnia)	us-west-1
Oeste dos EUA (Oregon)	us-west-2
Ásia-Pacífico (Mumbai)	ap-south-1
Ásia-Pacífico (Osaka)	ap-northeast-3
Ásia-Pacífico (Seul)	ap-northeast-2
Ásia-Pacífico (Singapura)	ap-southeast-1

Nome da região (compatibilidade padrão)	Região
Ásia-Pacífico (Sydney)	`ap-southeast-2`
Ásia-Pacífico (Tóquio)	`ap-northeast-1`
Canadá (Central)	`ca-central-1`
Europa (Frankfurt)	`eu-central-1`
Europa (Irlanda)	`eu-west-1`
Europa (Londres)	`eu-west-2`
Europa (Paris)	`eu-west-3`
Europa (Estocolmo)	`eu-north-1`
América do Sul (São Paulo)	`sa-east-1`

Informações de preços do Monitor de Internet do Amazon CloudWatch.

Com o Monitor de Internet do Amazon CloudWatch, não há custos iniciais nem compromissos de longo prazo. O preço do Monitor de Internet tem dois componentes: uma taxa por recurso monitorado e uma taxa por cidade-rede. Uma cidade-rede é o local de onde os clientes acessam os recursos da sua aplicação e a rede (ASN, como um provedor de serviços de Internet ou ISP) pela qual os clientes acessam os recursos. Além disso, observe que serão cobrados os preços padrão do CloudWatch para os logs e para quaisquer métricas, painéis, alarmes ou insights adicionais que você criar.

Você escolhe um percentual do tráfego para monitorar ao criar um monitor. Para ajudar a controlar sua fatura, também é possível definir um limite para o número máximo de cidades-redes a serem monitoradas. É possível atualizar o percentual de tráfego a ser monitorado ou o limite máximo de cidades-redes a qualquer momento editando seu monitor. As primeiras 100 cidades-redes (em todos os monitores por conta) estão incluídas. Depois disso, você paga apenas pelo número adicional real de cidades-redes que monitorar, até o número máximo.

Você paga somente o número adicional real de cidades-redes que monitorar, até o número máximo, sem cobrança pelas primeiras 100 cidades-redes (em todos os monitores por conta). Um valor fixo equivalente ao custo de 100 cidades-redes é deduzido da sua fatura mensal.

Por exemplo, uma grande empresa global poderia optar por monitorar 100% de seu tráfego voltado para a Internet e definir um máximo de 50.000 cidades-redes para um monitor com um recurso. Supondo que o tráfego atingisse 50.000 cidades-redes, essa parte de sua fatura seria de cerca de 2.700 USD/mês. Para outra empresa, em menos áreas geográficas, com 1 monitor com 1 recurso e 200 redes municipais, essa parte da conta seria de aproximadamente USD 13/mês. Para ter mais informações, consulte Escolha de um limite máximo de cidades-redes.

É possível experimentar diferentes opções com a calculadora de preços. Para explorar as opções de preços, na página Calculadora de preços do CloudWatch, role para baixo até o Monitor de Internet.

Para obter mais informações sobre os preços do Monitor de Internet e do CloudWatch, consulte a página Definição de preço do Amazon CloudWatch.

Componentes e termos do Monitor de Internet do Amazon CloudWatch

O Monitor de Internet do Amazon CloudWatch usa ou referencia os componentes a seguir.

Monitor

Um monitor inclui os recursos de uma única aplicação para o qual você deseja visualizar as medidas de performance e disponibilidade da Internet e sobre o qual deseja receber alertas de eventos de integridade. Ao criar um monitor para uma aplicação, você adiciona recursos para que a aplicação defina as cidades (locais) para o Monitor de Internet monitorar. O Monitor de Internet usa os padrões de tráfego dos recursos da aplicação que você adiciona para que possa publicar medições de performance e de disponibilidade de Internet específicas para apenas as localizações e os ASNs (geralmente, os provedores de serviços de Internet ou os ISPs) que se comunicam com a aplicação. Em outras palavras, os recursos que você adiciona criam um escopo das cidades-redes que você deseja que o Monitor de Internet monitore e para as quais você deseja publicar medições.

Recurso adicionado ao monitor (“recurso monitorado”)

Um recurso que você adiciona a um monitor é um “recurso monitorado” no Monitor de Internet. Ou seja:

- Cada VPC que você adiciona a uma região é um recurso monitorado. Quando você adiciona uma VPC, o Monitor de Internet realiza o monitoramento do tráfego para qualquer aplicação voltada para a Internet na VPC, por exemplo, uma aplicação hospedada em uma instância do Amazon EC2, protegida por um Network Load Balancer, ou em um contêiner do AWS Fargate.
- Cada Network Load Balancer que você adiciona em uma região é um recurso monitorado.

- Cada diretório do WorkSpaces que você adiciona a uma região é um recurso monitorado.
- Cada distribuição do CloudFront que você adiciona é um recurso monitorado.

Número de sistema autônomo (ASN)

No Monitor de Internet, um ASN normalmente se refere a um provedor de serviços de Internet (ISP), como a Verizon ou a Comcast. Um ASN é um provedor de rede que um cliente usa para acessar sua aplicação de Internet. Um sistema autônomo (AS) é um conjunto de prefixos de protocolo de Internet (IP) rotativos que pertencem a uma rede ou a uma coleção de redes que são todas gerenciadas, controladas e supervisionadas por uma organização.

Cidade-rede (localização e ASN)

Uma cidade-rede corresponde à localidade (por exemplo, uma cidade) da qual os clientes acessam os recursos da aplicação e ao ASN, que normalmente é um provedor de serviços de Internet (ISP), por meio do qual os clientes acessam os recursos. Para ajudar a controlar sua fatura, é possível definir um limite para o número máximo de redes de cidades que o Monitor de Internet monitorará para cada monitor. Você paga apenas pelo número real de cidades-redes que monitorar, até o número máximo. Para obter mais informações, consulte Escolha de um limite máximo de cidades-redes.

Medições da Internet

O Monitor de Internet publica medições da Internet em arquivos de log no CloudWatch Logs a cada cinco minutos para as 500 principais cidades-redes (locais de clientes e ASNs, geralmente provedores de serviços de Internet ou ISPs) em sua conta. Essas medidas quantificam a pontuação de performance, a pontuação de disponibilidade, os bytes transferidos (bytes de entrada e saída) e o tempo de ida e volta das cidades-redes da sua aplicação. Essas são medidas para as cidades-redes específicas de suas VPCs, Network Load Balancers, distribuições do CloudFront ou diretórios do WorkSpaces. Opcionalmente, é possível optar por publicar medições e eventos da Internet para todas as cidades-redes monitoradas (até o limite de serviço de 500.000 cidades-redes) em um bucket do Amazon S3.

Metrics

O Monitor de Internet gera métricas agregadas de métricas do CloudWatch para o tráfego global para a aplicação e o tráfego global para cada Região da AWS. Para ter mais informações, consulte Usar o CloudWatch Logs Metrics com o Monitor de Internet do Amazon CloudWatch.

Evento de integridade

O Monitor de Internet cria um evento de integridade para alertá-lo sobre um problema específico que afete sua aplicação. O Monitor de Internet detecta problemas de Internet, como aumento da

latência da rede, em todo o mundo. Depois, ele usa as medições históricas de Internet em toda a infraestrutura global da AWS para calcular o impacto dos problemas atuais na aplicação e, então, cria eventos de integridade. O Monitor de Internet, por padrão, cria eventos de integridade com base em ambos os limites de impacto geral e impacto local. Para obter mais informações sobre como configurar limites, consulte Alterar limites de eventos de integridade.

Cada evento de integridade inclui informações sobre as cidades-redes afetadas. É possível visualizar os eventos de integridade no console do CloudWatch, usando o AWS SDK ou a AWS CLI com as ações de API do Monitor de Internet. O Monitor de Internet também envia notificações do Amazon EventBridge para eventos de integridade. Para obter mais informações, consulte Quando o Monitor de Internet cria e resolve eventos de integridade.

Evento da Internet

O Monitor de Internet exibe informações sobre eventos de integridade globais mais recentes, denominados eventos da Internet, em um mapa de condições da Internet que está disponível para todos os clientes da AWS. Não é necessário criar um monitoramento no Monitor de Internet para visualizar o mapa de condições da Internet. Ao contrário dos eventos de integridade, os eventos da Internet não são específicos para clientes individuais ou para o tráfego de suas aplicações. Para ter mais informações, consulte Mapa de condições globais da Internet no Monitor de Internet do Amazon CloudWatch.

Limites

O Monitor de Internet cria eventos de integridade baseado em ambos os limites gerais e locais. É possível alterar os limites padrão e configurar outras opções, como desativar os limites locais. Para obter mais informações sobre como configurar limites, consulte Alterar limites de eventos de integridade.

Pontuações de performance e disponibilidade

Ao analisar os dados coletados pela AWS, o Monitor de Internet pode detectar quando a performance e a disponibilidade da aplicação diminuíram em comparação com as linhas de base estimadas que o Monitor de Internet calcula. Para facilitar a visualização dessas quedas, o Monitor de Internet relata as informações a você sob a forma de pontuações. Uma pontuação de performance representa a porcentagem estimada de tráfego que não está apresentando queda de performance. Da mesma forma, uma pontuação de disponibilidade representa a porcentagem estimada de tráfego que não está apresentando queda de disponibilidade. Para obter mais informações, consulte Como a AWS calcula pontuações de performance e disponibilidade.

Bytes transferidos e bytes transferidos monitorados

Bytes transferidos é o número total de bytes de tráfego de entrada e saída entre uma aplicação na AWS e a cidade-rede (ou seja, o local e o ASN, geralmente o provedor de serviços de Internet) em que os clientes acessam uma aplicação. Os bytes monitorados transferidos são uma métrica semelhante, mas incluem somente os bytes para o tráfego monitorado.

Tempo de ida e volta

O tempo de ida e volta (RTT) é o tempo necessário para que uma solicitação de um usuário cliente retorne uma resposta. Quando o RTT é agregado em todos os locais do cliente (cidades ou outras regiões geográficas), o valor é ponderado pela quantidade de tráfego da sua aplicação direcionado por cada local do cliente.

Mapa de condições globais da Internet no Monitor de Internet do Amazon CloudWatch

O Monitor de Internet do Amazon CloudWatch exibe um mapa de condições globais da Internet que está disponível para todos os clientes da AWS. Para visualizar o mapa, no console do Amazon CloudWatch, navegue até o Monitor de Internet.

O mapa destaca eventos da Internet (“interrupções”) que ocorrem em todo o mundo e afetam os clientes da AWS, com cidades e redes específicas (ASNs, que normalmente são provedores de serviços de Internet) em que existem problemas de performance ou de disponibilidade. O mapa de condições da Internet inclui eventos da Internet que ocorreram nas últimas 24 horas.

Não é necessário criar um monitoramento no Monitor de Internet para visualizar o mapa de condições da Internet. Ao contrário dos eventos de integridade no Monitor de Internet, os eventos da Internet não são específicos para clientes individuais ou para o tráfego de suas aplicações.

No mapa de condições da Internet, é possível escolher um evento da Internet para obter mais detalhes sobre ele. Em relação a um evento da Internet, é possível visualizar o horário de início, o horário de término (se o evento tiver terminado), o status atual (Ativo ou Resolvido) e o tipo de problema de interrupção que ocorreu (Disponibilidade ou Performance). Para saber mais sobre como o mapa de condições da Internet foi criado e o que está incluso nele, consulte as perguntas frequentes sobre o mapa de condições globais da Internet.

Para visualizar e trabalhar com informações detalhadas que são específicas para o tráfego e para as localidades de clientes da sua aplicação, é possível configurar com facilidade um monitoramento

no Monitor de Internet para a aplicação. Nesse caso, você visualizará padrões e eventos de performance e de disponibilidade, sejam eles atuais ou históricos, bem como receberá alertas de eventos de integridade, adaptados apenas à área de abrangência e aos clientes da sua aplicação. O mapa de condições da Internet disponibiliza uma visão geral, enquanto um monitoramento específico filtra as informações somente para as medidas e os detalhes relevantes para a sua aplicação. Com um monitoramento, também é possível explorar métricas históricas e obter recomendações para melhorar a experiência do cliente em sua aplicação. Para saber mais, consulte Introdução ao Monitor de Internet do Amazon CloudWatch usando o console.

Como o Monitor de Internet do Amazon CloudWatch funciona

Esta seção fornece informações sobre como o Monitor de Internet do Amazon CloudWatch funciona. Isso inclui descrições sobre como a AWS coleta os dados que usa para ajudar a detectar problemas de conectividade em toda a Internet e como as pontuações de performance e disponibilidade são calculadas.

Índice

Como o Monitor de Internet se concentra somente na abrangência de tráfego da sua aplicação

O Monitor de Internet concentra o monitoramento apenas no subconjunto da Internet que é acessado pelos usuários dos recursos da AWS, em vez de monitorar de modo amplo seu site em todas as regiões do mundo, como fazem outras ferramentas. Também é uma solução econômica, acessível para grandes e pequenas empresas.

O Monitor de Internet usa as mesmas sondas e algoritmos poderosos de detecção de problemas que a AWS usa internamente e alerta você sobre problemas de conectividade que afetam a

aplicação, criando eventos de integridade no Monitor de Internet. Depois, o Monitor de Internet fornece acesso ao mapa de performance e de disponibilidade resultante, sobrepondo o perfil de tráfego que ele cria a partir de seus visualizadores ativos, com base nos recursos da aplicação.

Usando essas informações, o Monitor de Internet mostra apenas os eventos relevantes (ou seja, os eventos de lugares onde você tem visualizadores ativos) e apenas o impacto que esses eventos têm no seu volume total de visualizadores. Portanto, o impacto de um evento, em termos percentuais, é baseado no seu tráfego total no mundo todo.

O Monitor de Internet publica no CloudWatch Logs medições da Internet a cada cinco minutos para as 500 principais cidades-redes (locais de clientes e ASNs, geralmente provedores de serviços de Internet ou ISPs) que enviem tráfego para cada monitor. Opcionalmente, é possível optar por publicar medições da Internet para todas as cidades-redes monitoradas (até o limite de serviço de 500.000 cidades-redes) em um bucket do Amazon S3. Para ter mais informações, consulte Publicação de medições da Internet para o Amazon S3 no Monitor de Internet do Amazon CloudWatch.

Os benefícios do Monitor de Internet incluem os seguintes:

- Usar o Monitor de Internet não impõe carga ou custo adicional à aplicação hospedada na AWS.
- Você não precisa incluir código de medição de performance nos recursos do lado do cliente nem na sua aplicação.
- Você pode ter visibilidade da performance e da disponibilidade em toda a Internet à qual a aplicação está conectada, incluindo as informações de "última milha".

Observe que, como o Monitor de Internet cria medições com base nos seus recursos da AWS, o Monitor de Internet só cria eventos específicos do tráfego da aplicação. Os problemas da Internet global em geral não são relatados. Além disso, quando o local do serviço é uma Região da AWS, as medições e os eventos emitidos pretendem representar a conectividade em um nível regional e não representam com precisão a conectividade entre um local do usuário final e uma zona de disponibilidade.

Como a AWS mede os problemas de conectividade e calcula as medições

O Monitor de Internet do Amazon CloudWatch usa dados de conectividade da Internet entre diferentes Regiões da AWS e pontos de presença (POPs) do Amazon CloudFront para diversas localidades de clientes por meio de números de sistema autônomo (ASNs), que geralmente são provedores de serviços de Internet (ISPs). Esses são os dados de conectividade usados internamente pelos operadores da AWS, diariamente, para detectar proativamente problemas de conectividade na Internet global.

Para cada Região da AWS, sabemos quais seções da Internet se comunicam com a região e fazemos o seguinte:

- Monitoramos ativamente essas seções da Internet com uma janela móvel de 30 dias.
- Usamos sondas de rede e de protocolos de nível superior, incluindo sondagem de entrada e saída.

A AWS tem sondas ativas e passivas que medem a latência (performance) no 90º percentil e a acessibilidade (disponibilidade) de todas as Região da AWS e do serviço CloudFront para toda a Internet. Os padrões anormais na conectividade entre um serviço e a localidade de um cliente são monitorados e, em seguida, relatados como alertas ao cliente.

Cálculo de disponibilidade e de RTT

O tempo de ida e volta (RTT) é o tempo necessário para que uma solicitação do usuário retorne uma resposta. Quando o tempo de ida e volta é agregado entre os locais de usuários finais, o valor é ponderado pela quantidade de tráfego que é direcionada por cada local de usuário final.

Por exemplo, com dois locais de usuário final, um servindo 90% do tráfego com um RTT de 5 ms e o outro servindo 10% do tráfego com um RTT de 10 ms, o resultado é um RTT agregado de 5,5 ms (resultado de 5 ms * 0,9 + 10 ms * 0,1).

Observe que há diferenças nos recursos sobre como medir a latência da última milha. Para medições de latência do Monitor de Internet, as VPCs, os Network Load Balancers e os diretórios do WorkSpaces não incluem a latência de última milha.

Cálculo de pontuações de desempenho e de disponibilidade

A AWS tem dados históricos substanciais sobre a performance e a disponibilidade de Internet entre os serviços da AWS e diferentes cidades-redes (locais ou ASNs). Aplicando a análise estatística aos dados, o Monitor de Internet pode detectar quando a performance e a disponibilidade da aplicação diminuíram, em comparação com a linha de base estimada que ele calcula. Para facilitar a visualização dessas quedas, essas informações são relatadas sob a forma de pontuações de integridade: uma pontuação de performance e uma pontuação de disponibilidade.

As pontuações de integridade são calculadas em diferentes granularidades. Com a maior granularidade, computamos a pontuação de integridade de uma região geográfica, como uma cidade ou área metropolitana, e um ASN (uma cidade-rede). Também somamos

as pontuações de integridade individuais às pontuações gerais de integridade para uma aplicação em um monitor. Se você visualizar as pontuações de performance ou de disponibilidade sem filtrar por área geográfica ou provedor de serviços específico, o Monitor de Internet fornecerá as pontuações gerais de integridade.

As pontuações gerais de integridade cobrem toda a sua aplicação durante o período de tempo especificado. Quando a pontuação de performance ou disponibilidade dos pares de cidades-redes da sua aplicação atinge ou cai abaixo do limite de eventos de integridade correspondente para performance ou disponibilidade, o Monitor de Internet aciona um evento de integridade. Por padrão, o limite é de 95% tanto para performance quanto para disponibilidade geral. O Monitor de Internet também cria eventos de integridade com base nos limites locais (se a opção estiver habilitada, como está por padrão) com base nos valores que você configurar. Para saber mais sobre a configuração de limites de eventos de integridade, consulte Alterar limites de eventos de integridade.

Ao explorar as informações nos arquivos de monitor e de log para investigar problemas e saber mais, será possível filtrar por cidades (locais), redes (ASNs ou provedores de Internet) específicas ou ambas. Portanto, é possível usar filtros para ver as pontuações de integridade de diferentes cidades, ASNs ou pares de cidades-redes, dependendo dos filtros escolhidos.

- Uma pontuação de disponibilidade representa a porcentagem estimada de tráfego que não está apresentando queda de disponibilidade. O Monitor de Internet estima a porcentagem de tráfego que está apresentando queda em relação ao tráfego total observado e às medições das métricas de disponibilidade. Por exemplo, uma pontuação de disponibilidade de 99% para um par de usuário final e local de serviço equivale a 1% do tráfego que apresenta queda de disponibilidade para esse par.
- Uma pontuação de performance representa a porcentagem do tráfego que não está apresentando queda de performance. Por exemplo, uma pontuação de performance de 99% para um par de usuário final e local de serviço equivale a 1% do tráfego que está apresentando queda de performance para esse par.

Cálculo de TTFB e de RTT (latência)

O tempo até o primeiro byte (TTFB) se refere ao tempo entre o momento em que um cliente faz uma solicitação e o momento em que ele recebe o primeiro byte de informação do servidor. Os cálculos do TTFB da AWS medem o tempo decorrido do Amazon EC2 ou do Amazon CloudFront até o nó de medição do Monitor de Internet (incluindo a última milha do nó). Ou seja, o Monitor de Internet mede o tempo do usuário até a região do Amazon EC2 para TTFB para EC2 e do usuário até o CloudFront para TTFB para CloudFront.

Para o tempo de ida e volta (RTT), o Monitor de Internet inclui o tempo da rede-cidade (ou seja, a localização do cliente e o ASN, geralmente um provedor de serviços de Internet), conforme mapeado pelo endereço IP público, até a Região da AWS. Isso significa que o Monitor de Internet não tem visibilidade de última milha para usuários que acessem a Internet por trás de um gateway ou VPN.

Observe que há diferenças nos recursos sobre como medir a latência da última milha. Para medições de latência do Monitor de Internet, as VPCs, os Network Load Balancers e os diretórios do WorkSpaces não incluem a latência de última milha.

O Monitor de Internet inclui informações médias de TTFB na seção Sugestões de otimização de tráfego da guia Insights de tráfego no painel do CloudWatch, para ajudar você a avaliar opções para diferentes configurações da sua aplicação que possam melhorar a performance.

Medições e agregação regional e de zona de disponibilidade

Embora o Monitor de Internet agregue medições e compartilhe o impacto em um nível regional, ele calcula o impacto em um nível de zona de disponibilidade (AZ). Isso significa que, se, para um evento, somente uma AZ for impactada e a maior parte do tráfego é transmitido por meio dessa AZ, você terá impacto no tráfego. No entanto, para o mesmo evento, se o tráfego da aplicação não é transmitido por uma AZ afetada, você não terá impacto.

Observe que isso se aplica somente a recursos que não são diretórios do WorkSpaces. Os diretórios do WorkSpaces são medidos somente em um nível regional.

Precisão de geolocalização no Monitor de Internet

Para obter informações de localização, o Monitor de Internet usa dados de geolocalização IP fornecidos pela MaxMind. A precisão das informações de localização nas medições do Monitor de Internet depende da precisão dos dados da MaxMind.

Esteja ciente de que as medições de nível `Metro` podem não ser precisas para localidades externas aos Estados Unidos.

Quando o Monitor de Internet cria e resolve eventos de integridade

O Monitor de Internet cria e fecha eventos de integridade para o tráfego da aplicação que você monitora com base nos limites atuais definidos. O Monitor de Internet tem uma configuração de limite padrão e também é possível definir sua própria configuração para limites. O Monitor de Internet determina o impacto geral que os problemas de conectividade estão causando na sua aplicação e o impacto nas áreas locais em que sua aplicação tem clientes e cria eventos de integridade quando os limites são ultrapassados.

O Monitor de Internet calcula o impacto dos problemas de conectividade no local do cliente com base nos dados históricos de performance e de disponibilidade de Internet para o tráfego de rede que está disponível para o serviço por meio da AWS. Ele aplica as informações relevantes para a sua aplicação, com base nas localizações geográficas dos ASNs e nos serviços em que os clientes usam sua aplicação: os pares cidade-rede afetados. Os locais são determinados pelos recursos adicionados ao seu monitor. Em seguida, o Monitor de Internet usa análise estatística para detectar quando a performance e a disponibilidade caem, afetando a experiência do cliente da sua aplicação.

As quedas de performance e de disponibilidade são representadas como o percentual de tráfego que não está apresentando queda. O impacto é o oposto: é uma representação da gravidade de um problema para os usuários finais de um cliente. Portanto, se houver uma queda de disponibilidade global de 93%, por exemplo, o impacto correspondente seria de 7%.

Quando a pontuação de performance ou disponibilidade dos pares de cidades-redes atinge ou cai abaixo do limite de eventos de integridade correspondente para performance ou disponibilidade globalmente, isso faz com que o Monitor de Internet acione um evento de integridade. Por padrão, o limite é de 95% tanto para performance quanto para disponibilidade. Os valores para atingir ou cair abaixo do limite são cumulativos, o que pode significar que vários eventos menores se combinem para atingir o percentual limite, ou que um único evento atinja ou caia abaixo do nível limite.

Desde que as pontuações de performance ou disponibilidade que acionaram o evento estejam iguais ou abaixo do percentual limite de eventos de integridade correspondente para o impacto geral, o evento de integridade permanecerá ativo. Quando a pontuação ou as pontuações combinadas que desencadearam o evento ultrapassam o limite, o Monitor de Internet resolve o evento de integridade.

O Monitor de Internet também cria eventos de integridade com base nos limites locais e no percentual do tráfego geral sobre o qual um problema afete. É possível configurar opções para limites locais ou desativar completamente os limites locais.

Para saber mais sobre a configuração de limites de eventos de integridade, consulte Alterar limites de eventos de integridade.

Temporização do relatório de eventos de integridade

O Monitor de Internet usa um agregador para reunir todos os sinais de problemas de Internet para criar eventos de integridade nos monitores em questão de minutos.

Quando possível, o Monitor de Internet analisa a origem de um evento de integridade para determinar se ele foi causado pela AWS ou por um ANS. A análise de eventos de integridade continua após a resolução de um evento. O Monitor de Internet pode atualizar eventos com novas informações por até uma hora.

Como o Monitor de Internet funciona com tráfego IPv4 e IPv6

O Monitor de Internet mede a integridade de uma rede somente por IPv4 e mostra eventos de integridade e métricas de disponibilidade e performance, se você distribuir tráfego para essa rede em qualquer família de IP (IPv4 ou IPv6). Se você transmitir o tráfego de um recurso de pilha dupla, como uma distribuição de pilha dupla do CloudFront, o Monitor de Internet gerará um evento de integridade e mostrará uma queda na pontuação de performance ou na pontuação de disponibilidade somente se o tráfego IPv4 apresentar problemas semelhantes aos do tráfego IPv6 para o recurso.

Observe que as métricas do Monitor de Internet para entrada e saída gerais de bytes refletem com precisão todo o tráfego da Internet (IPv4 e IPv6).

Como o Monitor de Internet seleciona o subconjunto de redes de cidades a ser incluído

Quando você define um limite máximo para o número de redes de cidades monitoradas pelo monitor ou escolhe uma porcentagem de tráfego para monitorar, o Monitor de Internet seleciona as redes de cidades a serem incluídas (monitoradas) com base no maior volume de tráfego recente.

Por exemplo, se você definir um limite máximo de redes de cidades de cem, o Monitor de Internet monitorará (no máximo) cem redes de cidades com base no tráfego da aplicação durante um período recente de uma hora. Especificamente, o Monitor de Internet monitora as cem principais redes de cidades que tiveram um tráfego elevado na janela de uma hora mais recente antes da última janela de uma hora.

Para ilustrar isso, suponhamos que o horário atual seja 14h30. Neste cenário, o tráfego que você visualiza em seu monitor foi capturado entre 13h e 14h, e a medição do volume de tráfego que o Monitor de Internet usa para determinar as cem principais redes de cidades foi capturada entre 12h e 13h.

Como o mapa de condições globais da Internet foi criado (perguntas frequentes)

O mapa de condições da Internet do Monitor de Internet do Amazon CloudWatch está disponível no console do Monitor de Internet para todos os clientes autenticados da AWS. Esta seção inclui detalhes sobre como o mapa de condições da Internet foi criado e como é possível usá-lo.

O que é o mapa de condições da Internet do Monitor de Internet?

O mapa de condições da Internet disponibiliza uma representação visual dos problemas relacionados com a Internet em todo o mundo. Há destaque para as localidades de clientes com impacto, ou seja, as cidades e o ASN (que normalmente são provedores de serviços de Internet). O mapa mostra uma combinação de problemas de disponibilidade e de performance que afetaram a experiência dos clientes na Internet recentemente para as principais localidades de clientes e serviços da AWS em todo o mundo.

Qual é a origem dos dados do mapa?

Os dados são baseados em uma combinação de investigações ativas e passivas da Internet. Para saber mais sobre como o Monitor de Internet realiza a medição de dados, você pode realizar a leitura da seção Como a AWS mede os problemas de conectividade.

Com que frequência ocorre a atualização do mapa?

O mapa de condições da Internet é atualizado a cada 15 minutos.

Quais redes são monitoradas em busca de interrupções?

A AWS monitora redes que representam prefixos IP importantes usados pelos clientes para efetuar conexões da Internet com a AWS em todo o mundo. Nós verificamos as interrupções para localidades de clientes que são os principais responsáveis pelo volume de tráfego enviado e recebido pela rede da AWS.

O que determina se um evento da Internet será incluso no mapa?

Apresentamos abaixo alguns critérios de alto nível que usamos para determinar se um evento da Internet será incluso no mapa de condições da Internet:

- A AWS detecta que existe um evento de disponibilidade ou de performance.
- Se o evento for de curta duração, por exemplo, durar menos de cinco minutos, nós o ignoramos.
- Se o evento ocorrer em uma localidade de cliente que é classificada como uma das principais responsáveis pelo tráfego, será considerado uma interrupção.

Quais limites são usados para o mapa de condições da Internet?

Os limites para a determinação de interrupções não são estáticos para o mapa de condições da Internet. O Monitor de Internet determina o que constitui um evento com base na detecção de um desvio dos valores esperados. Você pode obter mais informações sobre o funcionamento desse processo ao analisar como o Monitor de Internet determina quando criar eventos de integridade para monitoramentos criados com o serviço. Ao criar um

monitoramento, o Monitor de Internet gera medidas de integridade do tráfego da Internet que são específicas para o tráfego da sua própria aplicação. Além disso, o Monitor de Internet alerta você sobre os eventos de integridade relacionados a problemas que afetam o tráfego de Internet da sua aplicação.

Quais são as possibilidades de uso desses dados?

O mapa de condições da Internet fornece um breve resumo dos principais eventos da Internet que aconteceram em todo o mundo nas últimas 24 horas. Ele possibilita que você aproveite a experiência de monitoramento da Internet, sem a necessidade de integração do seu próprio tráfego de Internet ao Monitor de Internet. Para utilizar todo o potencial das funcionalidades de monitoramento da Internet da AWS e personalizá-las para suas aplicações e serviços hospedados na AWS, é possível criar um monitoramento no Monitor de Internet.

Ao criar um monitoramento, você habilita que o Monitor de Internet identifique os caminhos de Internet específicos que afetam os clientes das suas aplicações e obtém acesso a recursos e funcionalidades que podem auxiliar na melhora da experiência do cliente. Além disso, você receberá notificações de forma proativa sobre os novos problemas relacionados à Internet que afetam especificamente o tráfego e os clientes da sua aplicação.

Como é possível obter mais detalhes sobre os eventos?

Clique em uma interrupção no mapa para visualizar os detalhes, os quais incluem o momento em que um evento começou e terminou, a cidade e o ASN afetados, e qual tipo de problema ocorreu (ou seja, se foi um problema de performance ou de disponibilidade).

Para obter informações mais detalhadas sobre os eventos e obter medições personalizadas para o tráfego da sua aplicação, crie um monitoramento no Monitor de Internet.

Casos de exemplo de uso do Monitor de Internet do Amazon CloudWatch

Nesta seção, descrevemos vários exemplos específicos, com links para postagens de blog com mais detalhes. Esses exemplos mostram como é possível usar os recursos do Monitor de Internet do Amazon CloudWatch para ajudar a monitorar sua aplicação e melhorar a experiência de seus usuários.

Configure alertas e decida as ações a serem tomadas

É possível usar o Monitor de Internet para obter insights sobre as métricas médias de performance da Internet ao longo do tempo e sobre eventos de integridade por cidade-rede

(localização do cliente e ASN, normalmente um provedor de serviços de Internet). Usando o Monitor de Internet, é possível identificar os eventos que estão afetando a experiência do usuário final em aplicações hospedadas em Amazon Virtual Private Clouds (VPCs), Network Load Balancers, Amazon WorkSpaces ou Amazon CloudFront.

Depois de criar um monitor, você tem várias opções de como ser alertado sobre eventos de integridade do Monitor de Internet. Isso inclui notificações baseadas em alarmes do CloudWatch usando métricas de eventos ou regras do Amazon EventBridge para filtrar eventos de integridade. É possível escolher opções diferentes para notificações ou ações com base em alarmes, incluindo, por exemplo, notificações ou atualizações por AWS SMS para um grupo de logs do CloudWatch.

Para ver um exemplo com orientações detalhadas, consulte a postagem a seguir no blog: Apresentação do Monitor de Internet do Amazon CloudWatch.

Identifique problemas de latência e melhore o TTFB para melhorar a experiência de jogo multijogador

Use o Monitor de Internet para obter ajuda para identificar rapidamente onde os jogadores em aplicações globais de jogos em nuvem estão enfrentando problemas de latência em todo o mundo e fornecer insights sobre como melhorar a performance. Ao identificar onde a maioria dos jogadores atualmente tem o tempo mais lento até o primeiro byte (TTFB), você saberá como melhorar a latência para deixar sua maior base de jogadores mais feliz.

Agora, quando você estiver pronto para implantar o próximo servidor EC2 para seu jogo, escolha a Região da AWS que o Monitor de Internet sugerir que reduzirá o TTFB na área com alta latência e um grande grupo de jogadores.

Para obter detalhes sobre como configurar e usar o Monitor de Internet para esse caso de uso, consulte a postagem a seguir no blog: Usando o Monitor de Internet do Amazon CloudWatch para obter uma melhor experiência de jogo.

Observabilidade entre contas do Monitor de Internet

Com a observabilidade entre contas do Monitor de Internet, é possível monitorar aplicações que abrangem várias contas da AWS em uma única Região da AWS.

É possível usar o Amazon CloudWatch Observability Access Manager para configurar uma ou mais de suas contas da AWS como conta de monitoramento. Você fornecerá à conta de monitoramento a capacidade de visualizar dados em sua conta de origem criando um coletor em sua conta de monitoramento. Um coletor é um recurso que representa um ponto de conexão em uma conta de

monitoramento. Para o Monitor de Internet, o ponto de conexão do recurso é um monitor. Você usa o coletor para criar um link da sua conta de origem para sua conta de monitoramento. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

Recursos necessários do

Para a funcionalidade adequada de observabilidade entre contas do CloudWatch Application Insights, certifique-se de que os tipos a seguir de telemetria sejam compartilhados por meio do CloudWatch Observability Access Manager.

- Monitores no Monitor de Internet
- Métricas no Amazon CloudWatch
- Grupos de logs no Amazon CloudWatch Logs

Introdução ao Monitor de Internet do Amazon CloudWatch usando o console

Para começar a usar o Monitor de Internet do Amazon CloudWatch, você deve criar um monitor no Monitor de Internet para sua aplicação, adicionando recursos da AWS que ele utiliza e definindo várias opções de configuração. Este capítulo fornece o procedimento para adicionar um monitor no console. Ele também inclui uma seção com mais detalhes sobre os recursos do Monitor de Internet e, em seguida, seções adicionais com descrições e limitações das diferentes opções que podem ou devem ser configuradas para o seu monitor.

Conteúdo

Criação de um monitor no Monitor de Internet do Amazon CloudWatch usando o console

Você cria um monitor no Monitor de Internet do Amazon CloudWatch para sua aplicação adicionando recursos da AWS por ela usados e, em seguida, definindo várias opções de configuração. Os recursos que você adiciona, por exemplo, as Amazon Virtual Private Clouds (VPCs), os Network Load Balancers (NLBs), as distribuições do CloudFront ou os diretórios do WorkSpaces, fornecem as informações para o Monitor de Internet mapear as informações de tráfego da Internet para a aplicação. Depois de criar o monitor, aguarde de 15 a 30 minutos para gerar o perfil de tráfego específico para a localidade em que a aplicação é usada. Em seguida, é possível usar o Monitor de Internet ou outras ferramentas para visualizar e explorar o desempenho e a disponibilidade sobre o uso do seu cliente. Essas ferramentas fornecem insights usando as medições do tráfego da sua aplicação, coletadas e publicadas pelo monitor, por exemplo, no CloudWatch Logs.

Normalmente, é mais simples criar um monitor no Monitor de Internet para uma aplicação. No mesmo monitor, é possível pesquisar e classificar medidas e métricas nos arquivos de log do Monitor de Internet por diferentes locais e ASNs (geralmente provedores de serviços de Internet) ou outras informações. Não é necessário criar monitores separados para aplicações em áreas diferentes, por exemplo.

As etapas desta seção guiam você na configuração do seu monitor usando o console. Para ver exemplos de uso da AWS Command Line Interface com as ações de API do Monitor de Internet, para criar um monitor, visualizar eventos e assim por diante, consulte Exemplos de uso da CLI com o Monitor de Internet do Amazon CloudWatch.

Para criar um monitor usando o console

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, em Monitoramento de rede, escolha Monitor de Internet.
3. Escolha Criar monitor.
4. Em Monitor name (Nome do monitor), insira o nome que você deseja usar para esse monitor no Monitor de Internet.
5. Escolha Add resources (Adicionar recursos) e selecione os recursos para definir os limites de monitoramento para o Monitor de Internet usar para esse monitor.

> **Note**
>
> Esteja ciente do seguinte:

- Para gerar uma saída significativa com o Monitor de Internet, as VPCs adicionadas devem estar conectadas à Internet com um gateway da Internet configurado.
- É possível adicionar uma combinação de VPCs e distribuições do CloudFront, adicionar diretórios do WorkSpaces ou adicionar Network Load Balancers. Não é possível adicionar Network Load Balancers ou diretórios do WorkSpaces com outros tipos de recursos.

6. Escolha um percentual do seu tráfego na Internet para monitorar.
7. Opcionalmente, especifique opções adicionais em Configurações avançadas.

   - Em Limite máximo de redes urbanas, você pode selecionar um limite para o número de redes urbanas (localizações e ASNs ou prestadores de serviços de Internet) para as quais o Monitor de Internet monitorará o tráfego. É possível alterar isso a qualquer momento editando seu monitor. Consulte Escolha de um limite máximo de cidades-redes.

     Para redefinir para o padrão, insira `500000`.

     Se você definir um limite máximo de cidades-redes, ele definirá um limite para o número de cidades-redes que o Monitor de Internet monitorará para sua aplicação, independentemente do percentual de tráfego que você escolher monitorar.
   - Opcionalmente, você pode especificar um nome de bucket do Amazon S3 e um prefixo personalizado para publicar medições de Internet no Amazon S3 para todas as redes urbanas monitoradas.

     O Monitor de Internet publica as 500 principais medições da Internet (por volume de tráfego) para sua aplicação no CloudWatch Logs a cada cinco minutos. Se você optar por publicar medições no S3, as medidas ainda serão publicadas no CloudWatch Logs. Para ter mais informações, consulte Publicação de medições da Internet para o Amazon S3 no Monitor de Internet do Amazon CloudWatch.
   - Opcionalmente, você pode adicionar uma tag para o monitor.
8. Escolha Criar monitor.

Depois de criar um monitor, é possível editá-lo a qualquer momento, por exemplo, para alterar o percentual de tráfego da aplicação, atualizar o limite máximo de cidades-redes ou adicionar ou remover recursos. Também é possível excluir o monitor. Para realizar essas tarefas, no console do

Monitor de Internet, selecione um monitor e escolha uma opção no menu Ação. Observe que não é possível alterar o nome de um monitor.

Para visualizar o painel do Monitor de Internet

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Monitoramento de rede e depois Monitor de Internet.

   A guia Monitors (Monitores) exibe uma lista dos monitores que você criou.

Para ver mais informações sobre um monitor, escolha-o.

Adição de recursos ao seu monitor

Ao criar um monitor, você deve associar os recursos da aplicação a ele: as Amazon Virtual Private Clouds (VPCs), as distribuições do Amazon CloudFront, os Network Load Balancers (NLBs) ou os diretórios do Amazon WorkSpaces. Assim, o Monitor de Internet saberá onde estão localizados o tráfego e os clientes voltados para a Internet da aplicação e pode criar e manter um perfil de tráfego que determina as medidas relevantes a serem publicadas para o monitor.

É possível adicionar os tipos de recursos apresentados a seguir a um monitor no Monitor de Internet como “recursos monitorados”. Observe que o Monitor de Internet não é compatível com a adição de diferentes tipos de recursos em um monitor.

- VPCs: cada VPC que você adiciona a uma região é um recurso monitorado. Quando você adiciona uma VPC, o Monitor de Internet realiza o monitoramento do tráfego para qualquer aplicação voltada para a Internet na VPC, por exemplo, uma aplicação hospedada em uma instância do Amazon EC2, protegida por um Network Load Balancer, ou em um contêiner do AWS Fargate.
- Network Load Balancer: cada Network Load Balancer que você adiciona é um recurso monitorado.
- Distribuições do CloudFront: cada distribuição do CloudFront que você adiciona é um recurso monitorado.
- Diret’prios do WorkSpaces: cada diretório do WorkSpaces que você adiciona a uma região é um recurso monitorado.

Quando você monitora o tráfego de VPCs, o tráfego das aplicações hospedadas em balanceadores de carga atrás da VPC é monitorado. É possível optar por monitorar o tráfego de balanceadores de carga individuais do Network Load Balancer em vez de monitorar uma VPC com vários

balanceadores de carga. Isso pode ser útil, por exemplo, se você precisar entender e configurar recursos para melhorar a performance ou a eficiência no nível do balanceador de carga. Ou talvez você precise de informações de conformidade no nível do Network Load Balancer.

Quando você adiciona recursos a um monitor no Monitor de Internet, esteja ciente de que:

- Para gerar uma saída significativa com o Monitor de Internet, as VPCs adicionadas devem estar conectadas à Internet com um gateway da Internet configurado.
- O Monitor de Internet não é compatível com a adição de diferentes tipos de recursos em um monitor.

Há diferenças regionais para regiões de aceitação que devem ser lembradas ao adicionar VPCs ou NLBs como recursos. Para ter mais informações, consulte Regiões da AWS com suporte para o Monitor de Internet do Amazon CloudWatch.

Além disso, há diferenças nos recursos sobre como medir a latência da última milha. Para medições de latência do Monitor de Internet, as VPCs, os NLBs e os diretórios do WorkSpaces não incluem a latência de última milha.

Escolha de um percentual de tráfego da aplicação a monitorar

A cobertura que você escolhe para o percentual de tráfego da aplicação a ser monitorada determina quantas cidades-redes (locais de clientes e ASNs, geralmente provedores de serviços de Internet) da sua aplicação são monitoradas, até um limite máximo opcional de cidades-redes que também é possível definir.

Se você optar por monitorar menos de 100% do tráfego da sua aplicação, talvez exista uma lacuna de observabilidade no seu monitor. Isso porque, se houver eventos de integridade criados pelo Monitor de Internet do Amazon CloudWatch em que você não esteja monitorando o tráfego, você não estará ciente desses problemas. Também é possível ter menos cobertura para as informações de pontuação de performance e disponibilidade sobre o acesso do cliente à sua aplicação.

As seções a seguir descrevem opções para explorar as configurações de porcentagem de tráfego e cobertura, e também para ter uma ideia do impacto do aumento ou da diminuição da cobertura.

- Explore a alteração do percentual de tráfego da sua aplicação
- Veja o número de cidades-redes monitoradas em diferentes configurações de percentual de tráfego

Explore a alteração do percentual de tráfego da sua aplicação

É possível explorar valores para os quais talvez queira alterar o percentual de tráfego a sua aplicação, visualizando o número de cidades-redes monitoradas ao alterar o percentual. O procedimento nesta seção fornece informações detalhadas.

No console do Monitor de Internet, é possível tentar aumentar ou diminuir o percentual de tráfego de aplicações para o seu monitor e visualizar o número estimado de suas cidades-redes que seriam cobertas como resultado. Com essa opção, é possível ver rapidamente como a alteração do percentual de tráfego afeta o número de cidades-redes monitoradas. Isso pode ajudar você a ter uma ideia de qual seria um bom percentual de tráfego de aplicações a ser escolhido para sua aplicação.

Para explorar a cobertura de monitoramento aumentando e diminuindo o percentual de tráfego da aplicação

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, em Monitoramento de rede, escolha Monitor de Internet.
3. Na sua lista de monitores, escolha um monitor.
4. Na guia Visão geral, na seção Tráfego monitorado, escolha o gráfico percentual e, em seguida. escolha Atualizar cobertura de monitoramento.
5. Na caixa de diálogo Explorar e definir a cobertura do monitoramento de tráfego, clique nas setas para aumentar ou diminuir o percentual de tráfego a ser monitorado. Ao escolher 100% de tráfego, será possível ver quantas cidades-redes são monitoradas com cobertura total para monitorar sua aplicação.
6. Para saber mais sobre como o número de cidades-redes monitoradas (estimado aqui) pode afetar seus custos, escolha o link para a calculadora de preços do CloudWatch e, em seguida, role para baixo até o Monitor de Internet.
7. Para definir um novo percentual de tráfego a ser monitorado, escolha Atualizar cobertura do monitor. Ou, para manter o nível de cobertura atual, escolha Cancelar.

Veja o número de cidades-redes monitoradas em diferentes configurações de percentual de tráfego

É possível visualizar o número de cidades-redes que seriam monitoradas para sua aplicação em diferentes percentuais de tráfego da aplicação. O procedimento nesta seção fornece informações detalhadas.

No console do Monitor de Internet, é possível visualizar gráficos que mostram como a cobertura de suas cidades-redes mudaria em diferentes percentuais de tráfego de aplicações, em um intervalo de tempo especificado por você. Essa é uma maneira rápida de visualizar e comparar a cobertura de monitoramento da sua aplicação em percentuais específicos de tráfego, tudo em um só gráfico.

Para visualizar gráficos do percentual de tráfego de aplicações e da cobertura correspondente das cidades-redes

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, em Monitoramento de rede, escolha Monitor de Internet.
3. Na sua lista de monitores, escolha um monitor.
4. Escolha a guia Insights de tráfego e role para baixo até Gráficos de tráfego da Internet.
5. Em Comparar opções de cobertura de tráfego, na lista suspensa, selecione um ou mais percentuais. É possível escolher um ou mais percentuais de tráfego de aplicações, e o gráfico do Total de cidades-redes monitoradas é atualizado para exibir a cobertura de monitoramento que o Monitor de Internet fornece para esse percentual de tráfego. Ao escolher Cidades-redes com 100% de tráfego, será possível ver quantas cidades-redes são monitoradas com cobertura total para monitorar sua aplicação.

Lembre-se do seguinte:

- A cobertura de tráfego é calculada com base no número de cidades-redes na hora anterior do tráfego da sua aplicação. Isso significa que, depois de escolher uma porcentagem específica de tráfego para monitorar, menos cidades-redes poderão ser monitoradas para a sua aplicação do que é mostrado aqui em um gráfico de comparação de cobertura de tráfego.
- Para garantir que todo o tráfego da sua aplicação seja monitorado, defina `TrafficPercentageToMonitor` como 100 e não defina `MaxCityNetworksToMonitor`. Como alternativa, você pode definir `MaxCityNetworksToMonitor` como 500.000, o limite superior no Monitor de Internet.
- Se você definir um limite máximo de cidades-redes, o número total de cidades-redes monitoradas nunca excederá esse limite, independentemente da opção de porcentagem de tráfego de aplicações que você selecionar.
- É possível aprender mais sobre como o número de cidades-redes monitoradas pode afetar seus custos. Na página Calculadora de preços do CloudWatch, role para baixo até o Monitor de Internet.

Para definir um novo percentual de tráfego a ser monitorado, em Explorar outras opções de cobertura de tráfego, escolha Atualizar cobertura de monitoramento. Na caixa de diálogo, escolha um percentual de tráfego e, em seguida, escolha Atualizar cobertura do monitor.

Escolha de um limite máximo de cidades-redes

O Monitor de Internet do Amazon CloudWatch pode monitorar o tráfego da sua aplicação para alguns ou todos os locais em que os clientes acessam seus recursos de aplicações e todos os ASNs (geralmente provedores de serviços de Internet) pelos quais eles acessam sua aplicação, ou seja, as cidades-redes para o tráfego de Internet da sua aplicação. Você escolhe um percentual de tráfego de aplicaçcão para monitorar ao criar seu monitor, que pode ser atualizada a qualquer momento editando o monitor.

Além de definir um percentual de tráfego, também é possível definir um limite máximo para o número de cidades-redes monitoradas. Esta seção descreve como o limite de cidades-redes pode ajudá-lo a gerenciar os custos de cobrança e fornece informações e um exemplo para ajudar a determinar um limite a ser definido.

O limite máximo que você define para o número de cidades-redes ajuda a garantir que sua fatura seja previsível. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch). Também é possível aprender como valores diferentes do número de cidades-redes realmente monitoradas podem afetar sua fatura usando a calculadora de preços do CloudWatch. Para explorar as opções, na página Calculadora de preços do CloudWatch, role para baixo até o Monitor de Internet.

Para atualizar seu monitor e alterar o limite máximo de cidades-redes, consulte. Edição ou exclusão de um monitor do Monitor de Internet

Como o faturamento funciona com os limites máximos de cidades-redes

Definir um limite máximo para o número de cidades-redes monitoradas pode ajudar a evitar custos inesperados em sua fatura. Isso é útil, por exemplo, se seus padrões de tráfego variam muito. Os custos de cobrança aumentam para cada cidade-rede monitorada após as primeiras 100 cidades-redes, que estão incluídas (em todos os monitores por conta). Se você definir um limite máximo de cidades-redes, restringirá o número de cidades-redes que o Monitor de Internet monitorará para sua aplicação, independentemente do percentual de tráfego que você escolher monitorar.

Você paga apenas pelo número de cidades-redes que são realmente monitoradas. O limite máximo de cidades-redes que você escolher permite definir um limite para o total que poderá ser incluído

quando o Monitor de Internet monitorar o tráfego com seu monitor. É possível alterar o limite máximo a qualquer momento editando seu monitor.

Para explorar as opções, na página Calculadora de preços do CloudWatch, role para baixo até o Monitor de Internet. Para obter mais informações sobre os preços do Monitor de Internet, consulte a seção Monitor Internet na página de preços do Amazon CloudWatch.

Como escolher um limite máximo de cidades-redes

Para ajudar a decidir sobre o limite máximo de cidades-redes a ser selecionado, considere a quantidade de tráfego que você deseja monitorar para sua aplicação. As métricas a seguir do Monitor da Internet podem ajudar a analisar o uso e a cobertura do tráfego depois de criar o monitor: `CityNetworksMonitored`, `TrafficMonitoredPercent`, e uma ou mais da métricas `CityNetworksFor`*`NN`*`PercentTraffic`, onde *`NN`* é um dos valores percentuais a seguir: 25, 50, 90, 95, 99 ou 100. Para revisar as definições dessas métricas e de todas as outras métricas do Monitor de Internet, consulte Usar o CloudWatch Logs Metrics com o Monitor de Internet do Amazon CloudWatch.

Para ver um gráfico de visão geral da sua cobertura de tráfego na Internet, acesse a guia Insights de tráfego no painel do CloudWatch e, na seção Gráficos de tráfego da Internet, escolha uma opção para Comparar opções de cobertura de tráfego. O gráfico mostrado na seção exibe o número real de cidades-redes que são monitoradas para sua aplicação, bem como linhas gráficas para diferentes percentuais de tráfego de aplicações que você seleciona na lista suspensa. Para saber mais, consulte Configuração do percentual de tráfego da sua aplicação.

Para explorar suas opções com mais detalhes, é possível usar as métricas do Monitor de Internet, conforme descrito nos exemplos a seguir. Estes exemplos mostram como selecionar um limite máximo de cidades-redes que seja melhor para você, dependendo da amplitude da cobertura de tráfego de Internet da aplicação que você deseja. O uso das consultas para métricas do Monitor de Internet no CloudWatch Metrics pode ajudar a entender mais sobre a cobertura do tráfego de Internet da sua aplicação.

Exemplo de determinação de um limite máximo de cidades-redes

Por exemplo, digamos que você tenha definido um limite máximo de monitoramento de 100 cidades-redes e que sua aplicação seja acessada por clientes em 2637 cidades-redes. No CloudWatch Metrics, você veria as métricas a seguir do Monitor de Internet retornadas:

```
CityNetworksMonitored 100
TrafficMonitoredPercent  12.5
```

```
CityNetworksFor90PercentTraffic  2143
CityNetworksFor100PercentTraffic  2637
```

Neste exemplo, é possível ver que atualmente está monitorando 12,5% do seu tráfego na Internet, com o limite máximo definido para 100 cidades-redes. e você quiser monitorar 90% do seu tráfego, a próxima métrica fornecer informações sobre isso: `CityNetworksFor90PercentTraffic` indica que você precisaria monitorar 2.143 cidades-redes para obter 90% de cobertura. Para fazer isso, você deve atualizar seu monitor e definir o limite máximo de cidades-redes para 2.143.

Da mesma forma, digamos que você queira ter 100% de monitoramento do tráfego da Internet para sua aplicacão. A próxima métrica, `CityNetworksFor100PercentTraffic`, indica que, para fazer isso, você deve atualizar seu monitor para definir o limite máximo de cidades-redes para 2.637.

Se agora você definir o máximo para 5.000 cidades-redes, já que isso é maior que 2.637, você verá as métricas retornadas a seguir:

```
CityNetworksMonitored 2637
TrafficMonitoredPercent  100
CityNetworksFor90PercentTraffic  2143
CityNetworksFor100PercentTraffic  2637
```

A partir dessas métricas, é possível ver que, com o limite mais alto, você monitora todas as 2.637 cidades-redes, o que representa 100% do seu tráfego na Internet.

Publicação de medições da Internet para o Amazon S3 no Monitor de Internet do Amazon CloudWatch

É possível optar por fazer com que o Monitor de Internet do Amazon CloudWatch publique medições de Internet no Amazon S3 para seu tráfego de Internet para as cidades-redes monitoradas (localizações de clientes e ASNs, geralmente provedores de serviços de Internet) em seu monitor, até o limite de serviço de 500.000 cidades-redes. O Monitor de Internet publica automaticamente medições da Internet no CloudWatch Logs a cada cinco minutos para as 500 principais (por volume de tráfego) cidades-redes para cada monitor. As medidas que ele publica no S3 incluem as 500 principais publicadas no CloudWatch Logs.

É possível escolher a opção de publicar no S3 e especificar o bucket para publicar as medições ao criar ou atualizar seu monitor. O bucket já deve ter sido criado no S3 para que você possa especificá-lo no Monitor de Internet. Há um limite de serviço de 500.000 cidades-redes para medições de Internet publicadas no S3. O Monitor de Internet publica medições de Internet no S3 como eventos, uma série de objetos de log compactados armazenados no bucket.

Ao criar o bucket do S3 para o Monitor de Internet publicar medições, certifique-se de seguir as orientações de permissões fornecidas pelo CloudWatch Logs. Isso garante que o Monitor de Internet possa publicar logs diretamente no S3 e, que a AWS possa, se necessário, criar e alterar as políticas de recursos associadas ao grupo de logs que os recebe. Para obter mais informações, consulte Logs enviados para o CloudWatch Logs no Guia do usuário do Amazon CloudWatch Logs.

Os arquivos de log publicados são compactados. Se você abrir os arquivos de log usando o console do Amazon S3, eles serão descompactados, e os eventos de medição da Internet serão exibidos. Se você baixar os arquivos, será necessário descompactá-los para visualizar os eventos.

Também é possível consultar as medicões de Internet nos arquivos de log usando o Amazon Athena. O Amazon Athena é um serviço de consultas interativas que facilita a análise de dados no Amazon S3 usando SQL padrão. Para ter mais informações, consulte Uso do Amazon Athena para consultar medições de Internet nos arquivos de log do Amazon S3.

Uso de um monitor do Monitor de Internet

Há várias maneiras de se usar um monitor do Monitor de Internet do Amazon CloudWatch depois de tê-lo criado: por exemplo, é possível visualizar informações no painel do CloudWatch, obter informações usando o AWS Command Line Interface e definir alertas de integridade.

Seu monitor fornece informações sobre sua aplicação e preferências de configuração para que o Monitor da Internet possa personalizar as medições e métricas a serem publicadas em eventos para você. O Monitor de Internet coleta medições da cobertura da infraestrutura global para AWS. Essas medições são uma quantidade enorme de informações de performance e disponibilidade da rede, de todo o mundo. Ao usar as informações dos recursos que você adiciona à sua aplicação, o Monitor de Internet publica medidas de performance e disponibilidade para você com escopo nas cidades-redes (ou seja, locais de clientes e ASNs, geralmente provedores de serviços de Internet ou ISPs) em que sua aplicação esteja ativa. Portanto, as medidas e métricas no painel do Monitor de Internet e no CloudWatch Logs (sobre disponibilidade, performance, bytes monitorados transferidos e tempo de ida e volta) são específicas para os locais de seus clientes e ASNs.

O Monitor de Internet também determina quando há anomalias na performance e na disponibilidade. Por padrão, o Monitor de Internet sobrepõe ao seu tráfego as medições de disponibilidade e performance que a AWS coletou para cada par origem-destino nas localizações de seus clientes, para determinar quando há quedas significativas de performance ou de disponibilidade. Quando há uma degradação significativa nos locais e no escopo da sua aplicação, o Monitor de Internet gera um evento de integridade e publica informações sobre o problema no seu monitor.

Depois de configurar um monitor, é possível usá-lo para acessar ou ser alertado sobre as informações que o Monitor de Internet fornece, das formas a seguir:

- Use o painel do CloudWatch para visualizar e explorar os eventos de performance, disponibilidade e integridade, explorar os dados históricos da aplicação e obter insights sobre novas maneiras de configurar a aplicação para melhorar a performance. Para saber mais, consulte:
  - Monitoramento dar performance e disponibilidade em tempo real no Monitor de Internet do Amazon CloudWatch (guia Overview [Visão geral])
  - Filtragem e visualização de dados no Monitor de Internet do Amazon CloudWatch (guia Explorador de histórico)
  - Obter informações para melhorar a performance da aplicação no Monitor de Internet do Amazon CloudWatch (guia Insights de tráfego)
- Configure limites de eventos de integridade para alterar o que aciona o Monitor de Internet para criar um evento de integridade para sua aplicação. É possível configurar limites gerais e limites locais (cidade-rede). Para saber mais, consulte Alterar limites de eventos de integridade.
- Use comandos da AWS CLI com as operações de API do Monitor de Internet para visualizar informações sobre o perfil de tráfego, visualizar medições, listar eventos de integridade e assim por diante. Para saber mais, consulte Exemplos de uso da CLI com o Monitor de Internet do Amazon CloudWatch.
- Use ferramentas padrão do CloudWatch, como o CloudWatch Contributor Insights, o explorador do CloudWatch Metrics e o CloudWatch Logs Insights para visualizar os dados no CloudWatch. Para saber mais, consulte Como explorar seus dados com as ferramentas do CloudWatch e a interface de consulta do Monitor de Internet.
- Use o Athena com logs do S3 para acessar e analisar as medições de Internet do Monitor de Internet para sua aplicação, caso você tenha ativado a publicação de medições no S3.
- Crie notificações do Amazon EventBridge para receber alertas quando o Monitor de Internet determinar que há um evento de integridade. Para saber mais, consulte Usar o Monitor de Internet do Amazon CloudWatch com o Amazon EventBridge.
- Receba uma notificação do AWS Health Dashboard automaticamente, quando o Monitor de Internet determinar que um problema é causado pela AWS. A notificação inclui as etapas que a AWS está tomando para mitigar o problema.

Edição ou exclusão de um monitor do Monitor de Internet

Usando o menu Ação, é possível editar ou excluir um monitor no Monitor de Internet do Amazon CloudWatch depois de criá-lo. Por exemplo, é possível editar um monitor para fazer o seguinte:

- Escolha o percentual de tráfego de aplicação a monitorar
- Defina ou atualize o limite máximo das cidades-redes
- Altere os limites de eventos de integridade para obter pontuações de disponibilidade ou performance
- Adicione ou remova recursos.
- Habilite ou atualize eventos de publicação no Amazon S3

Também é possível excluir um monitor. Observe que você não pode alterar o nome de um monitor depois de criá-lo.

Para fazer alterações em um monitor ou excluí-lo, use um dos procedimentos a seguir.

Para editar um monitor

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, em Monitoramento de rede, escolha Monitor de Internet.
3. Escolha seu monitor e, em seguida, escolha o menu Ação.
4. Escolha Atualizar monitor.
5. Faça as atualizações desejadas. Por exemplo, para alterar o percentual de tráfego a ser monitorado, em Tráfego da aplicação a monitorar, selecione ou insira um percentual.
6. Escolha Atualizar.

Para excluir um monitor

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, em Monitoramento de rede, escolha Monitor de Internet.
3. Escolha seu monitor e, em seguida, escolha o menu Ação.
4. Escolha Disable.
5. Escolha o menu Ação novamente e, em seguida, escolha Excluir.

Para obter mais informações sobre as opções que podem ser atualizadas, consulte o seguinte:

- Para saber mais sobre os recursos que você adiciona no Monitor de Internet, consulte Adição de recursos ao seu monitor.
- Para saber mais sobre o percentual de tráfego de aplicação, consulte Escolha de um percentual de tráfego da aplicação a monitorar.
- Para saber mais sobre a alteração do limite de eventos de integridade, consulte Alterar limites de eventos de integridade.
- Para saber mais sobre o limite máximo de cidades-redes, consulte Escolha de um limite máximo de cidades-redes.
- Para saber mais sobre a opção de publicar eventos no S3, consulte Publicação de medições da Internet para o Amazon S3 no Monitor de Internet do Amazon CloudWatch.

Adicionar ou criar um Monitor de Internet do Amazon CloudWatch com a Amazon VPC

Ao criar uma VPC do Amazon Virtual Private Cloud no AWS Management Console, você pode optar por também configurar o monitoramento para ela no Monitor de Internet do Amazon CloudWatch. Você pode adicionar a VPC a um monitor existente ou pode optar por criar um novo monitor para a VPC no console do Amazon VPC.

Ao usar o Monitor de Internet com sua VPC, você pode visualizar e avaliar medidas e métricas sobre disponibilidade, performance, bytes monitorados transferidos e tempos de ida e volta específicos para os locais dos clientes e ASNs da sua aplicação (normalmente, provedores de serviços de Internet). O Monitor de Internet também determina quando há anomalias na performance e na disponibilidade e cria eventos de integridade em seu monitor, em relação aos quais é possível receber notificações. Para saber mais sobre como usar um monitor para gerenciar e melhorar a experiência dos clientes com sua aplicação, consulte Uso de um monitor do Monitor de Internet.

> **Important**
>
> Para criar um monitor ou adicionar uma VPC a um monitor existente, é necessário que as permissões corretas estejam em vigor. Para ter mais informações, consulte Identity and Access Management para o Monitor de Internet do Amazon CloudWatch.

Adicionar uma VPC a um monitor existente

Você pode optar por fazer com que o Monitor de Internet do Amazon CloudWatch adicione uma nova VPC a um monitor existente quando você criar a VPC no AWS Management Console. Depois de adicionar a VPC, aguarde alguns minutos e, em seguida, as métricas da VPC começarão a ser mostradas no console do Monitor de Internet.

Você pode editar o monitor a qualquer momento para remover a VPC ou adicionar outra VPC ou outros recursos. Você também pode alterar a porcentagem do tráfego que está sendo monitorado ou fazer outras alterações. Se você optar por remover a VPC do monitor, o tráfego dos clientes para essa VPC não será mais monitorado pelo Monitor de Internet.

Para saber mais sobre como atualizar um monitor, consulte Edição ou exclusão de um monitor do Monitor de Internet.

Criar um monitor para uma VPC

Se você optar por criar um monitor para uma VPC, o assistente Criar monitor apresentará a você as etapas. Você adiciona a VPC como um recurso monitorado ao criar o monitor. Se quiser, você também poderá escolher uma porcentagem do tráfego de clientes que deseja monitorar para a aplicação (o padrão é 100%).

Você pode saber mais revisando as informações em Criação de um monitor no Monitor de Internet do Amazon CloudWatch usando o console.

Definição de preço

Com o Monitor de Internet do Amazon CloudWatch, você paga apenas pelo que usa. O preço do Monitor de Internet tem dois componentes: uma taxa por recurso monitorado e uma taxa por cidade-rede. Uma rede urbana é o local de onde os clientes acessam os recursos da sua aplicação e a rede (um ASN, como um provedor de serviços de Internet ou ISP) pela qual os clientes acessam os recursos.

Para obter mais informações, incluindo exemplos de preços, consulte Informações de preços do Monitor de Internet do Amazon CloudWatch..

Interromper o monitoramento de uma VPC

Se quiser parar de monitorar seu recurso VPC com o Monitor de Internet, faça o seguinte no console do Monitor de Internet:

Remover um recurso de um monitor

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, em Monitoramento de rede, escolha Monitor de Internet.
3. Escolha seu monitor e, em seguida, escolha o menu Ação.
4. Escolha Atualizar monitor.
5. Em Recursos adicionados, selecione Remover recursos.
6. Escolha a VPC a ser removida e, em seguida, escolha Remover.
7. Escolha Atualizar.

Adicionar ou criar um Monitor de Internet do Amazon CloudWatch com o CloudFront

No painel de métricas de uma distribuição no console do Amazon CloudFront, você pode configurar um monitoramento adicional para uma distribuição no Monitor de Internet do Amazon CloudWatch. É possível adicionar a distribuição a um monitor existente ou criar um novo monitor para a distribuição.

Ao usar o Monitor de Internet com sua distribuição do CloudFront, você pode visualizar e avaliar medidas e métricas sobre disponibilidade, performance, bytes monitorados transferidos e tempos de ida e volta específicos para os locais dos clientes e ASNs da sua aplicação (normalmente, provedores de serviços de Internet). O Monitor de Internet também determina quando há anomalias na performance e na disponibilidade e cria eventos de integridade em seu monitor, em relação aos quais é possível receber notificações. Para saber mais sobre como usar um monitor para gerenciar e melhorar a experiência dos clientes com sua aplicação, consulte Uso de um monitor do Monitor de Internet.

> **Important**
>
> Para criar um monitor ou adicionar uma distribuição a um monitor existente, é necessário que as permissões corretas estejam em vigor. Para ter mais informações, consulte Identity and Access Management para o Monitor de Internet do Amazon CloudWatch.

Adicionar uma distribuição a um monitor existente

É possível fazer com que o Monitor de Internet adicione uma distribuição a um monitor existente diretamente do painel de métricas do CloudFront no AWS Management Console. Depois de

adicionar a distribuição, aguarde alguns minutos e, em seguida, as métricas para a distribuição começarão a ser mostradas no console do Monitor de Internet.

Você pode editar o monitor a qualquer momento para remover a distribuição ou adicionar outra distribuição ou outros recursos. Você também pode alterar a porcentagem do tráfego que está sendo monitorado ou fazer outras alterações. Se você optar por remover a distribuição do monitor, o tráfego dos clientes para essa distribuição não será mais monitorado pelo Monitor de Internet.

Para saber mais sobre como atualizar um monitor, consulte Edição ou exclusão de um monitor do Monitor de Internet.

Para criar um monitor para uma distribuição

Se você optar por criar um monitor para uma distribuição, o assistente Criar monitor o orientará pelas etapas. Você adiciona a distribuição como um recurso monitorado ao criar o monitor. Se quiser, você também poderá escolher uma porcentagem do tráfego de clientes que deseja monitorar para a aplicação (o padrão é 100%).

Você pode saber mais revisando as informações em Criação de um monitor no Monitor de Internet do Amazon CloudWatch usando o console.

Definição de preço

Com o Monitor de Internet do Amazon CloudWatch, você paga apenas pelo que usa. O preço do Monitor de Internet tem dois componentes: uma taxa por recurso monitorado e uma taxa por cidade-rede. Uma rede urbana é o local de onde os clientes acessam os recursos da sua aplicação e a rede (um ASN, como um provedor de serviços de Internet ou ISP) pela qual os clientes acessam os recursos.

Para obter mais informações, incluindo exemplos de preços, consulte Informações de preços do Monitor de Internet do Amazon CloudWatch..

Interromper o monitoramento de uma distribuição

Se quiser parar de monitorar recursos de sua distribuição com o Monitor de Internet, faça o seguinte no console do Monitor de Internet:

Remover um recurso de um monitor

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

2. No painel de navegação à esquerda, em Monitoramento de rede, escolha Monitor de Internet.
3. Escolha seu monitor e, em seguida, escolha o menu Ação.
4. Escolha Atualizar monitor.
5. Em Recursos adicionados, selecione Remover recursos.
6. Escolha a distribuição a ser removida e, em seguida, escolha Remover.
7. Escolha Atualizar.

Exemplos de uso da CLI com o Monitor de Internet do Amazon CloudWatch

Esta seção inclui exemplos de uso da AWS Command Line Interface com operações do Monitor de Internet do Amazon CloudWatch.

Antes de começar, certifique-se de fazer login para usar a AWS CLI com a mesma conta da AWS que tem as nuvens privadas virtuais (VPCs) da Amazon, os Network Load Balancers, as distribuições do Amazon CloudFront ou os diretórios do Amazon WorkSpaces que você deseja monitorar.
O Monitor de Internet não é compatível com o acesso a recursos em várias contas. Para obter informações sobre como usar a AWS CLI, consulte a Referência de comandos da AWS CLI. Para obter mais informações sobre o uso de ações de API com o Amazon Monitor de Internet do Amazon CloudWatch, consulte o Amazon CloudWatch Internet Monitor API Reference Guide (Guia de referência da API do Monitor de Internet do Amazon CloudWatch).

Tópicos

Criar um monitor

Ao criar um monitor no Monitor de Internet, você fornece um nome e associa recursos ao monitor para mostrar onde está o tráfego de Internet da aplicação. Você especifica um percentual de tráfego

que define quanto do tráfego da sua aplicação é monitorado. Isso também determina o número de cidades-redes, ou seja, locais de clientes e ASNs, geralmente provedores de serviços de Internet ou ISPs, que são monitorados. Também é possível optar por definir um limite para o número máximo de cidades-redes a serem monitoradas para os recursos da sua aplicação para ajudar a controlar suas despesas. Para ter mais informações, consulte Escolha de um limite máximo de cidades-redes.

Por fim, é possível escolher se deseja publicar todas as medições da Internet para sua aplicação no Amazon S3. As medições da Internet para as 500 principais cidades-redes (por volume de tráfego) são publicadas automaticamente no CloudWatch Logs pelo Monitor de Internet, mas também é possível optar por publicar todas as medições no S3.

Para criar um monitor com a AWS CLI, use o comando `create-monitor`. O comando a seguir cria um monitor que monitora 100% do tráfego, mas define um limite máximo de 10.000 cidades-redes, adiciona um recurso de VPC e opta por publicar medições da Internet no Amazon S3.

> **Note**
>
> O Monitor de Internet publica no CloudWatch Logs medições da Internet a cada cinco minutos para as 500 principais cidades-redes (locais de clientes e ASNs, geralmente provedores de serviços de Internet ou ISPs) que enviem tráfego para cada monitor. Opcionalmente, é possível optar por publicar medições da Internet para todas as cidades-redes monitoradas (até o limite de serviço de 500.000 cidades-redes) em um bucket do Amazon S3. Para ter mais informações, consulte Publicação de medições da Internet para o Amazon S3 no Monitor de Internet do Amazon CloudWatch.

```
aws internetmonitor --create-monitor monitor-name "TestMonitor" \
    --traffic-percentage-to-monitor 100 \
    --max-city-networks-to-monitor 10000 \
    --resources "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889" \
    --internet-measurements-log-delivery
 S3Config="{BucketName=MyS3Bucket,LogDeliveryStatus=ENABLED}"
```

```
{
"Arn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
 "Status": "ACTIVE"
}
```

> **Note**
>
> Não é possível alterar o nome de um monitor.

Visualizar detalhes do monitor

Para visualizar informações sobre um monitor com a AWS CLI, você usa o comando `get-monitor`.

```
aws internetmonitor get-monitor --monitor-name "TestMonitor"
```

```
{
    "ClientLocationType": "city",
    "CreatedAt": "2022-09-22T19:27:47Z",
    "ModifiedAt": "2022-09-22T19:28:30Z",
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "MonitorName": "TestMonitor",
    "ProcessingStatus": "OK",
    "ProcessingStatusInfo": "The monitor is actively processing data",
    "Resources": [
        "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889"
    ],
    "MaxCityNetworksToMonitor": 10000,
    "Status": "ACTIVE"
}
```

Listar eventos de integridade

Quando a performance do tráfego de Internet da aplicação se degrada, o Monitor de Internet cria eventos de integridade no monitor. Para ver uma lista dos eventos de integridade atuais com a AWS CLI, use o comando `list-health-events`

```
aws internetmonitor list-health-events --monitor-name "TestMonitor"
```

```
{
    "HealthEvents": [
        {
            "EventId": "2022-06-20T01-05-05Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:15:14Z",
```

```
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.21,
            "ClientLocations": [
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 60.370000000000005,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                }
            ],
            "StartedAt": "2022-06-20T01:05:05Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-05-05Z/latency"
        },
        {
            "EventId": "2022-06-20T01-17-56Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:30:23Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.29,
            "ClientLocations": [
                {
                    "City": "Toronto",
                    "PercentOfClientLocationImpacted": 75.32,
                    "PercentOfTotalTraffic": 1.05,
                    "Country": "Canada",
                    "Longitude": -79.3623,
                    "AutonomousSystemNumber": 14061,
                    "Latitude": 43.6547,
                    "Subdivision": "Ontario",
```

```
                    "CausedBy": {
                        "Status": "ACTIVE",
                        "Networks": [
                            {
                                "AutonomousSystemNumber": 16509,
                                "NetworkName": "Amazon.com"
                            }
                        ],
                        "NetworkEventType": "AWS"
                    },
                    "NetworkName": "DIGITALOCEAN-ASN"
                },
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 22.91,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                },
                {
                    "City": "Hangzhou",
                    "PercentOfClientLocationImpacted": 2.88,
                    "PercentOfTotalTraffic": 0.7799999999999999,
                    "Country": "China",
                    "Longitude": 120.1612,
                    "AutonomousSystemNumber": 37963,
                    "Latitude": 30.2994,
                    "Subdivision": "Zhejiang",
                    "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
                }
            ],
            "StartedAt": "2022-06-20T01:17:56Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-17-56Z/latency"
        },
        {
            "EventId": "2022-06-20T01-34-20Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:35:04Z",
```

```
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.15,
            "ClientLocations": [
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 39.45,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                },
                {
                    "City": "Toronto",
                    "PercentOfClientLocationImpacted": 29.770000000000003,
                    "PercentOfTotalTraffic": 1.05,
                    "Country": "Canada",
                    "Longitude": -79.3623,
                    "AutonomousSystemNumber": 14061,
                    "Latitude": 43.6547,
                    "Subdivision": "Ontario",
                    "CausedBy": {
                        "Status": "ACTIVE",
                        "Networks": [
                            {
                                "AutonomousSystemNumber": 16509,
                                "NetworkName": "Amazon.com"
                            }
                        ],
                        "NetworkEventType": "AWS"
                    },
                    "NetworkName": "DIGITALOCEAN-ASN"
                },
                {
                    "City": "Hangzhou",
                    "PercentOfClientLocationImpacted": 2.88,
                    "PercentOfTotalTraffic": 0.7799999999999999,
                    "Country": "China",
```

```
                    "Longitude": 120.1612,
                    "AutonomousSystemNumber": 37963,
                    "Latitude": 30.2994,
                    "Subdivision": "Zhejiang",
                    "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
                }
            ],
            "StartedAt": "2022-06-20T01:34:20Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-34-20Z/latency"
        }
    ]
}
```

Visualizar evento de integridade específico

Para ver informações mais detalhadas sobre um evento de integridade específico com a CLI, execute o comando `get-health-event` com o nome do monitor e o ID do evento de integridade.

```
aws internetmonitor get-monitor --monitor-name "TestMonitor" --event-id "health-event/
TestMonitor/2021-06-03T01:02:03Z/latency"
```

```
{
    "EventId": "2022-06-20T01-34-20Z/latency",
    "Status": "RESOLVED",
    "EndedAt": "2022-06-20T01:35:04Z",
    "ServiceLocations": [
        {
            "Name": "us-east-1"
        }
    ],
    "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor/
health-event/2022-06-20T01-34-20Z/latency",
    "LastUpdatedAt": "2022-06-20T01:35:04Z",
    "ClientLocations": [
        {
            "City": "Lockport",
            "PercentOfClientLocationImpacted": 39.45,
            "PercentOfTotalTraffic": 2.01,
            "Country": "United States",
            "Longitude": -78.6913,
            "AutonomousSystemNumber": 26101,
```

```
            "Latitude": 43.1721,
            "Subdivision": "New York",
            "NetworkName": "YAHOO-BF1"
        },
        {
            "City": "Toronto",
            "PercentOfClientLocationImpacted": 29.770000000000003,
            "PercentOfTotalTraffic": 1.05,
            "Country": "Canada",
            "Longitude": -79.3623,
            "AutonomousSystemNumber": 14061,
            "Latitude": 43.6547,
            "Subdivision": "Ontario",
            "CausedBy": {
                "Status": "ACTIVE",
                "Networks": [
                    {
                        "AutonomousSystemNumber": 16509,
                        "NetworkName": "Amazon.com"
                    }
                ],
                "NetworkEventType": "AWS"
            },
            "NetworkName": "DIGITALOCEAN-ASN"
        },
        {
            "City": "Shenzhen",
            "PercentOfClientLocationImpacted": 4.07,
            "PercentOfTotalTraffic": 0.61,
            "Country": "China",
            "Longitude": 114.0683,
            "AutonomousSystemNumber": 37963,
            "Latitude": 22.5455,
            "Subdivision": "Guangdong",
            "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
        },
        {
            "City": "Hangzhou",
            "PercentOfClientLocationImpacted": 2.88,
            "PercentOfTotalTraffic": 0.7799999999999999,
            "Country": "China",
            "Longitude": 120.1612,
            "AutonomousSystemNumber": 37963,
            "Latitude": 30.2994,
```

```
            "Subdivision": "Zhejiang",
            "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
        }
    ],
    "StartedAt": "2022-06-20T01:34:20Z",
    "ImpactType": "PERFORMANCE",
    "PercentOfTotalTrafficImpacted": 1.15
}
```

Visualizar lista de monitores

Para ver uma lista de todos os monitores em sua conta com a CLI, execute o comando `list-monitors`.

```
aws internetmonitor list-monitors
```

```
{
    "Monitors": [
        {
            "MonitorName": "TestMonitor",
            "ProcessingStatus": "OK",
            "Status": "ACTIVE"
        }
    ],
    "NextToken": " zase12"
}
```

Editar monitor

Para atualizar as informações sobre seu monitor usando a CLI, use o comando `update-monitor` e especifique o nome do monitor a ser atualizado. É possível atualizar o percentual de tráfego a ser monitorado, o limite do número máximo de cidades-redes a serem monitoradas, adicionar ou remover os recursos que o Monitor de Internet usa para monitorar o tráfego e alterar o status do monitor de `ACTIVE` para `INACTIVE` ou vice-versa. Observe que não é possível alterar o nome do monitor.

A resposta para uma chamada a `update-monitor` retorna apenas o `MonitorArn` e o `Status`.

O exemplo a seguir mostra como usar o comando `update-monitor` para alterar o número máximo de cidades-rede a monitorar para `50000`:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" --max-city-networks-to-
monitor 50000
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": " ACTIVE "
}
```

O exemplo a seguir mostra como adicionar e remover um recurso:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" \
    --resources-to-add "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889" \
    --resources-to-remove "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-2222444455556666"
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": "ACTIVE"
}
```

O exemplo a seguir mostra como usar o comando `update-monitor` para alterar o status do monitor para `INACTIVE`:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" --status "INACTIVE"
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": "INACTIVE"
}
```

Excluir um monitor

Você pode excluir um monitor com a CLI usando o comando `delete-monitor`. Primeiro, você deve configurar o monitor para ficar inativo. Para fazer isso, use o comando `update-monitor` para alterar o status para `INACTIVE`. Confirme se o monitor está inativo usando o comando `get-monitor` e verificando o status.

Quando o status do monitor for `INACTIVE`, você poderá usar a CLI para executar o comando `delete-monitor` para excluir o monitor. A resposta para uma chamada `delete-monitor` bem-sucedida é vazia.

```
aws internetmonitor delete-monitor --monitor-name "TestMonitor"
```

```
{}
```

Monitorar e otimizar com o painel do Monitor de Internet

As informações nesta seção descrevem como filtrar e visualizar informações no painel do Monitor de Internet do Amazon CloudWatch para visualizar e obter insights sobre o tráfego e a configuração de Internet da sua aplicação da AWS.

Depois de criar um monitor para monitorar a performance e a disponibilidade da aplicação na Internet, o Monitor de Internet do Amazon CloudWatch publica logs do CloudWatch contendo medições de Internet para os pares local-rede (cidade-rede) do cliente, e publica métricas agregadas do CloudWatch do tráfego para a aplicação, e para cada região e cada local da borda da Região da AWS. Você pode filtrar, explorar e obter sugestões orientadas à ação baseadas nessas informações do Monitor de Internet de várias maneiras diferentes.

Para começar, no console do CloudWatch, em Monitoramento de rede, escolha Monitor da Internet.

Esta seção descreve principalmente como filtrar e visualizar as métricas do Monitor de Internet usando o AWS Management Console. Como alternativa, é possível usar as operações da API do Monitor de Internet com a AWS CLI ou um SDK para trabalhar diretamente com eventos do Monitor de Internet armazenados nos arquivos do CloudWatch Logs. Para obter mais informações, consulte Uso do seu monitor e informações de medidas. Para obter mais informações sobre o uso de operações de API, consulte Exemplos de uso da CLI com o Monitor de Internet do Amazon CloudWatch e a Referência da API do Monitor de Internet do Amazon CloudWatch.

Há três guias no painel do Monitor de Internet:

- Na guia Overview (Visão geral), você pode ver informações atuais e históricas de performance e disponibilidade da aplicação, e os eventos de integridade que afetam os locais dos clientes.
- Na próxima guia, Explorador de histórico, é possível filtrar por local, ASN, data e assim por diante, e visualizar métricas de tráfego da Internet ao longo do tempo usando os gráficos.
- Na guia Insights de tráfego, além de visualizar as informações sobre o tráfego principal monitorado resumidas de várias maneiras personalizáveis, é possível obter sugestões de configurações otimizadas para melhorar a performance para diferentes pares de local e ASN. O Monitor de Internet prevê qual será o ganho de performance da aplicação, com base nos padrões de tráfego

e na performance anterior, quando você alterar a maneira de direcionar tráfego ou os recursos da AWS que você usa. Também é possível ver um gráfico para comparar quantas cidades-redes estão incluídas em sua cobertura de monitoramento, com base no percentual de tráfego da aplicação que você escolher para seu monitor.

Além disso, como o Monitor de Internet gera e publica arquivos de log com as medições de tráfego, é possível usar outras ferramentas do CloudWatch no console para visualizar mais os dados publicados pelo Monitor de Internet, incluindo o CloudWatch Contributor Insights, o CloudWatch Metrics e o CloudWatch Logs Insights. Para ter mais informações, consulte Como explorar seus dados com as ferramentas do CloudWatch e a interface de consulta do Monitor de Internet.

Saiba mais sobre o uso do Monitor de Internet para explorar as medições de performance e de disponibilidade nas seções a seguir.

Tópicos

- Monitoramento dar performance e disponibilidade em tempo real no Monitor de Internet do Amazon CloudWatch (guia Overview [Visão geral])
- Filtragem e visualização de dados no Monitor de Internet do Amazon CloudWatch (guia Explorador de histórico)
- Obter informações para melhorar a performance da aplicação no Monitor de Internet do Amazon CloudWatch (guia Insights de tráfego)

Monitoramento dar performance e disponibilidade em tempo real no Monitor de Internet do Amazon CloudWatch (guia Overview [Visão geral])

Use a guia Visão geral no console do CloudWatch, no Monitor de Internet, para obter uma visualização de alto nível da performance e da disponibilidade do tráfego que o seu monitor rastreia. A guia também exibe um mapa geral do tráfego da Internet, com clusters de tráfego que podem ajudar a visualizar o tráfego global da aplicação, e o local e o impacto dos eventos de integridade.

Pontuações de integridade

O gráfico Pontuações de integridade mostra informações de performance e de disponibilidade para o tráfego global. A AWS tem dados históricos substanciais sobre a performance e a disponibilidade de Internet para tráfego de rede entre áreas geográficas de diferentes ASNs e serviços da AWS. O Monitor de Internet usa os dados de conectividade que a AWS captura de sua rede global para calcular uma linha de base de performance e de disponibilidade para o

tráfego de Internet. Esses são os mesmos dados que usamos na AWS para monitorar o nosso próprio tempo de atividade e disponibilidade de Internet.

Com essas medições como linha de base, o Monitor de Internet pode detectar quando a performance e a disponibilidade da aplicação diminuíram, em comparação com a linha de base. Para facilitar a visualização dessas quedas, fornecemos essas informações a você como uma pontuação de performance e uma pontuação de disponibilidade. Para ter mais informações, consulte Como explorar seus dados com as ferramentas do CloudWatch e a interface de consulta do Monitor de Internet.

O gráfico Pontuações de integridade inclui os eventos de integridade que ocorreram durante o período de tempo que você escolher. Quando há um evento de integridade, você vê uma queda na linha de performance ou de disponibilidade no gráfico. Se você selecionar o evento, verá mais detalhes, e o gráfico exibirá faixas com informações de data e hora, mostrando quanto tempo o evento durou.

Você também pode analisar essas métricas acessando diretamente os arquivos de log para cada ponto de dados. No menu Actions (Ações), escolha View CloudWatch Logs (Visualizar logs do CloudWatch).

Visão geral do tráfego da Internet

O mapa Visão geral do tráfego da Internet mostra o tráfego da Internet e os eventos de integridade específicos dos locais e dos ASNs de onde seus usuários acessam a sua aplicação. Os países que estão em cinza no mapa são os que incluem tráfego para a sua aplicação.

Cada círculo no mapa indica um evento de integridade em uma área, para um período de tempo selecionado por você. O Monitor de Internet cria eventos de integridade quando detecta um problema, em um limite específico, com a conectividade entre um dos seus recursos hospedados na AWS e uma cidade-rede em que um usuário está acessando a aplicação. Escolher um círculo no mapa exibe mais detalhes sobre o evento de integridade daquele local. Além disso, para clusters que têm eventos de integridade, você pode ver informações detalhadas na tabela de Health events (Eventos de integridade) abaixo do mapa.

Observe que o Monitor de Internet cria eventos de integridade em um monitor quando determina que um evento tem um impacto global significativo na sua aplicação. Se não houve nenhum evento de integridade que tenha excedido o limite de impacto no tráfego para os locais de seus clientes no período selecionado, o mapa ficará em branco. Para obter mais informações, consulte Quando o Monitor de Internet cria e resolve eventos de integridade.

Alterar limites de eventos de integridade

É possível configurar várias opções sobre como e quando o Monitor de Internet cria eventos de integridade para sua aplicação. Escolha Atualizar limites para fazer alterações.

É possível alterar o limite geral que aciona o Monitor de Internet para criar um evento de integridade. O limite de evento de integridade padrão, tanto para pontuações de performance quanto de disponibilidade, é de 95%. Ou seja, quando a pontuação geral de performance ou disponibilidade da sua aplicação cai para 95% ou menos, o Monitor de Internet cria um evento de integridade. Para o limite geral, o evento de integridade pode ser acionado por um único grande problema ou pela combinação de vários problemas menores.

Também é possível alterar o limite local, ou seja, a cidade-rede, combinado com um percentual do nível geral de impacto que, combinado, desencadeará um evento de intregridade. Ao definir um limite que cria um evento de integridade quando uma pontuação cai abaixo do limite para uma ou mais cidades-redes (locais e ASNs, geralmente ISPs, é possível obter insights sobre quando há problemas em locais com tráfego mais baixo, por exemplo.

Uma opção adicional de limite local funciona em conjunto com o limite local para pontuações de disponibilidade ou performance. O segundo fator é o percentual do tráfego geral que deve ser afetado antes que o Monitor de Internet crie um evento de integridade com base no limite local.

Ao configurar as opções de limite para o tráfego geral e o tráfego local, é possível ajustar a frequência com que os eventos de integridade são criados, de acordo com o uso da sua aplicação e suas necessidades. Lembre-se de que, quando você define o limite local como menor, normalmente são criados mais eventos de integridade, dependendo da sua aplicação e dos outros valores de configuração de limite que você definir.

Em resumo, é possível configurar limites de eventos de integridade (para pontuações de performance, pontuações de disponibilidade ou ambas) das maneiras a seguir:

- Escolha limites globais diferentes para acionar um evento de integridade.
- Escolha limites locais diferentes para acionar um evento de integridade. Com esta opção, também é possível alterar o percentual de impacto geral na sua aplicação que deve ser excedido antes que o Monitor de Internet crie um evento.
- Escolha desativar o acionamento de um evento de integridade com base nos limites locais ou ativar as opções de limite local.

Também é possível configurar opções para pontuações de performance, pontuações de disponibilidade ou ambas. É possível configurar uma combinação das opções ou apenas uma delas.

Para atualizar os limites e outras opções de configuração para pontuações de performance, pontuações de disponibilidade ou ambas, faça o seguinte:

Para alterar as opções de configuração de limite

1. Em AWS Management Console, navegue até CloudWatch e, em seguida, no painel de navegação esquerdo, escolha Monitor de Internet.
2. Na guia Visão geral, na seção Cronograma de eventos de integridade, escolha Atualizar limites.
3. Na página de diálogo que se abre, escolha os novos valores e opções que você deseja para limites e outras opções que acionam o Monitor de Internet para criar um evento de integridade. Você pode realizar uma das seguintes ações:
   - Escolha um novo valor para Limite de pontuação de disponibilidade, Limite de pontuação de performance ou ambos.

     Os gráficos nas seções de cada configuração exibem a configuração de limite atual e as pontuações reais de eventos de integridade recentes, para disponibilidade ou performance, da sua aplicação. Ao visualizar os valores típicos, será possível ter uma ideia dos valores para os quais talvez queira alterar um limite.

     Dica: para ver um gráfico maior e alterar o período de tempo, escolha o expansor no canto superior direito do gráfico.
   - Escolha ativar ou desativar um limite local de disponibilidade ou performance, ou ambos. Quando uma opção está ativada, é possível definir o limite e o nível de impacto para quando quiser que o Monitor de Internet crie um evento de integridade.
4. Depois de configurar as opções de limite, salve suas atualizações escolhendo Atualizar limites de eventos de integridade.

Para saber mais sobre como os eventos de integridade funcionam, consulte Quando o Monitor de Internet cria e resolve eventos de integridade.

Tabela de eventos de integridade

A tabela de Eventos de integridade lista os locais dos clientes que foram afetados por eventos de integridade, junto com as informações sobre esses eventos. A tabela inclui as colunas a seguir.

	Descrição
Client location (Localização do cliente)	O local dos usuários finais que foram afetados pelo evento e experimentaram aumento de latência ou queda de disponibilidade. Para saber mais sobre a precisão da localização do cliente no Monitor de Internet, consulte Informações e precisão de geolocalização no Monitor de Internet.
Traffic impact (Impacto do tráfego)	Quanto impacto o evento causou em termos de aumento de latência ou queda de disponibilidade. Para latência, essa é a porcentagem de aumento de latência durante o evento em comparação com a performance típica do tráfego, do local do cliente até o local da AWS usando essa rede do cliente.
Client network (Rede do cliente)	A rede que o tráfego percorreu. Normalmente, esse é o provedor de serviços de Internet (ISP) ou o número de sistema autônomo (ASN) para o tráfego da rede.
Local do AWS	O local da AWS para o tráfego de rede, que pode ser uma Região da AWS ou um local da borda.
Impact type (Tipo de impacto)	O tipo de impacto do evento de integridade. Os eventos de integridade geralmente são causados por aumentos de latência (problemas de performance) ou por acessibilidade (problemas de disponibilidade).

	Descrição
	Você também pode clicar no tipo de impacto para ver a causa da deficiência. Quando possível, o Monitor de Internet analisa a origem de um evento de integridade para determinar se ele foi causado pela AWS ou por um ASN (provedor de serviços de Internet). Observe que essa análise continua após a resolução do evento. O Monitor de Internet pode atualizar eventos com novas informaçõ es por até uma hora.

Se você escolher um dos locais de cliente na tabela Eventos de integridade, poderá ver mais detalhes sobre o evento de integridade naquele local. Por exemplo, você pode ver quando o evento começou, quando terminou e o impacto no tráfego local.

Visualização do caminho da rede

A análise de deficiência concluída tem um caminho de rede completo em Visualização do caminho de rede. O Caminho completo mostra cada nó ao longo do caminho de rede percorrido pela sua aplicação para o evento de integridade, entre o local da AWS e o cliente, para um par cliente-local.

Se o Monitor de Internet determinar a causa de uma deficiência, ela será marcada com um círculo vermelho tracejado. As deficiências podem ser causadas por ASNs, geralmente provedores de serviços de Internet (ISPs), ou a causa pode ser a AWS. Se houver várias causas para uma deficiência, vários nós serão marcados.

Filtragem e visualização de dados no Monitor de Internet do Amazon CloudWatch (guia Explorador de histórico)

Use a guia Explorador de histórico no console do CloudWatch, em Monitor de Internet, para filtrar e visualizar dados da sua aplicação que estejam no CloudWatch Logs. O Monitor de Internet publica medições no CloudWatch Logs específicas para a sua aplicação para verificar a disponibilidade, a

performance, os bytes monitorados transferidos (ou o número de conexões de clientes, somente para os diretórios do WorkSpaces) e o tempo de ida e volta para as cidades-redes monitoradas nas Regiões da AWS.

> **Note**
>
> O Monitor de Internet publica medições de Internet no CloudWatch Logs a cada cinco minutos para as 500 principais (por volume de tráfego) cidades-redes (ou seja, locais de clientes e ASNs, geralmente provedores de serviços de Internet ou ISPs) que enviem tráfego para cada monitor. Opcionalmente, é possível optar por publicar medições da Internet para todas as cidades-redes monitoradas (até o limite de serviço de 500.000 cidades-redes) em um bucket do Amazon S3. Para ter mais informações, consulte Publicação de medições da Internet para o Amazon S3 no Monitor de Internet do Amazon CloudWatch.

Para começar a explorar os dados da sua aplicação, selecione um período de tempo. Em seguida, escolha uma localização geográfica específica, como uma cidade, e (opcionalmente) outros filtros. O Monitor de Internet aplica os filtros aos dados nos logs de medições de Internet publicados para as cidades-redes para o tráfego da sua aplicação. Em seguida, será possível ver gráficos dos dados mostrando a pontuação de performance, a pontuação de disponibilidade, os bytes monitorados transferidos (para VPCs, Network Load Balancers e distribuições do CloudFront) ou as contagens de conexões do cliente (para diretórios do WorkSpaces) e o tempo de ida e volta (RTT) para a aplicação ao longo do tempo.

A tabela All events (Todos os eventos) abaixo dos gráficos mostra os eventos de integridade que o filtro retorna para o tráfego da aplicação, com informações sobre cada evento. Isso inclui as colunas a seguir.

	Descrição
Event start (Início do evento)	A hora em que o evento de integridade começou.
Status	Se o evento ainda está ativo ou foi resolvido.
Client location (Localização do cliente)	O local dos usuários finais que foram afetados pelo evento, que sofreram aumento de latência ou queda de performance.

	Descrição
	Para saber mais sobre a precisão da localizaç ão do cliente no Monitor de Internet, consulte Informações e precisão de geolocalização no Monitor de Internet.
Traffic impact (Impacto do tráfego)	O impacto ponderado do evento no local do evento de integridade. Ou seja, por exemplo, o impacto na latência, em comparação com a performance típica de tráfego de um local do cliente até o local da AWS através do ASN do cliente, geralmente um provedor de serviços de Internet (ISP). Da mesma forma, para um evento que afete a disponibilidade, você vê o impacto na disponibilidade em comparaçã o com a disponibilidade típica do local do cliente para o local da AWS através do ASN do cliente.
Event duration (Duração do evento)	Quanto tempo durou o evento. O Monitor de Internet encerra os eventos de integridade quando eles não afetam mais de 5% (no total) dos locais de clientes da aplicação.
Client ISP (ISP do cliente)	O ASN, geralmente o provedor de serviços de Internet (ISP), que foi a operadora do tráfego da rede.
Service location (Local do serviço)	O local do serviço de origem do tráfego da rede, que pode ser uma Região da AWS ou um local da borda da Internet.

Como alternativa, é possível ver as medições da sua aplicação acessando os logs diretamente de cada ponto de dados. No menu Actions (Ações), escolha View CloudWatch Logs (Visualizar logs do CloudWatch). Observe que como os eventos de medição são publicados em sua conta quando são criados, também é possível criar outros painéis ou alarmes do CloudWatch baseados neles. Para

obter mais informações, consulte Obter informações para melhorar a performance da aplicação no Monitor de Internet do Amazon CloudWatch (guia Insights de tráfego) e Criação de alarmes com o Monitor de Internet do Amazon CloudWatch.

Além de explorar e analisar medições e métricas do Monitor de Internet e criar painéis e alarmes baseados nelas, é possível usar o Monitor de Internet para ajudar a entender como poderia melhorar a performance da sua aplicação. A guia Traffic insights (Insights sobre tráfego) inclui várias maneiras de ajudar você a explorar as opções. Para obter mais informações, consulte Sugestões de otimização de tráfego na guia Insights de tráfego. Além disso, é possível ver os exemplos específicos no capítulo Casos de uso do Monitor de Internet.

Obter informações para melhorar a performance da aplicação no Monitor de Internet do Amazon CloudWatch (guia Insights de tráfego)

Use a guia Insights de tráfego no console do CloudWatch, no Monitor de Internet, para ver informações resumidas do tráfego principal (por volume) da sua aplicação. É possível filtrar e classificar o tráfego da sua aplicação de várias maneiras. Em seguida, role para baixo e selecione diferentes combinações de configurações para sua aplicação, para ver o que o Monitor de Internet sugere como melhores alternativas para obter a performance mais rápida de tempo até o primeiro byte (TTFB).

O Monitor de Internet publica no CloudWatch Logs medições da Internet a cada cinco minutos para as 500 principais (por volume de tráfego) cidades-redes (ou seja, localizações de clientes e ASNs, geralmente provedores de serviços de Internet ou ISPs) que enviam tráfego para cada monitor. Opcionalmente, é possível optar por publicar medições da Internet para todas as cidades-redes monitoradas (até o limite de serviço de 500.000 cidades-redes) em um bucket do Amazon S3. Para ter mais informações, consulte Publicação de medições da Internet para o Amazon S3 no Monitor de Internet do Amazon CloudWatch.

Principais resumos de tráfego

> É possível começar visualizando resumos de alto nível do tráfego e da performance geral da sua aplicação durante um período de tempo específico, filtrado por local do cliente. Também é possível analisar a performance da sua aplicação nas principais (ou últimas) localizações do cliente por volume de tráfego, filtrado e classificado de várias formas. Por exemplo, é possível classificar por granularidade (ou seja, cidade, subdivisão, país ou área metropolitana), por tráfego total, tempo médio até o primeiro byte (TTFB) e outros fatores.

Para saber mais sobre a precisão da localização do cliente no Monitor de Internet, consulte Informações e precisão de geolocalização no Monitor de Internet.

> **Note**
>
> Os filtros que você usa se aplicam à página inteira, portanto, afetam as cidades-redes que estiverem incluídas nos gráficos de resumo e nas informações do tráfego total, bem como as cidades-redes que estiverem incluídas na seção Sugestões de otimização de tráfego a seguir.

Sugestões para otimização de tráfego

A seção Sugestões de otimização de tráfego exibe um conjunto filtrado de cidades-redes monitoradas (locais e ASNs, provedores de serviços de Internet) para seu tráfego, junto com o tráfego total de clientes de cada uma. As entradas na tabela são baseadas nos filtros que você escolheu para o tráfego da sua aplicação para Insights de tráfego na parte superior da página. O padrão são as 10 principais cidades por volume de tráfego. Normalmente, você vê mais de 10 linhas na tabela, porque há uma entrada para cada par cidade-rede único. Ou seja, há uma linha para cada combinação de localização (cidade) e ASN (provedor de rede) por meio da qual os clientes acessam sua aplicação, como Dallas, Texas, EUA e Comcast, por exemplo.

> **Note**
>
> Para ver sugestões de otimização de tráfego para todas as suas cidadades-redes monitoradas, é possível executar uma consulta diretamente ao CloudWatch Insights. Para ver um exemplo de consulta que não inclua o filtro de granularidade geográfica que limita a lista de cidades-redes nesta página, consulte Usar o CloudWatch Logs Insights com o Monitor de Internet do Amazon CloudWatch.

Nesta seção, selecione diferentes opções: Amazon EC2, CloudFront ou ambas. Isso permite que você veja quais são os valores previstos de tempo médio até o primeiro byte (TTFB) para clientes quando você usa sua aplicação com esses serviços em diferentes regiões da AWS, em comparação com o TTFB atual. Para obter mais informações sobre cálculos de TTFB, consulte Cálculos de TTFB e latência da AWS.

Ao selecionar diferentes opções e, em seguida, visualizar os resultados na tabela, é possível começar a planejar configurações e implantações que possam melhorar a performance dos

seus clientes. Observe que você pode ver um traço (-), em vez de um valor em uma coluna, quando os dados não estão disponíveis para exibição. Para revisar um exemplo específico de como melhorar a performance, consulte Using Amazon CloudWatch Internet Monitor for a Better Gaming Experience.

Por exemplo, para começar, para uma cidade-rede específica (par de localização do cliente e ASN), experimente selecionar a opção EC2 ou CloudFront, ou ambas. Para cada cidade-rede listada na tabela, o Monitor de Internet mostra as possíveis melhorias de performance de TTFB, com base em uma opção de roteamento de tráfego (por meio de uma Região da AWS específica) com essa opção, em comparação com a configuração atual. (Observe que, por completude, a tabela também inclui rotas que já estão otimizadas.) Por exemplo, seria possível ver um TTFB médio previsto de 50 ms usando o EC2 e roteando por `us-east-1` em comparação com a configuração atual com um TTFB de 100 ms, na qual você está usando o EC2 e roteando por `us-west-2`. Portanto, é possível considerar o roteamento por `us-west-2`.

Como outro exemplo, é possível selecionar o EC2 e, em seguida, ver que isso não faz uma diferença mensurável de performance para uma localização de cliente e um ASN, mas, em seguida, observar que, ao selecionar o CloudFront com a mesma região, o TTFB diminui um pouco. Isso sugere que se você adicionar uma distribuição do CloudFront na frente da sua aplicação, isso pode resultar em uma melhoria de performance e poderia valer a pena ser tentado, para este local e ASN do cliente.

Como explorar seus dados com as ferramentas do CloudWatch e a interface de consulta do Monitor de Internet

Além de visualizar a performance e a disponibilidade da sua aplicação com o painel do Monitor de Internet do Amazon CloudWatch, há vários métodos que você pode usar para se aprofundar nos dados que o Monitor de Internet gera para você. Esses métodos incluem o uso de ferramentas do CloudWatch com dados do Monitor de Internet armazenados nos arquivos de log do CloudWatch e o uso da interface de consulta do Monitor de Internet. As ferramentas que você pode usar incluem o CloudWatch Logs Insights, o CloudWatch Metrics, o CloudWatch Contributor Insights e o Amazon Athena. Você pode usar algumas ou todas essas ferramentas, bem como o painel, para explorar os dados do Monitor de Internet, dependendo de suas necessidades.

O Monitor de Internet agrega as métricas do CloudWatch sobre o tráfego para a sua aplicação e para cada Região da AWS, e inclui dados como o impacto total do tráfego, a disponibilidade e o tempo de ida e volta. Esses dados são publicados no CloudWatch Logs e também estão disponíveis para uso

com a interface de consulta do Monitor de Internet. Os detalhes sobre a geo-granularidade e outros aspectos das informações disponíveis para exploração variam para cada um deles.

O Monitor de Internet do Amazon CloudWatch publica os dados do seu monitor em intervalos de 5 minutos e, em seguida, disponibiliza os dados de várias maneiras. A tabela a seguir lista os cenários de acesso aos dados do Monitor de Internet e descreve os recursos dos dados coletados para cada um deles.

Atributo	CloudWatch Logs	Exportar para S3	Interface de consulta	Painéis do CloudWatch
Habilitada por padrão.	Sim	Não	Sim	Sim
Número de cidades-redes para as quais os dados são coletados	500 principais (consulte a nota abaixo)	Todos	Todos	Todos
Retenção de dados	Controlado pelo usuário	Controlado pelo usuário	30 dias	30 dias
Geo-granularidades para as quais os dados são coletados	Todos (cidade-rede, metro+rede, subdivisão+rede, país+rede)	Cidade-rede	Todos (cidade-rede, metro+rede, subdivisão+rede, país+rede)	Todos (cidade-rede, metro+rede, subdivisão+rede, país+rede)
Como consultar e filtrar dados	Usar o CloudWatch Logs Insights com o Monitor de Internet do Amazon CloudWatch	Uso do Amazon Athena para consultar medições de Internet nos arquivos de log do Amazon S3	Como usar a interface de consulta do Monitor de Internet do Amazon CloudWatch	Monitorar e otimizar com o painel do Monitor de Internet

Observação: as 500 principais medições são capturadas para cidades-redes; as 250 principais para metro+redes, as 100 principais para subdivisões+redes, as 50 principais para países+redes.

Este capítulo descreve como consultar e explorar seus dados usando as ferramentas do CloudWatch ou a interface de consulta do Monitor de Internet, juntamente com exemplos de cada método.

Conteúdo

Usar o CloudWatch Logs Insights com o Monitor de Internet do Amazon CloudWatch

O Monitor de Internet do Amazon CloudWatch publica medições granulares de disponibilidade e tempo de ida e volta no CloudWatch Logs, e é possível usar as consultas do CloudWatch Logs Insights para filtrar um subconjunto de logs para uma cidade ou área geográfica específica (localização do cliente), ASN (ISP) do cliente e local de origem na AWS.

Para saber mais sobre a precisão da localização do cliente no Monitor de Internet, consulte Informações e precisão de geolocalização no Monitor de Internet.

Os exemplos nesta seção podem ajudar a criar consultas do CloudWatch Logs Insights para saber mais sobre suas próprias medidas e métricas de tráfego de aplicações. Se você usar esses exemplos no CloudWatch Logs Insights, substitua *monitorName* pelo seu próprio nome de monitor.

Ver sugestões para otimização de tráfego

Na guia Insights de tráfego, no Monitor de Internet, é possível ver sugestões de otimização de tráfego, filtradas por um local. Para ver as mesmas informações exibidas na seção Sugestões de otimização de tráfego nessa guia, mas sem o filtro de granularidade de localização, é possível usar a consulta a seguir do CloudWatch Logs Insights.

1. No AWS Management Console, navegue até o CloudWatch Logs Insights.
2. Em Log Group (Grupo de logs), selecione `/aws/internet-monitor/`*monitorName*`/byCity` e `/aws/internet-monitor/`*monitorName*`/byCountry`, depois, especifique um intervalo de tempo.
3. Adicione e execute a consulta a seguir.

```
fields @timestamp,
clientLocation.city as @city, clientLocation.subdivision as @subdivision,
 clientLocation.country as @country,
`trafficInsights.timeToFirstByte.currentExperience.serviceName` as @serviceNameField,
concat(@serviceNameField, ` (`, `serviceLocation`, `)`) as @currentExperienceField,
concat(`trafficInsights.timeToFirstByte.ec2.serviceName`, ` (`,
 `trafficInsights.timeToFirstByte.ec2.serviceLocation`, `)`) as @ec2Field,
`trafficInsights.timeToFirstByte.cloudfront.serviceName` as @cloudfrontField,
concat(`clientLocation.networkName`, ` (AS`, `clientLocation.asn`, `)`) as @networkName
| filter ispresent(`trafficInsights.timeToFirstByte.currentExperience.value`)
| stats avg(`trafficInsights.timeToFirstByte.currentExperience.value`) as @averageTTFB,
avg(`trafficInsights.timeToFirstByte.ec2.value`) as @ec2TTFB,
avg(`trafficInsights.timeToFirstByte.cloudfront.value`) as @cloudfrontTTFB,
sum(`bytesIn` + `bytesOut`) as @totalBytes,
latest(@ec2Field) as @ec2,
latest(@currentExperienceField) as @currentExperience,
latest(@cloudfrontField) as @cloudfront,
count(*) by @networkName, @city, @subdivision, @country
| display @city, @subdivision, @country, @networkName, @totalBytes, @currentExperience,
 @averageTTFB, @ec2, @ec2TTFB, @cloudfront, @cloudfrontTTFB
| sort @totalBytes desc
```

Ver a disponibilidade da Internet e o RTT (p50, p90 e p95)

Para ver a disponibilidade da Internet e o tempo de ida e volta (p50, p90 e p95) do tráfego, é possível usar a consulta a seguir do CloudWatch Logs Insights.

Área geográfica do usuário final: Chicago, IL, Estados Unidos

Rede de usuário final (ASN): AS7018

Local do serviço da AWS: Região Leste dos EUA (N. da Virgínia)

Para obter os logs, faça o seguinte:

1. No AWS Management Console, navegue até o CloudWatch Logs Insights.
2. Em Log Group (Grupo de logs), selecione `/aws/internet-monitor/`*`monitorName`*`/byCity` e `/aws/internet-monitor/`*`monitorName`*`/byCountry`, depois, especifique um intervalo de tempo.
3. Adicione e execute a consulta a seguir.

A consulta retorna todos os dados de performance dos usuários conectados a partir de AS7018 em Chicago, IL, seguindo em direção à região Leste dos EUA (Norte da Virgínia) durante o período selecionado.

```
fields @timestamp,
internetHealth.availability.experienceScore as availabilityExperienceScore,
internetHealth.availability.percentageOfTotalTrafficImpacted as
 percentageOfTotalTrafficImpacted,
internetHealth.performance.experienceScore as performanceExperienceScore,
internetHealth.performance.roundTripTime.p50 as roundTripTimep50,
internetHealth.performance.roundTripTime.p90 as roundTripTimep90,
internetHealth.performance.roundTripTime.p95 as roundTripTimep95
 | filter clientLocation.country == `United States`
 and clientLocation.city == `Chicago`
 and serviceLocation == `us-east-1`
 and clientLocation.asn == 7018
```

Para obter mais informações, consulte Analisar logs de dados com o CloudWatch Logs Insights.

Usar Contributor Insights com o Monitor de Internet do Amazon CloudWatch

O CloudWatch Contributor Insights pode ajudar a identificar os principais locais e redes de clientes (ASNs ou provedores de serviços de Internet) da sua aplicação. Use o exemplo de regras do Contributor Insights a seguir para começar a usar regras que são úteis com o Monitor de Internet do Amazon CloudWatch. Para ter mais informações, consulte Criar uma regra do Contributor Insights.

Para saber mais sobre a precisão da localização do cliente no Monitor de Internet, consulte Informações e precisão de geolocalização no Monitor de Internet.

> **Note**
>
> O Monitor de Internet publica dados a cada cinco minutos, portanto, depois de configurar uma regra do Contributor Insights, você deve ajustar o período para cinco minutos para ver um gráfico.

Ver os principais locais e ASNs afetados por um impacto de disponibilidade

Para ver os principais locais e ASNs de clientes afetados por uma queda na disponibilidade, é possível usar a regra a seguir do Contributor Insights no editor de sintaxe. Substitua *monitor-name* pelo nome do seu próprio monitor.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.awsInternetHealth.availability.percentageOfTotalTrafficImpacted"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Ver os principais locais e ASNs de clientes afetados por um impacto de latência

Para ver os principais locais e ASNs de clientes afetados por um aumento de tempo de ida e volta (latência), é possível usar a regra a seguir do Contributor Insights no editor de sintaxe. Substitua *monitor-name* pelo nome do seu próprio monitor.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
```

```
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.awsInternetHealth.performance.percentageOfTotalTrafficImpacted"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Ver os principais locais e ASNs de clientes afetados pelo percentual total de tráfego

Para ver os principais locais e ASNs de clientes afetados pelo percentual total de tráfego, é possível usar a regra a seguir do Contributor Insights no editor de sintaxe. Substitua *monitor-name* pelo nome do seu próprio monitor.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.percentageOfTotalTraffic"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Usar o CloudWatch Logs Metrics com o Monitor de Internet do Amazon CloudWatch

O Monitor de Internet do Amazon CloudWatch publica métricas para a sua conta, incluindo métricas de performance, disponibilidade, ida e volta e throughput (bytes por segundo), que podem ser vistas no CloudWatch Metrics, no console do CloudWatch. Para localizar todas as métricas para seu monitor, no painel de métricas do CloudWatchconsulte o namespace personalizado `AWS/InternetMonitor`.

As métricas são agregadas ao longo de todo o tráfego da Internet até as suas VPCs, Network Load Balancers, distribuições do CloudFront ou diretórios do WorkSpaces no monitor, e para todo o tráfego para cada Região da AWS e cada local da borda de Internet que seja monitorado. As regiões são definidas pelo local do serviço, podendo ser todos os locais ou uma região específica, como a `us-east-1`.

Observação: as cidades-redes são locais e ASNs (geralmente provedores de serviços de Internet ou ISPs) de clientes.

O Monitor de Internet fornece as métricas a seguir.

Métrica	Descrição
PerformanceScore	Uma pontuação de performance representa a porcentagem estimada de tráfego que não está apresentando queda de performance.
AvailabilityScore	Uma pontuação de disponibilidade represent a a porcentagem estimada de tráfego que não está apresentando queda de disponibilidade.
BytesIn	Bytes transferidos na entrada no tráfego de Internet da sua aplicação em todas as cidades-redes da aplicação.
BytesOut	Bytes transferidos na saída no tráfego de Internet da sua aplicação em todas as cidades-redes da aplicação.

Métrica	Descrição
BytesInMonitored	Bytes transferidos na entrada no tráfego de Internet da sua aplicação nas cidades-redes monitoradas.
BytesOutMonitored	Bytes transferidos na saída no tráfego de Internet da sua aplicação nas cidades-redes monitoradas.
Tempo de ida e volta (RTT)	Tempo de ida e volta entre as Regiões da AWS, os ASNs (geralmente provedores de serviços de Internet ou ISPs) e os locais (como as cidades) específicos de suas VPCs, Network Load Balancers, distribuições do CloudFront ou diretórios do WorkSpaces.
CityNetworksMonitored	O número de cidades-redes que o Monitor de Internet monitora para o tráfego de Internet da sua aplicação. Isso nunca é mais do que o limite superior que você define como o máximo de cidades-redes para o monitor.
TrafficMonitoredPercent	O percentual do tráfego total de aplicações da Internet para esse monitor que é representada (incluída) pelas cidades-redes que o Monitor de Internet está monitorando. Isso será menos do que 100 (ou seja, menos de 100%) se os clientes acessarem sua aplicação em mais cidades-redes do que o limite máximo de cidades-redes que você definiu para o monitor.
CityNetworksFor100PercentTraffic	O número para o qual você deve definir o limite máximo de suas cidades-redes se quiser monitorar 100% do tráfego de Internet da sua aplicação no Monitor de Internet.

Métrica	Descrição
CityNetworksFor99PercentTraffic	O número para o qual você deve definir o limite máximo de suas cidades-redes se quiser monitorar 99% do tráfego de Internet da sua aplicação no Monitor de Internet.
CityNetworksFor95PercentTraffic	O número para o qual você deve definir o limite máximo de suas cidades-redes se quiser monitorar 95% do tráfego de Internet da sua aplicação no Monitor de Internet.
CityNetworksFor90PercentTraffic	O número para o qual você deve definir o limite máximo de suas cidades-redes se quiser monitorar 90% do tráfego de Internet da sua aplicação no Monitor de Internet.
CityNetworksFor75PercentTraffic	O número para o qual você deve definir o limite máximo de suas cidades-redes se quiser monitorar 75% do tráfego de Internet da sua aplicação no Monitor de Internet.
CityNetworksFor50PercentTraffic	O número para o qual você deve definir o limite máximo de suas cidades-redes se quiser monitorar 50% do tráfego de Internet da sua aplicação no Monitor de Internet.
CityNetworksFor25PercentTraffic	O número para o qual você deve definir o limite máximo de suas cidades-redes se quiser monitorar 25% do tráfego de Internet da sua aplicação no Monitor de Internet.

Note

Para ver exemplos do uso de várias dessas métricas para ajudar a determinar os valores a serem escolhidos para o máximo de uma cidade-rede para seu monitor, consulte Escolha do valor máximo de uma cidade-rede.

Para ter mais informações, consulte Usar métricas do Amazon CloudWatch.

Uso do Amazon Athena para consultar medições de Internet nos arquivos de log do Amazon S3

É possível usar o Amazon Athena para consultar e visualizar as medições da Internet que o Monitor de Internet do Amazon CloudWatch publicar em um bucket do Amazon S3. Há uma opção no Monitor de Internet de publicar medições da Internet para sua aplicação em um bucket S3 de tráfego voltado para a Internet para suas cidades-redes monitoradas (locais e ASNs de clientes, geralmente provedores de serviços de Internet ou ISPs). Independentemente de você escolher publicar medições no S3 ou não, o Monitor de Internet publica automaticamente medições da Internet no CloudWatch Logs a cada cinco minutos para as 500 principais (por volume de tráfego) cidades-redes para cada monitor.

Este capítulo inclui etapas sobre como criar uma tabela no Athena para medições da Internet localizadas em um arquivo de log do S3 e, em seguida, fornece exemplos de consultas para ver diferentes visualizações das medições. Por exemplo, é possível consultar as 10 principais redes de cidades afetadas por impacto de latência.

Uso do Amazon Athena para criar uma tabela para medições da Internet no Monitor de Internet

Para começar a usar o Athena com seus arquivos de log do S3 do Monitor de Internet, primeiro crie uma tabela para as medições da Internet.

Siga as etapas desse procedimento para criar uma tabela no Athena com base nos arquivos de log do S3. Em seguida, será possível executar consultas do Athena na tabela, como estes exemplos de consultas de medições na Internet, para obter informações sobre suas medições.

Para criar uma tabela do Athena

1. Abra o console do Athena em https://console.aws.amazon.com/athena/.
2. No editor de consultas do Athena, insira uma instrução de consulta para gerar uma tabela com as medições de Internet do Monitor de Internet. Substitua o valor do parâmetro LOCATION pelo local do bucket do S3 em que suas medições de Internet do Monitor de Internet estão armazenadas.

```
CREATE EXTERNAL TABLE internet_measurements (
    version INT,
    timestamp INT,
```

```
    clientlocation STRING,
    servicelocation STRING,
    percentageoftotaltraffic DOUBLE,
    bytesin INT,
    bytesout INT,
    clientconnectioncount INT,
    internethealth STRING,
    trafficinsights STRING
)
PARTITIONED BY (year STRING, month STRING, day STRING)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION
's3://bucket_name/bucket_prefix/AWSLogs/account_id/internetmonitor/AWS_Region/'
TBLPROPERTIES ('skip.header.line.count' = '1');
```

3. Insira uma instrução para criar uma partição para ler os dados. Por exemplo, a consulta a seguir cria uma única partição para uma data e um local especificado:

```
ALTER TABLE internet_measurements
ADD PARTITION (year = 'YYYY', month = 'MM', day = 'dd')
LOCATION
's3://bucket_name/bucket_prefix/AWSLogs/account_id/internetmonitor/AWS_Region/YYYY/
MM/DD';
```

4. Escolha Executar.

Exemplo de instruções do Athena para medições de Internet

A seguir há um exemplo de uma instrução para gerar uma tabela:

```
CREATE EXTERNAL TABLE internet_measurements (
    version INT,
    timestamp INT,
    clientlocation STRING,
    servicelocation STRING,
    percentageoftotaltraffic DOUBLE,
    bytesin INT,
    bytesout INT,
    clientconnectioncount INT,
    internethealth STRING,
    trafficinsights STRING
)
PARTITIONED BY (year STRING, month STRING, day STRING)
```

```
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://internet-measurements/TestMonitor/AWSLogs/1111222233332/internetmonitor/
us-east-2/'
TBLPROPERTIES ('skip.header.line.count' = '1');
```

A seguir há um exemplo de uma instrução para criar uma partição para ler os dados:

```
ALTER TABLE internet_measurements
ADD PARTITION (year = '2023', month = '04', day = '07')
LOCATION 's3://internet-measurements/TestMonitor/AWSLogs/1111222233332/internetmonitor/
us-east-2/2023/04/07/'
```

Exemplos de consultas do Amazon Athena para uso com medições de Internet no Monitor de Internet

Esta seção inclui exemplos de consultas que podem ser usados com o Amazon Athena para obter informações sobre as medições de Internet da sua aplicação publicadas no Amazon S3.

Consultar seus 10 principais locais e ASNs de clientes afetados (por percentual total de tráfego)

Execute essa consulta do Athena para retornar suas 10 principais cidades-redes afetadas (por percentual total do tráfego), ou seja, locais e ASNs de clientes, geralmente provedores de serviços de Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(percentageoftotaltraffic) as percentageoftotaltraffic
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageoftotaltraffic desc
limit 10
```

Consultar seus 10 principais locais e ASNs de clientes afetados (por disponibilidade)

Execute essa consulta do Athena para retornar suas 10 principais cidades-redes afetadas (por percentual total do tráfego), ou seja, locais e ASNs de clientes, geralmente provedores de serviços de Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
```

```
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(
        cast(
            json_extract_scalar(
                internetHealth,
                '$.availability.percentageoftotaltrafficimpacted'
            )
        as double )
    ) as percentageOfTotalTrafficImpacted
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageOfTotalTrafficImpacted desc
limit 10
```

Consultar seus 10 principais locais e ASNs de clientes afetados (pela latência)

Execute essa consulta do Athena para retornar suas 10 principais cidades-redes afetadas (pelo impacto da latência), ou seja, locais e ASNs de clientes, geralmente provedores de serviços de Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(
        cast(
            json_extract_scalar(
                internetHealth,
                '$.performance.percentageoftotaltrafficimpacted'
            )
        as double )
    ) as percentageOfTotalTrafficImpacted
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageOfTotalTrafficImpacted desc
limit 10
```

Consultar os destaques do tráfego para seus locais e ASNs de clientes

Execute essa consulta do Athena para retornar os destaques do tráfego, incluindo pontuação de disponibilidade, pontuação de performance e tempo até o primeiro byte de suas cidades-redes, ou seja, locais de clientes e ASNs, geralmente provedores de serviços de Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.subdivision') as subdivision,
    json_extract_scalar(clientLocation, '$.country') as country,
    avg(cast(json_extract_scalar(internetHealth, '$.availability.experiencescore') as
 double)) as availabilityScore,
    avg(cast(json_extract_scalar(internetHealth, '$.performance.experiencescore') as
 double)) performanceScore,
    avg(cast(json_extract_scalar(trafficinsights,
 '$.timetofirstbyte.currentexperience.value') as double)) as averageTTFB,
    sum(bytesIn) as bytesIn,
    sum(bytesOut) as bytesOut,
    sum(bytesIn + bytesOut) as totalBytes
FROM internet_measurements
where json_extract_scalar(clientLocation, '$.city') != 'N/A'
GROUP BY
json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.subdivision'),
    json_extract_scalar(clientLocation, '$.country')
ORDER BY totalBytes desc
limit 100
```

Para obter mais informações sobre como usar o Athena, consulte o Guia do usuário do Amazon Athena.

Como usar a interface de consulta do Monitor de Internet do Amazon CloudWatch

Uma opção para saber mais sobre o tráfego da Internet para sua aplicação da AWS é usar a interface de consulta do Monitor de Internet do Amazon CloudWatch. Para usar a interface de consulta, você cria uma consulta com filtros de dados escolhidos por você e, em seguida, executa a consulta para retornar um subconjunto dos dados do Monitor de Internet. A exploração dos dados que a consulta retorna pode fornecer insights sobre a performance da sua aplicação na Internet.

Você pode consultar e explorar todas as métricas que o Monitor de Internet captura com seu monitor, incluindo pontuações de disponibilidade e performance, bytes transferidos, tempos de ida e volta e tempo até o primeiro byte (TTFB).

O Monitor de Internet usa a interface de consulta para fornecer os dados que podem ser explorados no painel do console do Monitor de Internet. Ao usar as opções de pesquisa no painel, na guia Explorador histórico ou na guia Informações de tráfego, você pode consultar e filtrar dados da Internet para sua aplicação.

Se você quiser mais flexibilidade para explorar e filtrar seus dados do que o painel oferece, você pode usar a interface de consulta por conta própria, usando as operações da API do Monitor de Internet com a AWS Command Line Interface ou com um SDK da AWS. Esta seção apresenta os tipos de consultas que podem ser usados com a interface de consulta e os filtros que podem ser especificados para criar um subconjunto de dados, a fim de obter insights sobre o tráfego da Internet para a sua aplicação.

Tópicos

Como usar a interface de consulta

Você cria uma consulta com a interface de consulta ao escolher um tipo de consulta e, em seguida, especificar valores de filtro para retornar um subconjunto específico desejado dos dados do arquivo de log. Em seguida, você pode trabalhar com o subconjunto de dados para filtrar e classificar mais detalhadamente, criar relatórios e assim por diante.

O processo de consulta funciona da seguinte forma:

1. Quando você executa uma consulta, o Monitor de Internet retorna um `query ID` que é exclusivo da consulta. Esta seção descreve os tipos de consulta disponíveis e as opções de filtragem de dados nas consultas. Para entender como isso funciona, consulte também a seção sobre exemplos de consultas.
2. Você especifica o ID da consulta com o nome do monitor com a operação da API GetQueryResults para retornar os resultados dos dados da consulta. Cada tipo de consulta retorna um conjunto diferente de campos de dados. Para saber mais, consulte Como obter resultados de consulta.

A interface de consulta apresenta os três tipos de consulta a seguir. Cada tipo de consulta retorna um conjunto diferente de informações sobre seu tráfego a partir dos arquivos de log, conforme mostrado.

- Medições: fornece pontuação de disponibilidade, pontuação de performance, tráfego total e tempos de ida e volta, em intervalos de cinco minutos.

- Principais localizações: fornece pontuação de disponibilidade, pontuação de performance, tráfego total e informações de tempo até o primeiro byte (TTFB) para as principais combinações de localização e ASN que estão sendo monitorados, por volume de tráfego.
- Detalhes das principais localizações: fornece o TTFB para o Amazon CloudFront, sua configuração atual e a configuração do Amazon EC2 com melhor performance, em intervalos de uma hora.

Com cada um desses tipos de consulta, você pode filtrar mais os dados especificando um ou mais dos seguintes critérios:

- localização da AWS: para a localização da AWS, você pode especificar o CloudFront ou uma Região da AWS, como `us-east-2`, `us-west-2`, e assim por diante.
- ASN: especifique uma ASN, que normalmente é um provedor de serviços de Internet (ISP).
- Localização do cliente: para a localização, especifique uma cidade, metro, subdivisão ou país.
- Geo: especifique a `geo` para algumas consultas. Isso é necessário para as consultas que usam o tipo de consulta `Top locations`, mas não é permitido para outros tipos de consulta. Para saber quando especificar o `geo` para os parâmetros de filtro, consulte a seção de exemplos de consulta.

Os operadores que você pode usar para filtrar seus dados são `EQUALS` e `NOT_EQUALS`. Para obter detalhes sobre os parâmetros de filtragem, consulte a operação da API FilterParameter.

Para obter detalhes sobre as operações da interface de consulta, consulte as seguintes operações de API no Guia de referência da API do Amazon CloudWatch para o Monitor de Internet do Amazon CloudWatch:

- Para criar e executar uma consulta, consulte a operação da API StartQuery.
- Para interromper uma consulta, consulte a operação da API StopQuery.
- Para retornar dados de uma consulta que você criou, consulte a operação da API GetQueryResults.
- Para recuperar o status de uma consulta, consulte a operação da API GetQueryStatus.

Exemplos de consulta

Para criar uma consulta que você possa usar para recuperar um conjunto filtrado de dados do arquivo de log do seu monitor, use a operação da API StartQuery. Você especifica um tipo de consulta e parâmetros de filtro para a consulta. Então, quando você usar a operação da API da

interface de consulta do Monitor de Internet para obter os resultados da consulta usando a consulta, ela recuperará o subconjunto dos dados com os quais você deseja trabalhar.

Para ilustrar como funcionam os tipos de consulta e os parâmetros de filtro, vamos dar uma olhada em alguns exemplos.

Exemplo 1

Digamos que você queira recuperar todos os dados do arquivo de log do seu monitor para um país específico, exceto para uma cidade. O exemplo a seguir mostra os parâmetros de filtro para uma consulta que você poderia criar com a operação `StartQuery` para esse cenário.

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "MEASUREMENTS"
   FilterParameters: [
      {
       Field: "country",
       Operator: "EQUALS",
       Values: ["Germany"]
      },
      {
       Field: "city",
       Operator: "NOT_EQUALS",
       Values: ["Berlin"]
      },
    ]
}
```

Exemplo 2

Como outro exemplo, digamos que você queira ver suas principais localizações por área metropolitana. Você pode usar o exemplo de consulta a seguir para esse cenário.

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATIONS"
   FilterParameters: [
```

```
      {
       Field: "geo",
       Operator: "EQUALS",
       Values: ["metro"]
      },
    ]
}
```

Exemplo 3

Agora, digamos que você queira ver as principais combinações de cidade-rede na área metropolitana de Los Angeles. Para fazer isso, especifique `geo=city`, e depois defina `metro` como Los Angeles. Agora, a consulta retorna as principais cidades-redes na área metropolitana de Los Angeles em vez das principais metro+redes em geral.

Aqui está o exemplo de consulta que você pode usar:

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATIONS"
   FilterParameters: [
      {
       Field: "geo",
       Operator: "EQUALS",
       Values: ["city"]
      },
      {
       Field: "metro",
       Operator: "EQUALS",
       Values: ["Los Angeles"]
      }
    ]
}
```

Exemplo 4

Por fim, digamos que você queira recuperar dados TTFB para uma subdivisão específica (por exemplo, um estado dos EUA).

A seguir, um exemplo de consulta para esse cenário:

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATION_DETAILS"
   FilterParameters: [
      {
       Field: "subdivision",
       Operator: "EQUALS",
       Values: ["California"]
      },
    ]
}
```

Obter resultados da consulta

Depois de definir uma consulta, você pode retornar um conjunto de resultados com a consulta executando outra operação da API do Monitor de Internet, GetQueryResults. Ao executar `GetQueryResults`, você especifica o ID da consulta que você definiu, juntamente com o nome do seu monitor. `GetQueryResults` recupera os dados da consulta especificada em um conjunto de resultados.

Quando você executar uma consulta, certifique-se de que a consulta tenha sido concluída antes de usar `GetQueryResults` para visualizar os resultados. Você pode determinar se a consulta foi concluída usando a operação da API GetQueryStatus. Quando o `Status` da consulta for de `SUCCEEDED`, você pode prosseguir com a análise dos resultados.

Quando a consulta for concluída, você poderá usar as seguintes informações como auxílio para a análise dos resultados. Cada tipo de consulta que você usa para criar uma consulta inclui um conjunto exclusivo de campos de dados dos arquivos de log, conforme descrito na lista a seguir:

Medições

O tipo de consulta `measurements` retorna os seguintes dados:

`timestamp, availability, performance, bytes_in, bytes_out, rtt_p50, rtt_p90, rtt_p95`

Principais localizações

O tipo de consulta `top locations` agrupa os dados por localização e fornece a média dos dados ao longo do período. Os dados que esse tipo de consulta retorna incluem o seguinte:

`aws_location, city, metro, subdivision, country, asn, availability, performance, bytes_in, bytes_out, current_fbl, best_ec2, best_ec2_region, best_cf_fbl`

Observe que `city`, `metro`, e `subdivision` só serão retornados se você escolher esse tipo de localização para o campo `geo`. Os seguintes campos de localização são retornados, dependendo do tipo de localização que você especificar para `geo`:

```
city = city, metro, subdivision, country
metro = metro, subdivision, country
subdivision = subdivision, country
country = country
```

Detalhes das principais localizações

O tipo de consulta `top locations details` retorna dados agrupados hora a hora. A consulta retorna os seguintes dados:

`timestamp, current_service, current_fbl, best_ec2_fbl, best_ec2_region, best_cf_fbl`

Quando você executa a operação da API `GetQueryResults`, o Monitor de Internet retorna o seguinte na resposta:

- Uma matriz de cadeia de dados que contém os resultados que a consulta retorna. As informações são retornadas em matrizes alinhadas com o campo `Fields`, também retornadas pela chamada da API. Usando o campo `Fields`, você pode analisar as informações do repositório `Data` e, em seguida, filtrá-las ou classificá-las para seus objetivos.
- Uma matriz de campos que lista os campos para os quais a consulta retornou dados (na resposta do campo `Data`). Cada item na matriz é um par nome-tipo de dados, como `availability_score-float`.

Solução de problemas

Caso sejam retornados erros quando você usar as operações da API da interface de consulta, verifique se você tem as permissões necessárias para usar o Monitor de Internet do Amazon CloudWatch. Especificamente, certifique-se de que você tenha as seguintes permissões:

```
internetmonitor:StartQuery
```

```
internetmonitor:GetQueryStatus
internetmonitor:GetQueryResults
internetmonitor:StopQuery
```

Essas permissões estão incluídas na política do AWS Identity and Access Management recomendada para usar o painel do Monitor de Internet no console. Para ter mais informações, consulte Permissões do IAM para o Monitor de Internet do Amazon CloudWatch.

Criação de alarmes com o Monitor de Internet do Amazon CloudWatch

É possível criar alarmes do Amazon CloudWatch com base nas métricas do Monitor de Internet do Amazon CloudWatch, da mesma forma que é possível criar para outras métricas do Amazon CloudWatch.

Por exemplo, é possível criar um alarme com base na métrica `PerformanceScore` do Monitor de Internet e configurá-lo para enviar uma notificação quando a métrica for menor que o valor escolhido. Você configura alarmes para métricas do Monitor de Internet seguindo as mesmas diretrizes de outras métricas do CloudWatch.

A seguir há exemplos de métricas do Monitor de Internet para as quais é possível escolher criar um alarme:

- PerformanceScore
- AvailabilityScore
- RoundtripTime

Para ver todas as métricas disponíveis para o Monitor de Internet , consulte Usar o CloudWatch Logs Metrics com o Monitor de Internet do Amazon CloudWatch.

O procedimento a seguir fornece um exemplo de configuração de um alarme de PerformanceScore navegando até a métrica no painel do CloudWatch. Em seguida, você segue as etapas padrão do CloudWatch para criar um alarme com base em um limite escolhido e configurar uma notificação, ou escolher outras opções.

Para criar um alarme para PerformanceScore no CloudWatch Metrics

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Escolha Métricas e, em seguida, escolha Todas as métricas.

3. Filtre por Monitor de Internet escolhendo `AWS/InternetMonitor`.
4. Escolha MeasurementSource, MonitorName.
5. Na lista, selecione PerformanceScore.
6. Na guia GraphedMetrics, em Ações, escolha o ícone do sino para criar um alarme com base em um limite estático.

Agora, siga as etapas padrão do CloudWatch para escolher as opções para o alarme. Por exemplo, é possível optar por ser notificado por uma mensagem do Amazon SNS se o PerformanceScore estiver abaixo de um número limite específico. Como alternativa, ou além disso, é possível adicionar o alarme a um painel.

Lembre-se do seguinte:

- Normalmente, as métricas do Monitor de Internet são calculadas e publicadas em 20 minutos.
- Quando você criar um alarme com base nas métricas do Monitor de Internet, certifique-se de levar em conta o pequeno atraso antes da publicação ao definir o período de lookback de um alarme. Recomendamos que você configure os Períodos de avaliação com um período de lookback de, no mínimo, 25 minutos.

Para saber mais sobre como usar os alarmes do CloudWatch com o Monitor de Internet, consulte a seguinte postagem do blog: Using Amazon CloudWatch Internet Monitor for enhanced internet observability.

Para obter mais informações sobre as opções disponíveis quando você cria um alarme do CloudWatch, consulte Criar um alarme do CloudWatch com base em um limite estático.

Usar o Monitor de Internet do Amazon CloudWatch com o Amazon EventBridge

Os eventos de integridade que o Monitor de Internet do Amazon CloudWatch cria para problemas de rede são publicados com o Amazon EventBridge, para que você possa enviar notificações sobre qualquer degradação na experiência dos usuários finais para a aplicação.

Para usar o EventBridge para trabalhar com eventos de integridade do Monitor de Internet, siga as orientações aqui.

Para configurar uma regra para o Monitor de Internet no EventBridge

1. No AWS Management Console, no EventBridge, escolha Rules (Regras) e insira um nome e uma descrição. Crie a regra no barramento de eventos Default (Padrão).
2. Na Etapa 2, selecione Outra como origem do evento e, em Padrão de evento, selecione a origem a seguir.

```
{
  "source": ["aws.internetmonitor"]
}
```

3. Na Etapa 3, como destino, selecione AWS Service (Serviço) e CloudWatch Logs Group (Grupo de logs do CloudWatch) e depois selecione um grupo de logs ou crie um novo.
4. Adicione as tags desejadas e depois crie a regra. Isso deve preencher o grupo de logs do CloudWatch selecionado com eventos do EventBridge.

Para obter informações sobre como as regras do EventBridge funcionam com padrões de eventos, consulte Padrões de eventos no EventBridge no Guia do usuário do Amazon EventBridge.

Solução de problemas em erros de acesso a logs e métricas do CloudWatch

Para oferecer suporte a alguns recursos, o Monitor de Internet do Amazon CloudWatch deve interagir com determinados recursos do Amazon CloudWatch, incluindo logs e métricas. Se o Monitor de Internet não conseguir acessar os recursos do CloudWatch para os quais ele exige acesso, o Monitor de Internet definirá um código de status de `FAULT_ACCESS_CLOUDWATCH` para o monitor.

Há vários motivos pelos quais o monitor pode apresentar o estado `FAULT_ACCESS_CLOUDWATCH`. As seções a seguir listam as possíveis causas desses erros e as etapas de solução de problemas sugeridas.

O Monitor de Internet não conseguiu acessar os logs do CloudWatch na sua conta

O Monitor de Internet publica logs de diagnóstico sobre o tráfego de aplicações que seu monitor rastreia. Ele publica esses logs em grupos de logs no CloudWatch Logs na seguinte localização: `/aws/internet-monitor/`*`monitor_name`*`/[byCity|byMetro|bySubdivision|byCountry]`. O Monitor de Internet não conseguiu acessar esses grupos de logs.

Estados de erro e possíveis soluções:

- Erro de controle de utilização do PutLogEvents: o serviço do Monitor de Internet pode ter sido submetido ao controle de utilização ao tentar publicar os logs do monitor no CloudWatch. Revise os limites de controle de utilização da sua conta e, se necessário, solicite um aumento no limite.
- Grupo de logs não encontrado: desabilite e reabilite o monitor. Habilitar um monitor reinicia a criação do grupo de logs, o que pode corrigir o problema.
- Erro de acesso negado ao PutLogEvents: entre em contato com o suporte da AWS para obter assistência.
- Erro desconhecido ou geral do PutLogEvents: entre em contato com o suporte da AWS para obter assistência.

O Monitor de Internet não conseguiu acessar as métricas do CloudWatch na sua conta

O Monitor de Internet fornece métricas específicas do CloudWatch sobre o tráfego de aplicações que é rastreado por um monitor. Ocorreu um erro quando Monitor de Internet tentou entregar essas métricas ao CloudWatch.

Estados de erro e possíveis soluções:

- Erro de controle de utilização do PutMetricData: o serviço do Monitor de Internet pode ter sido submetido ao controle de utilização ao tentar publicar as métricas do monitor no CloudWatch. Revise os limites de controle de utilização da sua conta e, se necessário, solicite um aumento no limite.
- Erro de acesso negado ao PutMetricData: entre em contato com o suporte da AWS para obter assistência.
- Erro desconhecido ou geral do PutMetricData: entre em contato com o suporte da AWS para obter assistência.

Proteção de dados e privacidade de dados com o Monitor de Internet do Amazon CloudWatch

Saiba como o modelo de responsabilidade compartilhada da AWS se aplica à proteção e à privacidade dos dados no Monitor de Internet do Amazon CloudWatch. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem da AWS. Você é responsável por manter o controle sobre o conteúdo que hospeda nessa infraestrutura. Para obter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre

privacidade de dados. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem The AWS Shared Responsibility Model and GDPR no blog de segurança da AWS. Para obter mais recursos sobre o cumprimento dos requisitos do GDPR, consulte o Centro Geral de Regulamentação de Proteção de Dados (GDPR).

É altamente recomendável que você nunca coloque informações de identificação confidenciais de clientes finais, como números de conta, endereços de e-mail ou outras informações pessoais de clientes finais, em campos de formato livre. Todos os dados inseridos no Monitor de Internet do Amazon CloudWatch ou em outros serviços podem ser incluídos em logs de diagnóstico.

Identity and Access Management para o Monitor de Internet do Amazon CloudWatch

O AWS Identity and Access Management (IAM) é um serviço da AWS service (Serviço da AWS) que ajuda a controlar o acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode ser autenticado (fazer login) e autorizado (ter permissões) para usar os recursos do Monitor de Internet. O IAM é um AWS service (Serviço da AWS) que pode ser usado sem custo adicional.

Important

Alterações nos recursos do Monitor de Internet em 24 de fevereiro de 2023
Se você criou políticas do IAM que incluíam recursos do Monitor de Internet antes de 24 de fevereiro de 2023, esteja ciente das alterações a seguir nos recursos e tipos de recursos do Monitor de Internet.

- O recurso HealthEvents foi renomeado para HealthEvent.
- O ARN e os formatos Regex do recurso HealthEvent foram atualizados.
- O ARN e os formatos Regex do recurso Monitor foram atualizados.
- As permissões em nível de recurso para a ação GetHealthEvent agora têm suporte somente no tipo de recurso HealthEvent. Elas não têm suporte no recurso Monitor.
- TagResource, UntagResource e ListTagsForResource para o tipo de recurso Monitor foram atualizadas para serem obrigatórias.

Para ver mais informações sobre as ações, os recursos e as chaves de condição que podem ser especificados nas políticas para gerenciar o acesso aos recursos da AWS no Monitor

de Internet, consulte Ações, recursos e chaves de condição para o Monitor de Internet do Amazon CloudWatch.

Conteúdo

Como o Monitor de Internet do Amazon CloudWatch opera com o IAM

Antes de usar o IAM para gerenciar o acesso ao Monitor de Internet, saiba quais recursos do IAM estão disponíveis para uso com o Monitor de Internet.

Para ver tabelas que mostrem uma visão similar de alto nível sobre como os serviços da AWS funcionam com a maioria dos recursos do IAM, consulte Serviços da AWS compatíveis com o IAM no Guia do usuário do IAM.

Recursos do IAM que podem ser usados com o Monitor de Internet do Amazon CloudWatch

Atributo do IAM	Suporte para o Monitor de Internet
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não
Ações de políticas	Sim
atributos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC (tags em políticas)	Parcial

Atributo do IAM	Suporte para o Monitor de Internet
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Não
Funções vinculadas ao serviço	Sim

Políticas baseadas em identidade para o Monitor de Internet

É compatível com políticas baseadas em identidade	Sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou atributos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Não é possível especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou função à qual ela está anexado. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elementos da política JSON do IAM no Guia do Usuário do IAM.

Políticas baseadas em recursos no Monitor de Internet

Oferece suporte a políticas baseadas em recursos	Não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as

políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico.

Ações de políticas para o Monitor de Internet

Oferece suporte a ações de políticas	Sim

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista das ações do Monitor de Internet, consulte Ações definidas pelo Monitor de Internet do Amazon CloudWatch na Referência de autorização de serviço.

As ações de políticas no Monitor de Internet usam o prefixo a seguir antes da ação:

```
internetmonitor
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [
      "internetmonitor:action1",
      "internetmonitor:action2"
         ]
```

Você também pode especificar várias ações usando caracteres-curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação:

```
"Action": "internetmonitor:Describe*"
```

Recursos de política para o Monitor de Internet

Oferece suporte a atributos de políticas	Sim

Na Referência de autorização de serviço, é possível ver as informações a seguir relacionadas ao Monitor de Internet:

- Para ver uma lista dos tipos de recursos do Monitor de Internet e seus ARNs, consulte Recursos definidos pelo Monitor de Internet do Amazon CloudWatch.
- Para saber as ações que podem ser especificadas com o ARN de cada recurso, consulte Ações definidas pelo Monitor de Internet do Amazon CloudWatch.

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando Nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem suporte a um tipo de atributo específico, conhecido como permissões em nível de atributo.

Para ações não compatíveis com permissões no nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Chaves de condição de política para Monitor de Internet

Compatível com chaves de condição de política específicas do serviço	Sim

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou `Condition` bloco de) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões

condicionais que usam agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação lógica `OR`. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos de política do IAM: variáveis e tags no Guia do usuário do IAM.

A AWS é compatível com chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as AWS chaves de condição globais da , consulte AWSChaves de contexto de condição globais da no Guia do usuário do IAM.

Para ver uma lista de chaves de condição do Monitor de Internet, consulte Chaves de condição do Monitor de Internet do Amazon CloudWatch na Referência de autorização de serviço. Para saber com quais ações e recursos é possível usar a chave de condição, consulte Ações definidas pelo Monitor de Internet do Amazon CloudWatch.

ACLs no Monitor de Internet

Oferece suporte a ACLs	Não

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com o Monitor de Internet

Oferece suporte a ABAC (tags em políticas)	Parcial

O Monitor de Internet tem suporte parcial para tags nas políticas. Ele oferece suporte à aplicação de tags a um recurso, os monitores.

Para usar tags com o Monitor de Internet, use a AWS Command Line Interface ou um AWS SDK. Não há suporte para a aplicação de tags no Monitor de Internet com o AWS Management Console.

Para saber mais sobre o uso de tags em políticas em geral, consulte as informações a seguir.

O controle de acesso baseado em recurso (ABAC) é uma estratégia de autorização que define permissões com base em recursos. Na AWS, esses atributos são chamados de tags. É possível anexar tags a entidades do IAM (usuários ou perfis) e a muitos atributos da AWS. A marcação de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do atributo que ela está tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as chaves de condição `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial.

Para obter mais informações sobre o ABAC, consulte O que é ABAC? no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

Uso de credenciais temporárias com o Monitor de Internet

Oferece suporte a credenciais temporárias	Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Você está usando credenciais temporárias se faz login no AWS Management Console usando qualquer método, exceto um nome de usuário e uma senha. Por exemplo, quando você acessa a AWS usando o link de autenticação única (SSO) da empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz

login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar perfis, consulte Alternar para um perfil (console) no Guia do usuário do IAM.

Você pode criar credenciais temporárias manualmente usando a AWS CLI ou a API da AWS. Em seguida, você pode usar essas credenciais temporárias para acessar aAWS .AWS A recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para mais informações, consulte Credenciais de segurança temporárias no IAM.

Permissões de entidade principal entre serviços para o Monitor de Internet

Suporte para o recurso Encaminhamento de sessões de acesso (FAS)	Sim

Quando você usa um usuário do IAM ou um perfil para executar ações na AWS, você é considerado uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

Perfis de serviço para o Monitor de Internet

Oferece suporte a perfis de serviço	Não

O perfil de serviço é um perfil do IAMhttps://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do usuário do IAM.

Perfil vinculado ao serviço para o Monitor de Internet

Oferece suporte a perfis vinculados ao serviço	Sim

Um perfil vinculado ao serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode assumir o perfil de executar uma ação em seu nome. Os perfis vinculados ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

Para obter mais informações sobre o perfil vinculado a serviço para o Monitor de Internet, consulte Perfil vinculado ao serviço para o Monitor de Internet do Amazon CloudWatch.

Para obter detalhes sobre a criação ou o gerenciamento de funções vinculadas a serviços em geral na AWS, consulte Serviços da AWS compatíveis com o IAM. Encontre um serviço na tabela que inclua um Yes na coluna Função vinculada ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado ao serviço desse serviço.

Políticas gerenciadas pela AWS para o Monitor de Internet do Amazon CloudWatch

Uma política gerenciada pela AWS é uma política independente criada e administrada pela AWS. As políticas gerenciadas pela AWS são criadas para fornecer permissões a vários casos de uso comuns a fim de que você possa começar a atribuir permissões a usuários, grupos e perfis.

Lembre-se de que as políticas gerenciadas pela AWS podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para todos os clientes da AWS usarem. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo cliente da específicas para seus casos de uso.

Você não pode alterar as permissões definidas em políticas gerenciadas pela AWS. Se a AWS atualiza as permissões definidas em uma política gerenciada pela AWS, a atualização afeta todas as identidades de entidades principais (usuários, grupos e perfis) às quais a política está vinculada. É mais provável que a AWS atualize uma política gerenciada pela AWS quando um novo AWS service (Serviço da AWS) é lançado ou novas operações de API são disponibilizadas para os serviços existentes.

Para mais informações, consulte Políticas gerenciadas pela AWS no Manual do usuário do IAM.

Política gerenciada pela AWS: CloudWatchInternetMonitorServiceRolePolicy

Essa política está anexada ao perfil vinculado ao serviço chamado AWSServiceRoleForInternetMonitor para permitir que o Monitor de Internet acesse recursos na sua conta, como os da Amazon Virtual Private Cloud ou dos Network Load Balancers, para que você possa selecioná-los ao criar um monitor. Para ter mais informações, consulte Perfil vinculado ao serviço para o Monitor de Internet do Amazon CloudWatch.

Permissões do IAM para o Monitor de Internet do Amazon CloudWatch

Para acessar as ações para trabalhar com monitores e dados no Monitor de Internet do Amazon CloudWatch, os usuários devem ter as permissões corretas.

Para obter mais informações sobre segurança no Amazon CloudWatch, consulte Gerenciamento de Identidade e Acesso para o Amazon CloudWatch.

Permissões para acesso somente leitura no Monitor de Internet do Amazon CloudWatch

Para acessar ações somente leitura para trabalhar com monitores e dados no Monitor de Internet do Amazon CloudWatch, os usuários devem ter feito login como um usuário ou perfil que tenha as seguintes permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "internetmonitor:Get*",
                "internetmonitor:List*",
                "internetmonitor:StartQuery",
                "internetmonitor:StopQuery",
                "logs:DescribeLogGroups",
                "logs:GetQueryResults",
                "logs:StartQuery",
                "logs:StopQuery"
                ],
            "Resource": "*"
        }
    ]
}
```

Permissões para acesso total no Monitor de Internet do Amazon CloudWatch

Para criar um monitor no Monitor de Internet do Amazon CloudWatch e ter acesso total às ações de trabalho com monitores e dados do Monitor de Internet do Amazon CloudWatch, os usuários devem ter feito login como um usuário ou perfil que tenha as seguintes permissões:

- Permissões para criar um perfil vinculado ao serviço associado ao Monitor de Internet. Para ter mais informações, consulte Perfil vinculado ao serviço para o Monitor de Internet do Amazon CloudWatch.
- Permissões para ações que permitem acesso total para trabalhar com monitores e dados no Monitor de Internet.

> **Note**
>
> Se você criar uma política de permissões baseada em identidade mais restritiva, os usuários com essa política talvez não tenham acesso total para criar e trabalhar com monitores e dados no Monitor de Internet.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "internetmonitor:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
internetmonitor.amazonaws.com/AWSServiceRoleForInternetMonitor",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "internetmonitor.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
```

```
            "Resource": "arn:aws:iam::*:role/aws-service-role/
internetmonitor.amazonaws.com/AWSServiceRoleForInternetMonitor"
        },
        {
            "Action": [
                "ec2:DescribeVpcs",
                "elasticloadbalancing:DescribeLoadBalancers",
                "workspaces:DescribeWorkspaceDirectories",
                "cloudfront:GetDistribution"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Perfil vinculado ao serviço para o Monitor de Internet do Amazon CloudWatch

O Monitor de Internet do Amazon CloudWatch usa um perfil vinculado ao serviço do AWS Identity and Access Management (IAM). O perfil vinculado ao serviço é um tipo exclusivo de perfil do IAM que é vinculado diretamente ao Monitor de Internet. O perfil vinculado ao serviço é predefinido pelo Monitor de Internet e inclui todas as permissões de que o serviço precisa para chamar outros produtos da AWS em seu nome.

O Monitor de Internet define as permissões do perfil vinculado ao serviço e, a menos que definido em contrário, apenas o Monitor de Internet poderá assumir o perfil. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Você só poderá excluir os perfis após excluir os recursos relacionados. Essa restrição protege os recursos do Monitor de Internet, pois não é possível remover acidentalmente as permissões de acesso a eles.

Para obter informações sobre outros serviços compatíveis com funções vinculadas aos serviços, consulte Serviços da AWS que funcionam com o IAM e procure os serviços que apresentam Sim na coluna Funções vinculadas aos serviços. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

Permissões do perfil vinculado ao serviço para o Monitor de Internet

O Monitor de Internet usa o perfil vinculado ao serviço chamado AWSServiceRoleForInternetMonitor. Esse perfil permite que o Monitor de Internet acesse os recursos na sua conta, como os recursos

da Amazon Virtual Private Cloud, as distribuições do Amazon CloudFront, os diretórios do Amazon WorkSpaces e os Network Load Balancers, para que você possa selecioná-los ao criar um monitor.

Esse perfil vinculado ao serviço usa a política gerenciada `CloudWatchInternetMonitorServiceRolePolicy`.

O perfil vinculado ao serviço AWSServiceRoleForInternetMonitor confia no serviço a seguir para assumir o perfil:

- `internetmonitor.amazonaws.com`

Para visualizar as permissões dessa política, consulte CloudWatchInternetMonitorServiceRolePolicy na Referência de política gerenciada pela AWS.

Criar um perfil vinculado ao serviço para o Monitor de Internet

Não é necessário criar manualmente um perfil vinculado ao serviço para o Monitor de Internet. A primeira vez que você cria um monitor, o Monitor de Internet cria AWSServiceRoleForInternetMonitor para você.

Para obter mais informações, consulte Criar um perfil vinculado a serviço no Guia do usuário do IAM.

Editar um perfil vinculado ao serviço para o Monitor de Internet

Após criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil porque várias entidades podem referenciá-lo. Porém, você poderá editar a descrição do perfil usando o IAM. Para ter mais informações, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Excluir um perfil vinculado ao serviço para o Monitor de Internet

Se você não precisar mais usar um recurso ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. Porém, você deve limpar os recursos do perfil vinculado ao serviço antes de poder excluí-lo manualmente.

Após remover os recursos dos monitores no Monitor de Internet e depois excluir os monitores, você poderá excluir o perfil vinculado ao serviço AWSServiceRoleForInternetMonitor.

> **Note**
>
> Se o serviço Monitor de Internet estiver usando um perfil quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, aguarde alguns minutos e tente novamente.

Como excluir manualmente a função vinculada a serviço usando o IAM

Use o console do IAM, a AWS CLI ou a API da AWS para excluir o perfil vinculado ao serviço AWSServiceRoleForInternetMonitor. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Atualizações ao perfil vinculado ao serviço do Monitor de Internet

Para atualizações do AWSServiceRoleForInternetMonitor, a política gerenciada pela AWS para o perfil vinculado ao serviço do Monitor de Internet, consulte Atualizações do CloudWatch para políticas gerenciadas pela AWS. Para obter alertas automáticos sobre alterações em políticas gerenciadas, inscreva-se no feed RSS na página Histórico de documentos do CloudWatch.

Cotas do Monitor de Internet do Amazon CloudWatch

O Monitor de Internet do Amazon CloudWatch tem as cotas a seguir.

Recurso	Cota padrão
Monitores por região	50
Recursos por monitor	50
O número de dias que os eventos de integridade do Monitor de Internet resolvidos são retidos	400

Como usar o Amazon CloudWatch Network Monitor

O Amazon CloudWatch Network Monitor fornece visibilidade da performance da rede que conecta suas aplicações hospedadas na AWS aos seus destinos on-premises e permite que você identifique

a fonte de qualquer degradação da performance da rede em questão de minutos. O Network Monitor é totalmente gerenciado pela AWS. Portanto, você não precisará instalar agentes adicionais para monitorar a performance da sua rede. Você pode visualizar rapidamente a perda de pacotes e a latência das conexões de redes híbridas, definir alertas e limites e, em seguida, tomar medidas para melhorar a experiência de rede dos usuários finais.

O Network Monitor é destinado a operadores de rede e desenvolvedores de aplicações que desejam insights em tempo real sobre a performance da rede.

Atributos principais

- Use o Network Monitor para comparar um ambiente de rede híbrida em constante mudança com métricas contínuas de latência e perda de pacotes em tempo real.
- Quando você se conecta usando o AWS Direct Connect, o Network Monitor diagnostica rapidamente a degradação da rede ao gravar o indicador de integridade da rede da AWS na sua conta do CloudWatch. Essa métrica fornece uma pontuação probabilística para determinar se a degradação da rede estava na AWS.
- O Network Monitor fornece um monitoramento sem conflitos com uma abordagem de agente totalmente gerenciada, o que significa que você não precisa instalar agentes em VPCs ou on-premises. Você só precisa especificar uma sub-rede da VPC e um endereço IP on-premises para começar.
- O Network Monitor publica métricas no CloudWatch Metrics. Você pode criar painéis para visualizar as métricas e criar limites e alarmes práticos sobre as métricas específicas da sua aplicação.

Para obter mais detalhes, consulte the section called “Como o Network Monitor funciona”.

Terminologia e componentes do Network Monitor

- Monitor: um monitor mostra os recursos para os quais você deseja visualizar as medidas de performance e disponibilidade de rede e sobre os quais deseja receber alertas de eventos de integridade. Ao criar um monitor para uma aplicação, você adiciona um recurso hospedado na AWS como a origem da rede. Em seguida, o Network Monitor cria uma lista de todos os testes possíveis entre os recursos hospedados na AWS e os seus endereços IP de destino.
- Sondas :uma sonda é o tráfego enviado do recurso hospedado na AWS para o endereço IP de destino on-premises. As métricas do Network Monitor são gravadas na sua conta do CloudWatch para cada sonda configurada em um monitor.

- Origem da rede da AWS: essa é a origem da AWS da sonda do monitor de rede, que será uma sub-rede em qualquer uma das VPCs.
- Destino: esse é o destino na rede on-premises para a origem da rede da AWS. O destino é uma combinação dos endereços IP on-premises, dos protocolos de rede, das portas e do tamanho do pacote da rede. Os endereços IPv4 e IPv6 são compatíveis.

Limitações e requisitos do Network Monitor

- O Network Monitor suporta no máximo quatro endereços IP de destino e até 24 sondas por monitor.
- Você pode ter até cem monitores por região por conta.
- As sub-redes do monitor devem pertencer à mesma conta do monitor.
- O Network Monitor não fornece failover de rede automático no caso de um problema de rede da AWS.
- Há uma cobrança para cada sonda que você cria. Para obter detalhes de preço, consulte the section called “Definição de preço”.

Como o Amazon CloudWatch Network Monitor funciona

O Network Monitor facilita o monitoramento ao fornecer uma solução totalmente gerenciada e sem agentes. Quando você cria um monitor em um recurso hospedado na AWS, a AWS cria e gerencia toda a infraestrutura em segundo plano para executar medições de tempo de ida e volta e de perda de pacotes. Como resultado, você pode escalar seu monitoramento rapidamente sem precisar instalar ou desinstalar qualquer agente na sua infraestrutura da AWS.

O Network Monitor se concentra no monitoramento das rotas percorridas pelos fluxos originários dos recursos hospedados na AWS, em vez de monitorar amplamente todos os fluxos da sua Região da AWS. Se suas workloads estiverem espalhadas por várias zonas de disponibilidade (AZs), o Network Monitor poderá monitorar as rotas de cada uma das suas sub-redes privadas.

O Network Monitor publica métricas de tempo de ida e volta e de perda de pacotes na sua conta do Amazon CloudWatch com base no intervalo de agregação definido quando você criou um monitor. Você também pode definir limites individuais de latência e perda de pacotes para cada monitor usando o CloudWatch. Por exemplo, você poderá criar um alarme para avisar se sua média de perda de pacotes for maior que o limite estático de 0,1% para uma workload sensível à perda de

pacotes. Você também pode usar a detecção de anomalias do CloudWatch para alertar sobre perda de pacotes ou métricas de latência fora dos intervalos desejados.

Medições de disponibilidade e performance

O Network Monitor envia sondas ativas periódicas do seu recurso da AWS para seus destinos on-premises. Ao criar um monitor, especifique o seguinte:

- O intervalo de agregação. O tempo, em segundos, em que o CloudWatch recebe os resultados medidos. Isso será a cada 30 ou 60 segundos. O período de agregação escolhido para o monitor se aplica a todas as sondas desse monitor.
- O protocolo da sonda. Cada sonda adicionada a um monitor deve usar os protocolos ICMP (Internet Control Message Protocol) ou TCP (Transmission Control Protocol). Consulte the section called “Protocolos de comunicação” para obter mais detalhes.
- O tamanho do pacote. O tamanho, em bytes, de cada pacote transmitido entre seu recurso hospedado na AWS e seu destino em uma única sonda. Cada sonda em um monitor pode ter seu próprio tamanho de pacote.

Em métricas,

- A métrica de tempo de ida e volta, medida em milissegundos, mede e registra uma medição de performance e registra o tempo necessário para que a sonda seja transmitida ao endereço IP de destino e para que a resposta associada seja recebida.
- A métrica de perda de pacotes mede a porcentagem do total de pacotes enviados e registra o número de sondas transmitidas que não receberam uma resposta associada, o que implica que esses pacotes foram efetivamente perdidos ao longo do caminho da rede.

Protocolos de comunicação compatíveis

As sondas baseadas em ICMP transportam solicitações de eco ICMP dos recursos hospedados na AWS para o endereço de destino e esperam uma resposta de eco ICMP do endereço de destino. O Network Monitor usa as informações sobre a solicitação de eco ICMP e as mensagens de resposta para calcular o tempo de ida e volta e as métricas de perda de pacotes.

As sondas baseadas em TCP transportam pacotes TCP SYN dos recursos hospedados na AWS para o endereço e a porta de destino e esperam um pacote TCP SYN+ACK ou RST de volta do endereço e da porta de destino. O Network Monitor usa as informações sobre as mensagens TCP

SYN e TCP SYN+ACK ou RST para calcular o tempo de ida e volta e as métricas de perda de pacotes. Além disso, o Network Monitor alterna periodicamente as portas TCP de origem para aumentar a cobertura da rede, o que pode aumentar a probabilidade de detecção de perda de pacotes.

Indicador de integridade da rede da AWS

O Network Monitor publica uma métrica do Indicador de integridade da rede (NHI), que fornece informações sobre a performance e a disponibilidade da rede para destinos conectados por meio do AWS Direct Connect. A métrica é uma medida estatística da integridade do caminho de rede controlado pela AWS do recurso hospedado na AWS, que é onde o monitor é implantado, até o local do Direct Connect.

O Network Monitor emprega detecção de anomalias para calcular quedas de disponibilidade ou degradação de performance ao longo dos caminhos da rede.

> **Note**
>
> Toda vez que você cria um novo monitor, adiciona uma sonda ou reativa uma sonda, o NHI desse monitor é adiado em algumas horas para permitir que a AWS colete dados para realizar a detecção de anomalias.

Para fornecer a métrica de integridade do NHI, o Network Monitor aplica correlação estatística a conjuntos de dados de amostra da AWS, bem como às métricas de perda de pacotes e de latência de ida e volta para tráfego que simula o seu caminho de rede. A métrica pode ser uma de duas variáveis: 1 ou 0. Um valor de 1 indica que o Network Monitor observou uma degradação da rede dentro do caminho de rede controlado pela AWS. Um valor de 0 indica que o Network Monitor não observou qualquer degradação da rede ao longo do caminho. Isso permite que você solucione problemas de rede mais rapidamente. Você pode definir alertas na métrica do NHI para receber informações sobre problemas contínuos ao longo do seus caminhos de rede.

Suporte para endereços IPv4 e IPv6

O Network Monitor fornece métricas de disponibilidade e performance em redes IPv4 ou IPv6 e pode monitorar endereços IPv4 ou IPv6 de VPCs de pilha dupla. O Network Monitor não permite que destinos IPv4 e IPv6 sejam configurados no mesmo monitor, mas você pode criar destinos separados para somente IPv4 e somente IPv6.

Disponibilidade de regiões

No momento, o Network Monitor está disponível nas seguintes Regiões da AWS:

Região	
Ásia-Pacífico (Hong Kong)	ap-east-1
Ásia-Pacífico (Mumbai)	ap-south-1
Ásia-Pacífico (Seul)	ap-northeast-2
Ásia-Pacífico (Singapura)	ap-southeast-1
Ásia-Pacífico (Sydney)	ap-southeast-2
Ásia-Pacífico (Tóquio)	ap-northeast-1
Oeste do Canadá (Calgary)	ca-west-1
Europa (Frankfurt)	eu-central-1

Região	
Europa (Irlanda)	eu-west-1
Europa (Londres)	eu-west-2
Europa (Paris)	eu-west-3
Europa (Estocolmo)	eu-north-1
Oriente Médio (Barém)	me-south-1
América do Sul (São Paulo)	sa-east-1
Leste dos EUA (Norte da Virgínia)	us-east-1
Leste dos EUA (Ohio)	us-east-2
Oeste dos EUA (Norte da Califórnia)	us-west-1

Região	
Oeste dos EUA (Oregon)	us-west-2

Como criar um Network Monitor

As etapas a seguir descrevem a criação de um monitor e a adição das sondas necessárias. Para sondas, você escolherá a sub-rede de origem e até quatro endereços IP de destino para um máximo de 24 sondas por monitor. Você pode criar um monitor usando o console do Amazon CloudWatch, a linha de comando ou a API.

Tópicos

Criar um Network Monitor usando o console

As etapas a seguir descrevem a criação de um monitor usando o console do Amazon CloudWatch. Você escolherá as sub-redes de origem e, em seguida, adicionará até quatro destinos para criar até 24 sondas por monitor. Você pode criar um monitor usando o console do Amazon CloudWatch, a linha de comando ou o SDK.

> **Important**
>
> Essas etapas foram projetadas para serem concluídas de uma só vez. Não será possível salvar trabalhos em andamento para continuar mais tarde.

Definir os detalhes do monitor

A primeira etapa para criar um monitor é definir os detalhes básicos. Isso inclui dar um nome ao monitor e definir o período de agregação. Você pode adicionar tags opcionais ao monitor.

Como definir os detalhes do monitor

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/ e, em seguida, em Monitoramento de rede, escolha Network Monitor.
2. Escolha Criar monitor.
3. Em Nome do monitor, insira o nome que você deseja usar para esse monitor.
4. No Período de agregação, escolha com que frequência você deseja enviar métricas para o CloudWatch. Os períodos de agregação disponíveis são:

   - 30 segundos
   - 60 segundos

   > **Note**
   >
   > Um período de agregação mais curto fornece uma detecção mais rápida de problemas de rede. No entanto, o período de agregação escolhido pode afetar sua estrutura de faturamento. Para obter mais informações sobre a definição de preços, consulte a página Definição de preços do Amazon CloudWatch.

5. (Opcional) Na seção Tags, adicione pares de Chave e Valor para ajudar ainda mais a identificar esse recurso, permitindo que você pesquise ou filtre informações específicas.

   1. Selecione Adicionar nova tag.
   2. Insira um nome de Chave e o Valor associado.
   3. Escolha Adicionar nova tag para adicionar essa nova tag.

      Você pode adicionar várias tags escolhendo Adicionar nova tag ou remover qualquer tag escolhendo Remover.

   4. Se você quiser associar as tags ao monitor, mantenha a opção Adicionar tags às sondas criadas pelo monitor marcada. Isso adiciona as tags às sondas do monitor, o que pode ser útil se você estiver usando autenticação ou medição baseada em tags.

6. Escolha Próximo para the section called “Escolher a origem e o destino”.

Escolher a origem e o destino

Um monitor de rede usa uma origem da AWS para as VPCs e sub-redes associadas nas regiões em que sua rede opera. O destino de um monitor é a combinação dos endereços IP on-premises, dos protocolos de rede, das portas e do tamanho do pacote da rede.

A combinação de origem e destino é denominada sonda. Você pode ter até quatro sondas por sub-rede e até um total de 24 sondas por monitor.

> **Important**
>
> Essas etapas foram projetadas para serem concluídas de uma só vez. Não será possível salvar trabalhos em andamento para continuar mais tarde.

Como escolher uma origem e um destino

1. Em Origem da rede da AWS, escolha uma ou mais sub-redes para incluir no monitor. Você pode escolher uma única VPC, que então escolherá todas as sub-redes dentro dessa VPC, ou você pode escolher sub-redes específicas. As VPCs e sub-redes que você escolher serão a origem do monitor de rede.
2. Em Destino 1, insira o endereço IP de destino da rede on-premises. Os endereços IPv4 e IPv6 são compatíveis.
3. Selecione Advanced settings (Configurações avançadas).
4. Para esse destino gerenciado pelo cliente, escolha o Protocolo da rede. Pode ser uma das duas opções abaixo:
   - ICMP
   - TCP
5. Se o Protocolo for TCP, insira as informações a seguir. Caso contrário, pule para a próxima etapa:
   1. Insira a Porta que a rede usa para se conectar. A porta deve ter um número de 1 a 65535.
   2. Insira o Tamanho do pacote. Esse é o tamanho, em bytes, de cada pacote enviado na sonda entre a origem e o destino. O tamanho do pacote deve ser um número de 56 a 8.500.
6. Escolha Adicionar destino para adicionar outro destino on-premises a esse monitor. Repita essas etapas para cada destino que você quiser adicionar.

7. Escolha Próximo ao terminar para confirmar as sondas.

Confirmar sondas

A confirmação das sondas permite que você revise a combinação de sondas de rede para o monitor. Esta página mostra todas as combinações possíveis das origens e destinos que você escolheu. Por exemplo, caso você possua seis sub-redes de origem e quatro IPs de destino, terá um total de 24 combinações de sondas possíveis.

> **Important**
>
> - Essas etapas foram projetadas para serem concluídas de uma só vez. Não será possível salvar trabalhos em andamento para continuar mais tarde.
> - A página Confirmar sondas não indica se uma sonda é válida. Portanto, é recomendável revisar cuidadosamente esta página e excluir as sondas inválidas. Se você não remover as sondas inválidas, poderá receber cobranças por elas.

Como confirmar sondas de monitor

1. Pré-requisito: the section called “Escolher a origem e o destino”.
2. Na página Confirmar sondas, revise a lista de combinações de origem e destino.
3. Escolha uma ou mais sondas que você deseja remover do monitor e escolha Remover.

   > **Note**
   >
   > Você não é solicitado a confirmar a exclusão. Depois que uma sonda for excluída, ela deverá ser configurada novamente. Você poderá adicionar uma sonda novamente a um monitor na seção Monitores de rede na página Network Monitor. Para ter mais informações, consulte the section called “Adicionar uma sonda a um monitor”.

4. Escolha Próximo para revisar os detalhes do monitor antes de criá-lo.

Examinar e criar

A última etapa na criação de um monitor e das sondas é revisar os detalhes do monitor e das sondas. Nesse momento, você poderá alterar quaisquer informações. Quando tiver terminado de

revisar e tiver criado o monitor e as métricas começarem a ser rastreadas, você começará a receber cobranças por quaisquer sondas.

> **Important**
>
> - Essa etapa foi projetada para ser concluída de uma só vez ao se criar um monitor e uma sonda. Não será possível salvar trabalhos em andamento para continuar mais tarde.
> - Caso opte por editar qualquer seção, você precisará passar pela criação do monitor começando no ponto em que estiver editando. No entanto, você não precisará recriar qualquer etapa subsequente. Essas páginas mantêm as informações preenchidas anteriormente.

Como revisar e criar um monitor

1. Na página Revisar e criar sondas, escolha Editar para a seção em que você deseja fazer alterações.
2. Faça as alterações nessa seção.
3. Escolha Próximo.
4. Faça o seguinte:
   - Faça as alterações desejadas nas páginas adicionais do monitor e escolha Próximo até voltar à página Revisar e criar.
   - Se nenhuma outra página exigir alterações, escolha Próximo até voltar à página Revisar e criar.
5. Escolha Criar monitor.

   A página Network Monitor exibe o estado atual da criação do monitor na seção Monitores de rede. Durante a criação do monitor, o Estado é Pendente. Quando o Estado muda para Ativo, você pode acessar o painel do monitor para visualizar as métricas do CloudWatch.

   Para obter mais informações sobre como trabalhar com o painel do monitor, consulte the section called “Painéis do Network Monitor”.

> **Note**
>
> Pode levar vários minutos para que o monitor de rede recém-adicionado comece a coletar métricas de rede.

Criar um Network Monitor usando a linha de comando ou a API

Use a linha de comando ou a API para visualizar e criar um monitor de rede.

Como criar um monitor de rede usando a linha de comando ou a API

1. Crie um monitor de rede usando create-monitor.
2. Crie uma sonda do monitor de rede usando create-probe.

Trabalhar com monitores e sondas do Network Monitor

Você pode executar qualquer uma das tarefas a seguir com monitores e sondas, usando o console do Amazon CloudWatch, a linha de comando ou a API.

Tópicos:

- Editar um monitor
- Excluir um monitor
- Ativar ou desativar uma sonda
- Adicionar uma sonda a um monitor
- Editar uma sonda
- Excluir uma sonda
- Marcar ou desmarcar recursos usando a linha de comando ou a API

Editar um monitor

Você pode editar quaisquer informações em um Network Monitor, inclusive renomeá-lo, definir um novo período de agregação ou adicionar ou remover tags. Alterar as informações de um monitor não altera as sondas associadas a ele. Você pode editar um monitor usando o console do Amazon CloudWatch, a linha de comando ou a API.

Editar um monitor usando o console

Use o console do CloudWatch para editar um monitor.

Como editar um monitor usando o console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/ e, em seguida, em Monitoramento de rede, escolha Network Monitor.
2. Na seção Monitores de rede, escolha o monitor que você deseja editar.
3. Na página do painel do monitor, escolha Editar.
4. Em Nome do monitor, insira o novo nome do monitor.
5. No Período de agregação, escolha com que frequência você deseja enviar métricas para o CloudWatch. Os períodos válidos são:

   - 30 segundos
   - 60 segundos

   > **Note**
   >
   > Um período de agregação mais curto fornece uma detecção mais rápida de problemas de rede. No entanto, o período de agregação escolhido pode afetar sua estrutura de faturamento. Para obter mais informações sobre a definição de preços, consulte a página Definição de preços do Amazon CloudWatch.

6. (Opcional) Na seção Tags, adicione pares de Chave e Valor para ajudar ainda mais a identificar esse recurso, permitindo que você pesquise ou filtre informações específicas. Você também pode simplesmente alterar o Valor de qualquer Chave atual.

   1. Selecione Adicionar nova tag.
   2. Insira um nome de Chave e o Valor associado.
   3. Escolha Adicionar nova tag para adicionar essa nova tag.

      Você pode adicionar várias tags escolhendo Adicionar nova tag ou remover qualquer tag escolhendo Remover.
   4. Se você quiser associar as tags ao monitor, mantenha a opção Adicionar tags às sondas criadas pelo monitor marcada. Isso adiciona as tags às sondas do monitor, o que pode ser útil se você estiver usando autenticação ou medição baseada em tags.

7. Escolha Salvar alterações.

Editar um monitor usando a CLI ou a API

Use a linha de comando ou a API para visualizar e criar um monitor.

Como editar um monitor usando a linha de comando ou a API

1. Use list-monitors para obter uma lista dos monitores se você não souber o nome do monitor. Anote o nome do monitor que você deseja editar.
2. Use edit-monitor usando o nome do monitor da etapa anterior.

Excluir um monitor

Antes de excluir um monitor, você deve desativar ou excluir todas as sondas associadas a esse monitor, independentemente do Estado do monitor. Depois que um monitor for desativado ou excluído, você não receberá mais cobranças por essas sondas do monitor. Não é possível restaurar um monitor excluído. Você pode excluir um monitor usando o console do Amazon CloudWatch ou a linha de comando ou API.

Embora uma sonda possa ser excluída ou desativada, o CloudWatch ainda retém as métricas por 15 dias.

Excluir um monitor usando o console

Use o console do CloudWatch para excluir um monitor.

Como excluir um monitor usando o console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/ e, em seguida, em Monitoramento de rede, escolha Network Monitor.
2. Na seção Monitores de rede, escolha o monitor que você deseja excluir.
3. Escolha Ações e, em seguida, escolha Excluir.
4. Se você tiver sondas ativas, será solicitado a desativá-las. Escolha Desativar sondas.

> **Note**
>
> Não é possível cancelar ou desfazer essa ação depois de escolher Desativar sondas. No entanto, as sondas desativadas não são removidas do monitor. Você poderá reativá-las posteriormente. Consulte the section called “Ativar ou desativar uma sonda”.

5. Insira **`confirm`** no campo de confirmação e escolha Excluir.

Excluir um monitor usando a linha de comando ou a API

Exclua um monitor usando a linha de comando ou a API.

Como excluir um monitor de rede usando a linha de comando ou a API

1. Você precisará do nome do monitor que deseja excluir. Se você não souber o nome, use list-monitors para obter uma lista dos seus monitores. Anote o nome do monitor que você deseja excluir.
2. Verifique se esse monitor contém alguma sonda. Use get-monitor usando o nome do monitor da etapa anterior. Isso retorna uma lista de todas as sondas associadas a esse monitor.
3. Se o monitor contiver sondas, primeiro você precisará definir essas sondas como inativas ou excluí-las.
   - Para definir uma sonda como inativa, use update-probe e defina o estado como `INACTIVE`.
   - Para excluir uma sonda, use delete-probe.
4. Depois que as sondas forem definidas como `INACTIVE` ou excluídas, use delete-monitor para excluir o monitor. Sondas inativas não são excluídas.

Ativar ou desativar uma sonda

Você pode ativar ou desativar uma sonda de monitor conforme necessário. Você talvez queira desativar uma sonda caso não a esteja usando no momento, mas talvez possa querer usá-la novamente no futuro. Ao desativar uma sonda, você não precisará perder tempo configurando-a novamente. Você não será cobrado pelas sondas desativadas.

Você pode editar o status de um monitor usando o console do Amazon CloudWatch, a linha de comando ou a API.

Definir uma sonda como ativa ou inativa usando o console

Use o console do CloudWatch para definir uma sonda como ativa ou inativa.

Como definir uma sonda como ativa ou inativa usando o console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/ e, em seguida, em Monitoramento de rede, escolha Network Monitor.
2. Escolha a guia Detalhes do monitor.
3. Na seção Sondas, escolha a sonda que você deseja ativar ou desativar.
4. Escolha Ações e depois Ativar ou Desativar.

> **Note**
>
> Caso esteja reativando uma sonda desativada, você começará a receber cobranças relativas a essa sonda.

Definir uma sonda como ativa ou inativa usando a linha de comando ou a API

Defina uma sonda como ativa ou inativa ou desative-a usando a linha de comando ou a API. Você só pode usar esse comando para uma única sonda.

Como definir uma sonda como ativa ou inativa usando a linha de comando ou a API

1. Use list-monitors para obter uma lista dos monitores se você não souber o nome do monitor. Anote o nome do monitor cujo status da sonda você deseja alterar.
2. Use get-monitor usando o nome do monitor da etapa anterior. Isso retorna uma lista de todas as sondas associadas a esse monitor. Anote os IDs das sondas cujos status você deseja alterar.
3. Use update-probe e defina a sonda cujo estado você deseja alterar para `ACTIVE` ou `INACTIVE`.

Adicionar uma sonda a um monitor

Você pode adicionar uma sonda a um monitor existente. Observe que, se você adicionar qualquer sonda a um monitor, sua estrutura de faturamento será atualizada para mostrar que uma nova sonda foi adicionada.

Adicionar uma sonda a um monitor usando o console

Como adicionar uma sonda a um monitor usando o console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/ e, em seguida, em Monitoramento de rede, escolha Network Monitor.
2. Na guia Monitores de rede, siga um dos seguintes procedimentos:
   - Escolha o link Nome do monitor ao qual você deseja adicionar uma sonda. Escolha a guia Detalhes do monitor e, na seção Sondas, escolha Adicionar sonda.
   - Escolha a caixa de seleção do monitor, escolha Ações e, em seguida, Adicionar sonda.
3. Na página Adicionar sonda, faça o seguinte:
   1. Em Origem da rede da AWS, escolha uma sub-rede para adicionar ao monitor.

      > **Note**
      >
      > Você só pode adicionar uma sonda por vez e até quatro sondas por monitor.

   2. Insira o endereço IP de destino da rede on-premises. Os endereços IPv4 e IPv6 são compatíveis.
   3. Selecione Advanced settings (Configurações avançadas).
   4. Escolha o Protocolo da rede para o destino. Pode ser ICMP ou TCP.
   5. Se o Protocolo for TCP, insira as informações a seguir. Caso contrário, pule para a próxima etapa:
      - Insira a Porta que a rede usa para se conectar. A porta deve ter um número de 1 a 65535.
      - Insira o Tamanho do pacote. Esse é o tamanho, em bytes, de cada pacote enviado com a sonda entre a origem e o destino. O tamanho do pacote deve ser um número de 56 a 8.500.
4. (Opcional) Na seção Tags, adicione pares de Chave e Valor para ajudar ainda mais a identificar esse recurso, permitindo que você pesquise ou filtre informações específicas.
   1. Selecione Adicionar nova tag.
   2. Insira um nome de Chave e o Valor associado.
   3. Escolha Adicionar nova tag para adicionar a nova tag.

Você pode adicionar várias tags escolhendo Adicionar nova tag ou remover qualquer tag escolhendo Remover.

5. Escolha Adicionar sonda.

   Enquanto a sonda está sendo ativada, o Estado mostra Pendente. Pode levar vários minutos para que a sonda se torne Ativa.

Adicionar uma sonda ao monitor usando a linha de comando ou a API

Adicione uma sonda ao monitor usando a linha de comando ou a API. Você só pode usar esse comando para adicionar uma única sonda de cada vez.

Como adicionar uma sonda ao monitor usando a linha de comando ou a API

1. Use list-monitors para obter uma lista dos monitores se você não souber o nome do monitor. Anote o nome do monitor ao qual você deseja adicionar uma sonda.
2. Use create-probe para adicionar uma sonda ao monitor.

Editar uma sonda

Você pode alterar quaisquer informações de uma sonda atual, independentemente de essa sonda estar ativada ou desativada. Você pode editar uma sonda usando o console do Amazon CloudWatch, a linha de comando ou a API.

Editar uma sonda usando o console

Como editar uma sonda usando o console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/ e, em seguida, em Monitoramento de rede, escolha Network Monitor.

   Escolha o link do Nome para abrir o painel do monitor.
2. Escolha a guia Detalhes do monitor.
3. Na seção Sondas, escolha o link para a sonda que você deseja editar.
4. Na página do painel da sonda, escolha Editar ou escolha Ações e, em seguida, escolha Editar.
5. Na página Editar sonda, insira o novo endereço IP da sonda de destino. Os endereços IPv4 e IPv6 são compatíveis.

6. Selecione Advanced settings (Configurações avançadas).
7. Escolha o Protocolo da rede. Pode ser ICMP ou TCP.
8. Se o Protocolo for TCP, insira as informações a seguir. Caso contrário, pule para a próxima etapa:
   - Insira a Porta que a rede usa para se conectar. A porta deve ter um número de 1 a 65535.
   - Insira o Tamanho do pacote. Esse é o tamanho, em bytes, de cada pacote enviado com a sonda entre a origem e o destino. O tamanho do pacote deve ser um número de 56 a 8.500.
9. (Opcional) Na seção Tags, adicione pares de Chave e Valor para ajudar ainda mais a identificar esse recurso, permitindo que você pesquise ou filtre informações específicas.
   1. Selecione Adicionar nova tag.
   2. Insira um nome de Chave e o Valor associado.
   3. Escolha Adicionar nova tag para adicionar a nova tag.

      Você pode adicionar várias tags escolhendo Adicionar nova tag ou remover qualquer tag escolhendo Remover.
10. Escolha Salvar alterações.

Editar uma sonda usando a linha de comando ou a API

Use a linha de comando para editar a sonda de um monitor. Você só pode usar esse comando para uma única sonda.

Como editar uma sonda usando a linha de comando ou a API

1. Use list-monitors para obter uma lista dos monitores se você não souber o nome do monitor. Anote o nome do monitor cujo status da sonda você deseja alterar.
2. Use get-monitor usando o nome do monitor da etapa anterior. Isso retorna uma lista de todas as sondas associadas a esse monitor. Anote o ID da sonda que você deseja editar.
3. Use update-probe para alterar as informações da sonda.

Excluir uma sonda

Você pode excluir uma sonda em vez de desativá-la se souber que não precisará dela novamente no futuro. Você não pode recuperar uma sonda excluída, então vai precisar recriá-la. Quando uma

sonda é excluída, a cobrança relativa a ela é interrompida. Você pode excluir uma sonda usando o console do Amazon CloudWatch, a linha de comando ou a API.

Excluir uma sonda usando o console

Como excluir uma sonda usando o console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/ e, em seguida, em Monitoramento de rede, escolha Network Monitor.
2. Na seção Monitores de rede, escolha o link do Nome para abrir o painel do monitor.
3. Escolha a guia Detalhes do monitor.
4. Escolha a caixa de seleção do monitor, depois Ações e, em seguida, escolha Excluir.
5. Na caixa de diálogo Excluir sonda, escolha Excluir para confirmar que você deseja excluir a sonda.
6. Escolha Excluir para confirmar que você deseja excluir a sonda.

   O Estado da sonda na seção Sondas mostra Excluindo. Depois de excluída, a sonda é removida da seção Sondas.

Excluir uma sonda usando a linha de comando ou a API

Exclua uma sonda usando a linha de comando ou a API. Você só pode usar esse comando para uma única sonda.

Como definir uma sonda como ativa ou inativa usando a linha de comando ou a API

1. Use list-monitors para obter uma lista dos monitores se você não souber o nome do monitor. Anotar o nome do monitor associado à sonda que você deseja excluir
2. Use get-monitor usando o nome do monitor da etapa anterior. Isso retorna uma lista de todas as sondas associadas a esse monitor. Anote o ID da sonda que você deseja excluir.
3. Use delete-probe.

Marcar ou desmarcar recursos usando a linha de comando ou a API

Você pode usar a linha de comando ou a CLI para adicionar ou atualizar tags de recursos.

Como atualizar tags do monitor de rede usando a linha de comando ou a API

- Para listar tags de recursos, use list-tags-for-resources.
- Para marcar um recurso, use tag-resource.
- Para desmarcar um recurso, use untag-resource.

Painéis do Network Monitor

Você pode usar o painel do Network Monitor do Amazon CloudWatch para visualizar a integridade da rede da AWS e investigar o tempo de ida e volta e a perda de pacotes. Você pode visualizar essas métricas tanto para monitores quanto para sondas individuais.

Painéis do Network Monitor

- Painel do monitor
- Painel de sondas

Alarmes de sonda

É possível criar alarmes do Amazon CloudWatch com base nas métricas do Amazon CloudWatch Network Monitor, da mesma forma que é possível criar para outras métricas do Amazon CloudWatch. Qualquer alarme que você criar aparecerá na coluna de Status da sonda na seção de Detalhes do monitor do painel do Network Monitor quando o alarme for acionado. O status será OK ou Em alarme. Se nenhum status for exibido para uma sonda, nenhum alarme foi criado para essa sonda.

Por exemplo, é possível criar um alarme com base na métrica `PacketLoss` do Network Monitor e configurá-lo para enviar uma notificação quando a métrica for maior do que o valor escolhido. Configure alarmes para métricas do Network Monitor seguindo as mesmas diretrizes de outras métricas do CloudWatch.

As métricas a seguir estão disponíveis em `AWS/NetworkMonitor` quando um alarme do CloudWatch para o Network Monitor é criado.

- HealthIndicator
- PacketLoss
- RTT (tempo de ida e volta)

Para conhecer as etapas para criar um alarme do Network Monitor, consulte the section called “Criar um alarme com base em um limite estático”.

Definir um período de métricas

As métricas e os eventos nos dois painéis usam um tempo padrão de duas horas, calculado a partir da hora atual. É possível alterar o padrão para usar uma das seguintes predefinições:

- 1h: uma hora
- 2h: duas horas
- 1d: um dia
- 1w: uma semana

Também é possível definir um período personalizado. Escolha Personalizado, escolha uma hora Absoluta ou Relativa e, em seguida, defina o período para uma hora de sua escolha. O tempo relativo aceita apenas 15 dias retroativamente com base na data atual, de acordo com os padrões do CloudWatch.

Além disso, você pode escolher a hora exibida nos gráficos com base no fuso horário UTC ou no fuso horário local.

Painel do monitor

Você pode usar o painel do Network Monitor do Amazon CloudWatch para visualizar a integridade da rede da AWS e investigar o tempo de ida e volta e a perda de pacotes. O Network Monitor tem painéis para monitores e sondas.

Como acessar um painel de um monitor

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/ e, em seguida, em Monitoramento de rede, escolha Network Monitor.
2. Na seção Monitores de rede, escolha o link do Nome para abrir o painel do monitor.

Visão geral

A página Visão geral exibe as seguintes informações para o monitor:

- Integridade da rede da AWS: a integridade da rede da AWS só exibe a integridade geral da rede da AWS. O status será Íntegro ou Degradado. Um status Íntegro indica que o Network Monitor não

observou qualquer problema com a rede da AWS. Um status Degradado indica que o Network Monitor observou um problema com a rede da AWS. A barra de status nesta seção mostra o status da rede durante um tempo padrão de uma hora. Passe o cursor sobre qualquer ponto da barra de status para ver detalhes adicionais.

- Resumo do tráfego da sonda: exibe o estado atual do tráfego entre as sub-redes de origem da AWS no monitor e os endereços IP de destino. O Resumo do tráfego da sonda exibe o seguinte:
  - Sondas em alarme: esse número indica quantas sondas estão em estado degradado. Um alarme é acionado quando uma métrica que você configurou como alarme é acionada. Para obter informações sobre alarmes de métrica do Network Monitor, consulte the section called “Alarmes de sonda”.
  - Perda de pacotes: o número de pacotes que foram perdidos entre a sub-rede de origem e o endereço IP de destino. Isso é representado como uma porcentagem do total de pacotes enviados.
  - Tempo de ida e volta: o tempo necessário, em milissegundos, para que um pacote da sub-rede de origem atinja o endereço IP de destino e volte novamente.

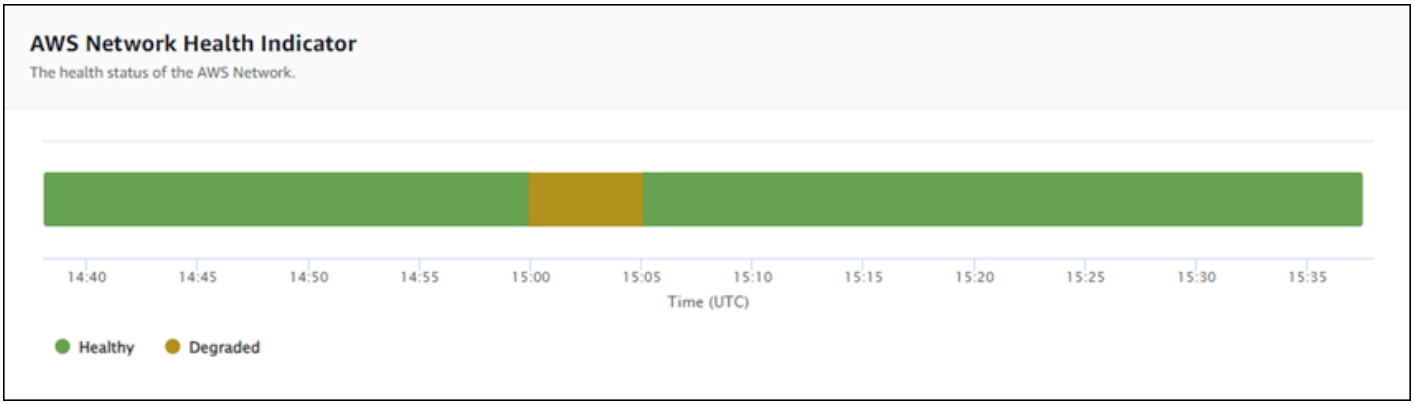
Os dados são representados por um gráfico interativo, permitindo que você veja os detalhes.

Por padrão, os dados são exibidos para um período de duas horas, calculado a partir da data e hora atuais. No entanto, você pode alterar o intervalo de acordo com as suas necessidades. Para ter mais informações, consulte the section called “Definir um período de métricas”.

Métricas de rastreamento

O painel do Network Monitor exibe uma representação gráfica dos monitores e das sondas. Os gráficos a seguir estão disponíveis:

- Indicador de integridade da rede da AWS: representa a integridade da rede da AWS durante um período especificado. O status será Íntegro ou Degradado. No exemplo a seguir, você verá que das 15:00 UTC às 15:05 UTC, a rede da AWS estava em um estado degradado. Depois das 15:05, a rede voltou para um estado íntegro. Você pode passar o cursor sobre qualquer seção do gráfico para visualizar detalhes adicionais.



> **Note**
>
> O indicador de integridade da rede não indica a integridade da sonda, somente da rede da AWS.

- Perda de pacotes: esse gráfico exibe uma linha exclusiva mostrando a porcentagem de perda de pacotes para cada sonda no monitor. A legenda na parte inferior da página exibe cada uma das sondas no monitor, codificada por cores para garantir a exclusividade. Passar o cursor sobre uma sonda nesse gráfico exibe a sub-rede de origem, o IP de destino e a porcentagem de perda de pacotes. No exemplo a seguir, um alarme de perda de pacotes foi configurado para uma sonda de uma sub-rede para o endereço IP 127.0.0.1. O alarme foi acionado quando o limite de perda de pacotes foi excedido para a sonda. Passar o cursor sobre o gráfico mostra a origem e o destino da sonda e mostra que houve uma perda de pacotes de 30,97% dessa sonda em 21 de novembro às 02:41:30.

- Tempo de ida e volta: esse gráfico exibe uma linha para cada sonda, mostrando o tempo de ida e volta para cada sonda. A legenda na parte inferior da página exibe cada uma das sondas no monitor, codificada por cores para garantir a exclusividade. Passar o cursor sobre uma sonda nesse gráfico exibe a sub-rede de origem, o endereço IP de destino e o tempo de ida e volta. O exemplo a seguir mostra que, na terça-feira, 21 de novembro, às 21:45:30, o tempo de ida e volta para uma sonda de uma sub-rede para o endereço IP 127.0.0.1 foi de 0,075 segundo.

Detalhes do monitor

A página Detalhes do monitor exibe os detalhes sobre o monitor, incluindo as sondas. Nessa página, você pode gerenciar tags ou adicionar uma sonda. A página está dividida nas três seções abaixo:

- Detalhes do monitor: esta página fornece detalhes sobre o monitor. As informações mostradas nessa seção não podem ser editadas. No entanto, você pode escolher o link do Nome do perfil para ver detalhes do perfil vinculado ao serviço do Network Monitor.
- Sondas: esta seção exibe uma lista de todas as sondas associadas ao monitor. Escolha um link de VPC ou ID de sub-rede para abrir os detalhes da VPC ou da sub-rede no console da Amazon VPC. Você também pode modificar uma sonda, inclusive ativá-la ou desativá-la. Para ter mais informações, consulte the section called “Trabalhar com monitores e sondas”.

  A seção Sondas exibe informações sobre cada sonda configurada para esse monitor, incluindo o ID, o ID da VPC, o ID da sub-rede, o Endereço IP, o Protocolo e se o Estado da sonda é Ativo ou Inativo. Se você configurou um alarme para uma sonda, o Status atual desse alarme será exibido. OK indica que não há nenhum evento de métrica que tenha acionado nenhum alarme; Em alarme indica que uma métrica que você configurou no CloudWatch acionou um alarme. Se nenhum status for exibido para uma sonda, nenhum alarme do CloudWatch foi configurado. Para obter informações sobre os tipos de alarmes de sonda do Network Monitor que você pode criar, consulte the section called “Alarmes de sonda”.
- Tags: visualize as tags atuais de um monitor. É possível adicionar ou remover tags escolhendo Gerenciar tags. Isso abre a página Editar sonda. Para obter mais informações sobre a edição de tags, consulte the section called “Editar um monitor”.

Painel de sondas

Você pode usar o painel do Amazon CloudWatch Network Monitor para visualizar a integridade da rede da AWS e informações sobre um tempo de ida e volta específico e a perda de pacotes para sondas específicas. Há dois painéis de sondas, Visão geral e Detalhes da sonda.

Você pode criar alarmes do CloudWatch para definir limites de métricas para perda de pacotes e tempo de ida e volta. Quando o limite de uma métrica é atingido, um alarme do CloudWatch avisa você. Para obter informações sobre como criar alarmes de sonda, consulte the section called “Alarmes de sonda”.

Como acessar o painel de uma sonda

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/ e, em seguida, em Monitoramento de rede, escolha Network Monitor.
2. Na seção Monitores de rede, escolha o link do Nome para abrir o painel do monitor.
3. Escolha o link do ID para visualizar o painel dessa sonda.

Visão geral

A página Visão geral exibe as seguintes informações para a sonda:

- Detalhes do indicador de integridade da rede da AWS: isso só fornece a integridade geral da rede da AWS. O status será Íntegro ou Degradado. O status Degradado indica que há um problema com a rede da AWS e não indica se há algum problema com a sonda.
- Perda de pacotes: o número de pacotes que foram perdidos entre a sub-rede de origem e o endereço IP de destino para essa sonda.
- Tempo de ida e volta: o tempo que foi necessário, em milissegundos, para que um pacote da sub-rede de origem atingisse o endereço IP de destino e voltasse novamente.

Detalhes da sonda

A página Detalhes da sonda exibe os detalhes sobre uma sonda. Nessa página, você pode editar a sonda. Para ter mais informações, consulte the section called “Trabalhar com monitores e sondas”.

- Detalhes da sonda: esta página fornece informações gerais sobre a sonda. As informações mostradas nessa seção não podem ser editadas.
- Origem e destino da sonda: esta seção exibe detalhes sobre a sonda. Escolha um link de VPC ou ID de sub-rede para abrir os detalhes da VPC ou da sub-rede no console da Amazon VPC. Você também pode modificar uma sonda, inclusive ativá-la ou desativá-la.
- Tags: visualize as tags atuais de um monitor. É possível adicionar ou remover tags escolhendo Gerenciar tags. Isso abre a página Editar sonda. Para obter mais informações sobre a edição de tags, consulte the section called “Editar uma sonda ”.

Cotas do Network Monitor

Abaixo, são mostradas as cotas do Network Monitor:

Cota	Padrão	Ajustável
Número máximo de monitores por conta por Região da AWS	100	Sim
Número máximo de sondas por monitor	24	Sim
Número máximo de sondas por sub-rede por monitor	4	Sim

Segurança e proteção de dados no Network Monitor

A segurança para com a nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se beneficiará de data centers e arquiteturas de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O modelo de responsabilidade compartilhada descreve isso como a segurança da nuvem e segurança na nuvem:

- Segurança da nuvem: AWS é responsável pela proteção da infraestrutura que executaAWS produtos da Nuvem AWS na AWS. A também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos AWSProgramas de conformidade. Para saber mais sobre os programas de conformidade que se aplicam ao Amazon CloudWatch Network Monitor, consulte Serviços da AWS no escopo por programa de conformidade.
- Segurança na nuvem: sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Essa documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada quando usar o CloudWatch Network Monitor. Os tópicos a seguir mostram como configurar o CloudWatch Network Monitor para atender aos seus objetivos de segurança e conformidade. Você também saberá como usar outros serviços da AWS que ajudam a monitorar e proteger os recursos do CloudWatch Network Monitor.

Tópicos

- Proteção de dados no Amazon CloudWatch Network Monitor
- Segurança de infraestrutura no Amazon CloudWatch Network Monitor

Proteção de dados no Amazon CloudWatch Network Monitor

O modelo de responsabilidade compartilhada da AWS se aplica à proteção de dados no Amazon CloudWatch Network Monitor. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para ter mais informações sobre a proteção de dados na Europa, consulte a AWSpostagem do blog Shared Responsibility Model and GDPR no AWSBlog de segurança da.

Para fins de proteção de dados, recomendamos que você proteja asConta da AWS credenciais da e configure as contas de usuário individuais com o AWS IAM Identity Center ou o AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os atributos da AWS. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure o registro em log das atividades da API e do usuário com o .AWS CloudTrail
- Use AWS as soluções de criptografia da , juntamente com todos os controles de segurança padrão dos Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comandos ou uma API, use um endpoint do FIPS. Para ter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de email dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso vale para quando você trabalha com o CloudWatch Network Monitor

ou outros Serviços da AWS que usam o console, a API, a AWS CLI ou SDKs da AWS. Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendemos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Segurança de infraestrutura no Amazon CloudWatch Network Monitor

Como serviço gerenciado, o Amazon CloudWatch Network Monitor é protegido pelos procedimentos de segurança da rede global da AWS descritos no whitepaper Amazon Web Services: Overview of Security Processes.

Você usa as chamadas de API publicadas da AWS para acessar o CloudWatch Network Monitor por meio da rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos usar o TLS 1.2 ou posterior. Os clientes também devem ter suporte a conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas utilizando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Gerenciamento de identidade e acesso para o Amazon CloudWatch Network Monitor

O AWS Identity and Access Management (IAM) é um serviço da AWS que ajuda administradores a controlar o acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode ser autenticado (fazer login) e autorizado (ter permissões) para usar os recursos do CloudWatch Network Monitor. O IAM é um serviço da AWS que pode ser usado sem custo adicional. Você pode usar recursos do IAM para permitir que outros usuários, serviços e aplicações usem seus recursos da AWS totalmente ou de maneira limitada, sem compartilhar suas credenciais de segurança.

Por padrão, os usuários do IAM não têm permissão para criar, visualizar ou modificar os recursos da AWS. Para permitir que um usuário do IAM acesse recursos, como uma rede global, e execute tarefas, veja o que é necessário fazer:

- Criar uma política do IAM que conceda permissão ao usuário para usar os recursos específicos e ações de API de que ele precisa
- Anexar a política ao usuário do IAM ou ao grupo ao qual o usuário pertence

Quando você anexa uma política a um usuário ou a um grupo de usuários, isso concede ou nega ao usuário permissões para realizar as tarefas especificadas nos recursos especificados.

Chaves de condição

O elemento `Condition` (ou bloco Condition) permite que você especifique condições nas quais uma instrução está em vigor. O elemento Condition é opcional. É possível criar expressões condicionais que usam operadores de condição, como "igual a" ou "menor que", para fazer a condição da política corresponder aos valores na solicitação. Para obter mais informações, consulte Elementos de política JSON do IAM: operadores de condição no Guia do usuário do AWS Identity and Access Management.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação lógica `OR`. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM.

Você pode anexar tags a recursos do CloudWatch Network Monitor ou passar tags em uma solicitação ao Cloud WAN. Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as chaves de condição `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`. Consulte Elementos de política JSON do IAM: Condition no Guia do usuário do AWS Identity and Access Management para obter mais informações.

Para ver todas as chaves de condição globais da AWS, consulte Chaves de contexto de condição globais da AWS no Guia do usuário do AWS Identity and Access Management.

Marcar os principais recursos da rede

Uma tag é um rótulo de metadados que você ou a AWS atribui a um recurso da AWS. Cada tag consiste em uma chave e um valor. Para tags atribuídas por você, é possível definir a chave e o valor. Por exemplo, você pode definir a chave como `purpose` e o valor como `test` para um recurso. As tags ajudam você a fazer o seguinte:

- Identificar e organizar seus recursos da AWS. Muitos serviços da AWS oferecem suporte à marcação para que você possa atribuir a mesma tag a recursos de diferentes serviços para indicar que os recursos estão relacionados.
- Controle o acesso aos recursos da AWS. Para obter mais informações, consulte Controlar o acesso a recursos da AWS usando tags no Guia do usuário do AWS Identify and Access Management.

Como o Amazon CloudWatch Network Monitor opera com o IAM

Antes de usar o IAM para gerenciar o acesso ao CloudWatch Network Monitor, saiba quais recursos do IAM estão disponíveis para uso com o CloudWatch Network Monitor.

Recursos do IAM que podem ser usados com o Amazon CloudWatch Network Monitor

Atributo do IAM	Suporte ao CloudWatch Network Monitor
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não
Ações de políticas	Sim
atributos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Não
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Permissões de entidade principal	Sim

Atributo do IAM	Suporte ao CloudWatch Network Monitor
Perfis de serviço	Não
Funções vinculadas ao serviço	Sim

Para obter uma visão geral de como o CloudWatch Network Monitor e outros serviços da AWS funcionam com a maioria dos recursos do IAM, consulte Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Políticas baseadas em identidade do Amazon CloudWatch Network Monitor

É compatível com políticas baseadas em identidade	Sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou atributos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Não é possível especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou função à qual ela está anexado. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elementos da política JSON do IAM no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para o CloudWatch Network Monitor

Para visualizar exemplos de políticas baseadas em identidade do CloudWatch Network Monitor, consulte Exemplos de políticas baseadas em identidade do Amazon CloudWatch.

Políticas baseadas em recursos no CloudWatch Network Monitor

Oferece suporte a políticas baseadas em recursos	Não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de função do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. As entidades principais podem incluir contas, usuários, funções, usuários federados ou Serviços da AWS.

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em atributo. Adicionar uma entidade principal entre contas à política baseada em atributo é apenas metade da tarefa de estabelecimento da relação de confiança. Quando a entidade principal e o atributo estão em diferentes Contas da AWS, um administrador do IAM da conta confiável também deve conceder à entidade principal (usuário ou perfil) permissão para acessar o atributo. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em atributo conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Ações de políticas do CloudWatch Network Monitor

Oferece suporte a ações de políticas	Sim

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista das ações do CloudWatch Network Monitor, consulte Ações definidas pelo Amazon CloudWatch Network Monitor na Referência de autorização do serviço.

As ações de políticas no CloudWatch Network Monitor usam o seguinte prefixo antes da ação:

```
networkmonitor
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [
      "networkmonitor:action1",
      "networkmonitor:action2"
         ]
```

Para visualizar exemplos de políticas baseadas em identidade do CloudWatch Network Monitor, consulte Exemplos de políticas baseadas em identidade do Amazon CloudWatch.

Recursos de políticas do CloudWatch Network Monitor

Oferece suporte a atributos de políticas	Sim

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando Nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem suporte a um tipo de atributo específico, conhecido como permissões em nível de atributo.

Para ações que não oferecem suporte a permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de recursos do CloudWatch Network Monitor e seus ARNs, consulte Tipos de recursos definidos pelo Amazon CloudWatch Network Monitor na Referência de autorização do serviço. Para saber com quais ações é possível especificar o ARN de cada recurso, consulte Ações definidas pelo Amazon CloudWatch Network Monitor.

Chaves de condição de políticas do Amazon CloudWatch Network Monitor

Compatível com chaves de condição de política específicas do serviço	Sim

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou `Condition` bloco de) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação lógica `OR`. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos de política do IAM: variáveis e tags no Guia do usuário do IAM.

A AWS oferece suporte a chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as AWS chaves de condição globais da , consulte AWSChaves de contexto de condição globais da no Guia do usuário do IAM.

Para ver uma lista de chaves de condição do CloudWatch Network Monitor, consulte Chaves de condição do Amazon CloudWatch Network Monitor na Referência de autorização do serviço. Para saber com quais ações e recursos é possível usar uma chave de condição, consulte Ações definidas pelo Amazon CloudWatch Network Monitor.

ACLs no CloudWatch Network Monitor

Oferece suporte a ACLs	Não

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Controle de acesso baseado em atributos (ABAC) com o CloudWatch Network Monitor

Oferece suporte a ABAC (tags em políticas)	Parcial

O controle de acesso baseado em recurso (ABAC) é uma estratégia de autorização que define permissões com base em recursos. Na AWS, esses atributos são chamados de tags. É possível anexar tags a entidades do IAM (usuários ou perfis) e a muitos atributos da AWS. A marcação de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do atributo que ela está tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as chaves de condição `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de atributo, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de atributos, o valor será Parcial.

Para obter mais informações sobre o ABAC, consulte O que é ABAC? no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

Usar credenciais temporárias com o CloudWatch Network Monitor

Oferece suporte a credenciais temporárias	Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Você está usando credenciais temporárias se faz login no AWS Management Console usando qualquer método, exceto um nome de usuário e uma senha. Por exemplo, quando você acessa a AWS usando o link de autenticação única (SSO) da empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar perfis, consulte Alternar para um perfil (console) no Guia do usuário do IAM.

Você pode criar credenciais temporárias manualmente usando a AWS CLI ou a API da AWS. Em seguida, você pode usar essas credenciais temporárias para acessar aAWS .AWS A recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para mais informações, consulte Credenciais de segurança temporárias no IAM.

Permissões de entidade principal entre serviços do CloudWatch Network Monitor

Suporte para o recurso Encaminhamento de sessões de acesso (FAS)	Sim

Quando você usa um usuário ou um perfil do IAM para executar ações na AWS, você é considerado uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

Perfis de serviço para o CloudWatch Network Monitor

Compatível com perfis de serviço	Não

Um perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do usuário do IAM.

> **Warning**
>
> Alterar as permissões de um perfil de serviço pode interromper a funcionalidade do CloudWatch Network Monitor. Edite perfis de serviço somente quando o CloudWatch Network Monitor fornecer orientação para isso.

Usar um perfil vinculado ao serviço para o CloudWatch Network Monitor

Oferece suporte a perfis vinculados ao serviço	Sim

Um perfil vinculado ao serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode assumir a função de executar uma ação em seu nome. Os perfis vinculados ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar funções vinculadas a serviços, consulte Serviços do AWS que funcionam com o IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna Função vinculada ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a esse serviço.

Exemplos de políticas baseadas em identidade para o CloudWatch Network Monitor

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do CloudWatch Network Monitor. Eles também não podem executar tarefas usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou a API AWS. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis, e os usuários podem assumir os perfis.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documento de política JSON, consulte Criação de políticas do IAM no Guia do Usuário do IAM.

Para obter detalhes sobre ações e tipos de recurso definidos pelo CloudWatch Network Monitor, como o formato dos ARNs para cada um dos tipos de recurso, consulte Ações, recursos e chaves de condição do Amazon CloudWatch Network Monitor na Referência de autorização do serviço.

Tópicos

- Melhores práticas de política
- Usar o console do CloudWatch Network Monitor
- Permitir que os usuários visualizem suas próprias permissões
- Solução de problemas de identidade e acesso do CloudWatch Network Monitor

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do CloudWatch Network Monitor na sua conta. Essas ações podem incorrer em custos para a Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas gerenciadas pela AWS e avance para as permissões de privilégio mínimo: para começar a conceder permissões a seus usuários e workloads, use as políticas gerenciadas pela AWS que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis na sua Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo cliente AWS específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para perfis de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e atributos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso a ações de serviço, se elas forem usadas por meio de um AWS service (Serviço da AWS) específico, como o AWS CloudFormation. Para obter mais informações, consulte Elementos de política JSON do IAM: condições no Manual do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações acionáveis para ajudar você a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do usuário do IAM.

- Exigir autenticação multifator (MFA): se houver um cenário que exija usuários do IAM ou um usuário raiz em sua Conta da AWS, ative a MFA para obter segurança adicional. Para exigir a MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do usuário do IAM.

Para mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usar o console do CloudWatch Network Monitor

Para acessar o console do Amazon CloudWatch Network Monitor, você deve ter um conjunto mínimo de permissões. Essas permissões devem dar autorização para que você liste e visualize detalhes sobre os recursos do CloudWatch Network Monitor na sua Conta da AWS. Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Não é necessário conceder permissões mínimas do console para usuários que fazem chamadas somente à AWS CLI ou à AWS API. Em vez disso, permita o acesso somente a ações que correspondam a operação de API que estiverem tentando executar.

Para garantir que usuários e perfis ainda possam usar o console do CloudWatch Network Monitor, anexe também a política *ConsoleAccess* ou *ReadOnly* gerenciada pela AWS do CloudWatch Network Monitor às entidades. Para obter mais informações, consulte Adicionando Permissões a um Usuário no Guia do Usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permite que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou de forma programática usando a AWS CLI ou a AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
```

```
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Solução de problemas de identidade e acesso do CloudWatch Network Monitor

Use as informações a seguir para ajudar a diagnosticar e corrigir problemas comuns que você possa encontrar ao trabalhar com o CloudWatch Network Monitor e o IAM.

Tópicos

- Não tenho autorização para executar uma ação no CloudWatch Network Monitor
- Não estou autorizado a executar iam:PassRole
- Quero permitir que as pessoas fora da minha Conta da AWS acessem meus recursos do CloudWatch Network Monitor

Não tenho autorização para executar uma ação no CloudWatch Network Monitor

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo *`my-example-widget`* fictício, mas não tem as permissões `networkmonitor:`*`GetWidget`* fictícias.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 networkmonitor:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso *`my-example-widget`* usando a ação `networkmonitor:`*`GetWidget`*.

Se você precisar de ajuda, entre em contato com seu administrador da AWS. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a executar iam:PassRole

Se você receber uma mensagem de erro informando que você não tem autorização para realizar a ação `iam:PassRole`, as suas políticas deverão ser atualizadas para permitir que você passe um perfil para o CloudWatch Network Monitor.

Alguns Serviços da AWS permitem que você passe um perfil existente para o serviço, em vez de criar um novo perfil de serviço ou perfil vinculado ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM, `marymajor`, tenta usar o console para executar uma ação no CloudWatch Network Monitor. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se você precisar de ajuda, entre em contato com seu administrador da AWS. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que as pessoas fora da minha Conta da AWS acessem meus recursos do CloudWatch Network Monitor

Você pode criar uma função que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em atributos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus atributos.

Para saber mais, consulte:

- Para saber se o CloudWatch Network Monitor oferece suporte a esses recursos, consulte Como o Amazon CloudWatch funciona com o IAM.
- Para saber como conceder acesso a seus recursos em todas as Contas da AWS pertencentes a você, consulte Fornecimento de acesso a um usuário do IAM em outra Conta da AWS pertencente a você no Guia de usuário do IAM.
- Para saber como conceder acesso a seus atributos para terceiros Contas da AWS, consulte Fornecimento de acesso a Contas da AWS pertencentes a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Políticas gerenciadas pela AWS para o CloudWatch Network Monitor

Para adicionar permissões a usuários, grupos e perfis, é mais fácil usar políticas gerenciadas pela AWS do que gravar políticas por conta própria. É necessário tempo e experiência para criar políticas gerenciadas pelo cliente do IAM que fornecem à sua equipe apenas as permissões de que precisam. Para começar rapidamente, é possível usar nossas políticas gerenciadas pela AWS. Essas políticas abrangem casos de uso comuns e estão disponíveis na sua conta da AWS. Para obter mais informações sobre políticas gerenciadas pela AWS, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

Os serviços da AWS mantêm e atualizam políticas gerenciadas pela AWS. Não é possível alterar as permissões em políticas gerenciadas pela AWS. Os serviços ocasionalmente acrescentam permissões adicionais a uma política gerenciada pela AWS para oferecer suporte a novos atributos.

Esse tipo de atualização afeta todas as identidades (usuários, grupos e perfis) em que a política está anexada. É mais provável que os serviços atualizem uma política gerenciada pela AWS quando um novo atributo for iniciado ou novas operações se tornarem disponíveis. Os serviços não removem permissões de uma política gerenciada pela AWS, portanto, as atualizações de políticas não suspendem suas permissões existentes.

Além disso, a AWS oferece suporte a políticas gerenciadas para perfis de trabalho que abrangem vários serviços. Por exemplo, o `ReadOnlyAccess` da política gerenciada pela AWS, concede acesso somente leitura a todos os atributos e serviços da AWS. Quando um serviço executa um novo atributo, a AWS adiciona permissões somente leitura para novas operações e atributos. Para obter uma lista e descrições das políticas de perfis de trabalho, consulte Políticas gerenciadas pela AWS para perfis de trabalho no Guia do usuário do IAM.

Política gerenciada pela AWS: CloudWatchNetworkMonitorServiceRolePolicy

O `CloudWatchNetworkMonitorServiceRolePolicy` é anexado a um perfil vinculado ao serviço que permite que o serviço execute ações em seu nome e acesse os recursos associados ao CloudWatch Network Monitor. Não é possível anexar essa política às suas identidades do IAM. Para ter mais informações, consulte the section called “Funções vinculadas ao serviço”.

O CloudWatch Network Monitor é atualizado para políticas gerenciadas pela AWS

Veja detalhes sobre atualizações nas políticas gerenciadas pela AWS para o CloudWatch Network Monitoring desde que esse serviço começou a rastrear essas alterações, em novembro de 2023.

Alteração	Descrição	Data
CloudWatchNetworkMonitorServiceRolePolicy: nova política.	Nova política adicionada ao CloudWatch Network Monitor.	27 de novembro de 2023
the section called “`AWSServiceRoleForNetworkMonitor`”. Novo perfil.	Novo perfil adicionado ao CloudWatch Network Monitor.	27 de novembro de 2023

Permissões do IAM para o CloudWatch Network Monitor

Para usar o Amazon CloudWatch Network Monitor, os usuários devem ter as permissões corretas.

Para obter mais informações sobre segurança no Amazon CloudWatch, consulte Gerenciamento de Identidade e Acesso para o Amazon CloudWatch.

Permissões requeridas para visualizar um monitor

Para visualizar um monitor do Amazon CloudWatch Network Monitor no AWS Management Console, você deve ter feito login como um usuário ou perfil que tenha as seguintes permissões:

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "networkmonitor:Get*",
                "networkmonitor:List*"
                ],
            "Resource": "*"
        }
    ]
}
```

Permissões requeridas para criar um monitor

Para criar um monitor no Amazon CloudWatch Network Monitor, os usuários devem ter permissão para criar um perfil vinculado ao serviço associado ao Network Monitor. Para saber mais sobre o perfil vinculado ao serviço, consulte Usar um perfil vinculado ao serviço para o CloudWatch Network Monitor.

Para criar um monitor para o Amazon CloudWatch Network Monitor no AWS Management Console, você deve ter feito login como um usuário ou perfil que tenha as permissões incluídas na política a seguir.

> **Note**
>
> Se você criar uma política de permissões baseada em identidade mais restritiva, os usuários com essa política não poderão criar um monitor.

```
{
```

```
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "networkmonitor:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
networkmonitor.amazonaws.com/AWSServiceRoleForNetworkMonitor",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "networkmonitor.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:GetRole",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
networkmonitor.amazonaws.com/AWSServiceRoleForNetworkMonitor"
        },
        {
            "Action": [
                "ec2:CreateSecurityGroup",
                "ec2:CreateNetworkInterface",
                "ec2:CreateTags"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Usar um perfil vinculado ao serviço para o CloudWatch Network Monitor

o Amazon CloudWatch Network Monitor usa os seguintes perfis vinculados ao serviço para as permissões necessárias para chamar outros serviços da AWS em seu nome:

- `AWSServiceRoleForNetworkMonitor`

`AWSServiceRoleForNetworkMonitor`

O CloudWatch Network Monitoring usa o perfil vinculado ao serviço denominado `AWSServiceRoleForNetworkMonitor` para atualizar e gerenciar os monitores de rede do CloudWatch.

A função vinculada ao serviço `AWSServiceRoleForNetworkMonitor` confia no seguinte serviço para assumir a função:

- `networkmonitor.amazonaws.com`

O `CloudWatchNetworkMonitorServiceRolePolicy` é anexado à função vinculada ao serviço e concede acesso ao serviço para acessar recursos de VPC e EC2 na sua conta, além de gerenciar os monitores de rede que foram criados.

Agrupamentos de permissões

A política é agrupada nos seguintes conjuntos de permissões:

- **`cloudwatch`**: permite que a entidade principal do serviço publique métricas de monitoramento de rede em recursos do CloudWatch.
- **`ec2`**: permite que a entidade principal do serviço descreva VPCs e sub-redes na sua conta para criar ou atualizar monitores e sondas. Também permite que a entidade principal do serviço crie, modifique e exclua grupos de segurança, interfaces de rede e suas permissões associadas para configurar o monitor ou a sonda para enviar tráfego de monitoramento aos endpoints.

Para obter mais informações sobre a política, consulte the section called “Políticas gerenciadas pela AWS”.

A seguir, é mostrado o `CloudWatchNetworkMonitorServiceRolePolicy`:

```
{
```

```
"Version": "2012-10-17",
"Statement": [
 {
  "Sid": "PublishCw",
  "Effect": "Allow",
  "Action": "cloudwatch:PutMetricData",
  "Resource": "*",
  "Condition": {
   "StringEquals": {
    "cloudwatch:namespace": "AWS/NetworkMonitor"
   }
  }
 },
 {
  "Sid": "DescribeAny",
  "Effect": "Allow",
  "Action": [
   "ec2:DescribeNetworkInterfaces",
   "ec2:DescribeNetworkInterfaceAttribute",
   "ec2:DescribeVpcs",
   "ec2:DescribeNetworkInterfacePermissions",
   "ec2:DescribeSubnets",
   "ec2:DescribeSecurityGroups"
  ],
  "Resource": "*"
 },
 {
  "Sid": "DeleteModifyEc2Resources",
  "Effect": "Allow",
  "Action": [
   "ec2:AuthorizeSecurityGroupEgress",
   "ec2:CreateNetworkInterfacePermission",
   "ec2:DeleteNetworkInterfacePermission",
   "ec2:RevokeSecurityGroupEgress",
   "ec2:ModifyNetworkInterfaceAttribute",
   "ec2:DeleteNetworkInterface",
   "ec2:DeleteSecurityGroup"
  ],
  "Resource": [
   "arn:aws:ec2:*:*:network-interface/*",
   "arn:aws:ec2:*:*:security-group/*"
  ],
  "Condition": {
   "StringEquals": {
```

```
      "aws:ResourceTag/ManagedByCloudWatchNetworkMonitor": "true"
    }
   }
  }
 ]
}
```

Criar a função vinculada ao serviço

`AWSServiceRoleForNetworkMonitor`

Você não precisa criar manualmente o perfil `AWSServiceRoleForNetworkMonitor`.

- O CloudWatch Network Monitor cria o perfil `AWSServiceRoleForNetworkMonitor` quando você cria seu primeiro monitor de rede. Esse perfil se aplicará a todos os monitores subsequentes que você criar.

Para criar um perfil vinculado ao serviço em seu nome, você deve ter as permissões necessárias. Para obter mais informações, consulte Permissões de função vinculada ao serviço no Guia do usuário do IAM.

Editar a função vinculada ao serviço

É possível editar a descrição de `AWSServiceRoleForNetworkMonitor` usando o IAM. Para obter mais informações, consulte Editar uma função vinculada ao serviço no Guia do usuário do IAM.

Excluir a função vinculada ao serviço

Se você não precisar mais usar o CloudWatch Network Monitor, recomendamos excluir o perfil `AWSServiceRoleForNetworkMonitor`.

Só é possível excluir esses perfis vinculados ao serviço depois de excluir o monitor de rede. Para obter informações sobre como excluir o monitor de rede, consulte Delete a network monitor.

Você pode usar o console, a CLI ou a API do IAM para excluir funções vinculadas ao serviço. Para obter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Depois de excluir `AWSServiceRoleForNetworkMonitor` , o CloudWatch Network Monitor criará novamente o perfil quando você criar um novo monitor.

Regiões compatíveis com o perfil vinculado ao serviço do CloudWatch Network Monitor

O CloudWatch Network Monitor é compatível com o perfil vinculado ao serviço em todas as Regiões da AWS em que o serviço está disponível. Para obter mais informações, consulte AWS endpoints na Referência geral da AWS.

Excluir a função vinculada ao serviço

Se você não precisar mais usar o CloudWatch Network Monitor, recomendamos excluir o perfil `AWSServiceRoleForNetworkMonitor`.

Só é possível excluir esses perfis vinculados ao serviço depois de excluir o monitor de rede. Para obter informações sobre como excluir o monitor de rede, consulte Delete a network monitor.

Você pode usar o console, a CLI ou a API do IAM para excluir funções vinculadas ao serviço. Para obter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Depois de excluir `AWSServiceRoleForNetworkMonitor` , o CloudWatch Network Monitor criará novamente o perfil quando você criar um novo monitor.

Definição de preço

Com o Amazon CloudWatch Network Monitor, não há custos iniciais nem compromissos de longo prazo. O preço do Network Monitor tem os dois seguintes componentes:

- uma taxa horária por recurso monitorado e
- taxas pelas métricas do CloudWatch.

Ao criar um monitor de rede, você associa recursos a ele que serão monitorados. Para o Network Monitor, isso consiste em sub-redes na Amazon Virtual Private Cloud (VPC). Cada recurso monitorado permite que você crie até quatro sondas de cada sub-rede nas VPCs para quatro destinos. Para ajudar a controlar sua fatura, você pode ajustar sua cobertura de sub-rede e a cobertura de IP on-premises ao reduzir o número dos recursos monitorados.

Para obter mais informações sobre a definição de preços, consulte a página Definição de preços do Amazon CloudWatch.

Monitoramento da infraestrutura

Os tópicos desta seção explicam os recursos do CloudWatch que podem ajudar você a obter visibilidade operacional dos seus recursos da AWS.

Tópicos

Container Insights

Use o CloudWatch Container Insights para coletar, agregar e resumir métricas e logs de suas aplicações e seus microsserviços conteinerizados. O Container Insights está disponível para Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) e plataformas do Kubernetes no Amazon EC2. O Container Insights é compatível com a coleta de métricas de clusters implantados no AWS Fargate para o Amazon ECS e Amazon EKS.

O CloudWatch coleta automaticamente métricas de muitos recursos, como CPU, memória, disco e rede. O Container Insights também fornece informações de diagnóstico, como falhas de reinicialização de contêiner, para ajudar a isolar problemas e resolvê-los rapidamente. Também é possível definir alarmes do CloudWatch em métricas que o Container Insights coleta.

O Container Insights coleta dados como eventos de log de performance usando formato de métrica incorporado. Esses eventos de log de performance são entradas que usam um esquema JSON estruturado que permite que dados de alta cardinalidade sejam ingeridos e armazenados em escala. Com base nesses dados, o CloudWatch cria métricas agregadas no nível de cluster, nó, pod, tarefa e serviço como métricas do CloudWatch. As métricas que o Container Insights coleta estão disponíveis nos painéis automáticos do CloudWatch e também podem ser visualizadas na seção Métricas do console do CloudWatch. As métricas não estarão visíveis até que as tarefas do contêiner estejam em execução por algum tempo.

Quando você implanta o Container Insights, ele cria automaticamente um grupo de logs para os eventos do log de performance. Você não precisa criar esse grupo de logs sozinho.

Para ajudar você a gerenciar os custos do Container Insights, o CloudWatch não cria automaticamente todas as métricas possíveis dos dados de logs. Porém, é possível visualizar outras métricas e outros níveis de detalhamento usando o CloudWatch Logs Insights para analisar os eventos de log de performance brutos.

Com a versão original do Container Insights, as métricas coletadas e os registros ingeridos são cobrados como métricas personalizadas. Com o Container Insights com observabilidade aprimorada para o Amazon EKS, as métricas e os logs do Container Insights são cobrados por observação em vez de serem cobrados por métrica armazenada ou log ingerido. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

No Amazon EKS e no Kubernetes, o Container Insights usa uma versão conteinerizada do atendente do CloudWatch para detectar todos os contêineres que estão em execução em um cluster. Depois, ele coleta dados de performance em cada camada da pilha de performance.

O Container Insights é compatível com a criptografia com o AWS KMS key dos registros e métricas que ele coleta. Para habilitar essa criptografia, você deve habilitar manualmente a criptografia do AWS KMS para o grupo de logs que recebe dados do Container Insights. Isso faz com que o Container Insights criptografe esses dados usando a chave do KMS fornecida. Somente chaves simétricas têm suporte. Não use chaves do KMS assimétricas para criptografar seus grupos de logs.

Para obter mais informações, consulte Criptografar dados de log no CloudWatch Logs usando o AWS KMS.

Container Insights com observabilidade aprimorada para o Amazon EKS

Em 6 de novembro de 2023, uma nova versão do Container Insights foi lançada. Essa versão é compatível com a observabilidade aprimorada dos clusters do Amazon EKS em execução no Amazon EC2 e pode coletar métricas mais detalhadas desses clusters. Após a instalação, ela coleta automaticamente a telemetria detalhada da infraestrutura e os registros de contêineres dos clusters do Amazon EKS. Em seguida, é possível usar painéis selecionados e imediatamente utilizáveis para detalhar a telemetria de aplicações e infraestrutura.

O Container Insights com observabilidade aprimorada para o Amazon EKS coleta métricas granulares de integridade, performance e status até o nível do contêiner, além de métricas do ambiente de gerenciamento. Para obter mais informações sobre as métricas e dimensões adicionais coletadas, consulte Métricas do Amazon EKS e do Kubernetes Container Insights.

Se você instalou o Container Insights usando o agente do CloudWatch em um cluster do Amazon EKS no Amazon EC2 após 6 de novembro de 2023, você tem o Container Insights com

observabilidade aprimorada para o Amazon EKS. Caso contrário, você pode atualizar um cluster do Amazon EKS para essa nova versão seguindo as instruções em Como atualizar para o Container Insights com observabilidade aprimorada para o Amazon EKS.

O Container Insights é compatível com a observabilidade entre contas do CloudWatch. Você usa uma única conta de monitoramento para monitorar e solucionar problemas relacionados a aplicações que abrangem diversas contas da AWS em uma única região. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

O Container Insights com observabilidade aprimorada para o Amazon EKS também é compatível com nós de processamento do Windows.

O Container Insights com observabilidade aprimorada para o Amazon EKS não é compatível com o Fargate.

> **Note**
>
> Você pode descobrir se tem clusters que podem ser atualizados para o Container Insights com observabilidade aprimorada para o Amazon EKS ao navegar até o console do Container Insights. Para fazer isso, selecione Insights, Container Insights no painel de navegação do console do CloudWatch. No console do Container Insights, um banner informa se você tem clusters do Amazon EKS que podem ser atualizados e links para a página de atualização.

Plataformas compatíveis

O Container Insights está disponível para o Amazon Elastic Container Service, o Amazon Elastic Kubernetes Service e as plataformas do Kubernetes em instâncias do Amazon EC2.

- Para o Amazon ECS, o Container Insights coleta métricas nos níveis de cluster, tarefa e serviço em instâncias do Linux e do Windows Server. Ele pode coletar métricas no nível de instância apenas em instâncias do Linux.

  No Amazon ECS, as métricas de rede estão disponíveis apenas para contêineres nos modos de rede `bridge` e `awsvpc`. Elas não estão disponíveis para contêineres no modo de rede `host`.
- No Amazon Elastic Kubernetes Service e nas plataformas do Kubernetes em instâncias do Amazon EC2, o Container Insights tem suporte apenas em instâncias do Linux.

Imagem de contêiner do atendente do CloudWatch

A Amazon fornece uma imagem de contêiner do atendente do CloudWatch no Amazon Elastic Container Registry. Para obter mais informações, consulte cloudwatch-agent no Amazon ECR.

Regiões compatíveis

O Container Insights para Amazon ECS tem suporte nas seguintes regiões:

- Leste dos EUA (Norte da Virgínia)
- Leste dos EUA (Ohio)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)
- África (Cidade do Cabo)
- Ásia-Pacífico (Hong Kong)
- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Osaka)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Tóquio)
- Ásia-Pacífico (Sydney)
- Oeste do Canadá (Calgary)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milão)
- Europe (Paris)
- Europa (Espanha)

- Europa (Estocolmo)
- Europa (Zurique)
- Oriente Médio (Barém)
- Oriente Médio (Emirados Árabes Unidos)
- South America (São Paulo)
- AWS GovCloud (Leste dos EUA)
- AWS GovCloud (Oeste dos EUA)
- China (Pequim)
- China (Ningxia)

Regiões compatíveis com Amazon EKS e Kubernetes

O Container Insights para o Amazon EKS e o Kubernetes tem suporte nas seguintes regiões:

- Leste dos EUA (Norte da Virgínia)
- Leste dos EUA (Ohio)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Hong Kong)
- Asia Pacific (Mumbai)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- China (Pequim)
- China (Ningxia)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)

- Europa (Estocolmo)
- Oriente Médio (Bahrein)
- South America (São Paulo)
- AWS GovCloud (Leste dos EUA)
- AWS GovCloud (Oeste dos EUA)

Configurar o Container Insights

O processo de configuração do Container Insights é diferente para o Amazon ECS e Amazon EKS e o Kubernetes.

Tópicos

- Configurar o Container Insights no Amazon ECS
- Configurar o Container Insights no Amazon EKS e no Kubernetes

Configurar o Container Insights no Amazon ECS

É possível usar uma ou ambas as opções a seguir para habilitar o Container Insights em clusters do Amazon ECS:

- Use o AWS Management Console ou a AWS CLI para começar a coletar métricas no nível de cluster, no nível de tarefa e no nível de serviço.
- Implante o agente do CloudWatch com um serviço daemon para começar a coletar métricas no nível de instância em clusters que são hospedados em instâncias do Amazon EC2.

Tópicos

- Configurar o Container Insights no Amazon ECS para métricas no nível de cluster e no nível de serviço
- Configurar o Container Insights no Amazon ECS usando o AWS Distro for OpenTelemetry
- Implantar o atendente do CloudWatch para coletar métricas no nível de instância do EC2 no Amazon ECS
- Implantar o AWS Distro for OpenTelemetry para coletar métricas no nível de instância do EC2 em clusters do Amazon ECS
- Configurar o FireLens para enviar logs ao CloudWatch Logs

Configurar o Container Insights no Amazon ECS para métricas no nível de cluster e no nível de serviço

É possível habilitar o Container Insights em clusters do Amazon ECS novos e existentes. O Container Insights coleta métricas nos níveis de cluster, de tarefa e de serviço. Você pode habilitar o Container Insights usando o console do Amazon ECS ou a AWS CLI.

Se estiver usando o Amazon ECS em uma instância do Amazon EC2 e quiser coletar métricas de rede e de armazenamento do Container Insights, execute essa instância usando uma AMI que inclua o atendente do Amazon ECS versão 1.29. Para obter informações sobre como atualizar a versão do atendente, consulte Atualizar o atendente de contêiner do Amazon ECS

É possível usar a AWS CLI para definir a permissão no nível de conta para habilitar o Container Insights para todos os novos clusters do Amazon ECS criados na conta. Para fazer isso, insira o comando a seguir.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

> **Note**
>
> Se a chave do AWS KMS gerenciada pelo cliente que você usa para as métricas do Amazon ECS Container Insights ainda não estiver configurada para funcionar com o CloudWatch, você deverá atualizar a política de chave para permitir registros criptografados no CloudWatch Logs. Você também deve associar sua própria chave do AWS KMS ao grupo de logs em `/aws/ecs/containerinsights/`*`ClusterName`*`/performance`. Para obter mais informações, consulte Criptografar dados de logs no CloudWatch Logs usando o AWS Key Management Service.

Configurar o Container Insights em clusters existentes do Amazon ECS

Para habilitar o Container Insights em um cluster existente do Amazon ECS, insira o comando a seguir. Você deve estar executando a versão 1.16.200 ou posterior da AWS CLI para que o comando a seguir funcione.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

Configurar o Container Insights em novos clusters do Amazon ECS

Há duas maneiras de habilitar o Container Insights em novos clusters do Amazon ECS. É possível configurar o Amazon ECS para que todos os novos clusters sejam habilitados para o Container Insights por padrão. Caso contrário, você poderá ativar um novo cluster ao criá-lo.

Usando a AWS Management Console

Por padrão, é possível ativar o Container Insights em todos os novos clusters ou em um cluster individual ao criá-lo.

Para ativar o Container Insights em todos os novos clusters por padrão

1. Abra o console em https://console.aws.amazon.com/ecs/v2.
2. Na página de navegação, selecione Account Settings (Configurações da conta).
3. Escolha Atualizar.
4. Para usar o CloudWatch Container Insights por padrão para clusters, em CloudWatch Container Insights, selecione ou desmarque CloudWatch Container Insights.
5. Escolha Salvar alterações.

Se você não tiver usado o procedimento anterior para habilitar o Container Insights em todos os novos clusters por padrão, poderá usar as etapas a seguir para criar um cluster com o Container Insights habilitado.

Para criar um cluster com o Container Insights ativado

1. Abra o console em https://console.aws.amazon.com/ecs/v2.
2. No painel de navegação, escolha Clusters.
3. Na página Clusters, escolha Create Cluster (Criar cluster).
4. Em Cluster configuration (Configuração do cluster), em Cluster name (Nome do cluster), insira um nome exclusivo.

   O nome pode conter até 255 letras (minúsculas e maiúsculas), números e hifens.
5. Para ativar o Container Insights, expanda Monitoramento e, em seguida, ative Usar o Conteiner Insights.

Agora você pode criar definições de tarefa, executar tarefas e iniciar serviços no cluster. Para mais informações, consulte:

- Criar uma definição de tarefa
- Tarefas em execução
- Criar um serviço

Configurar o Container Insights em novos clusters do Amazon ECS usando a AWS CLI

Para ativar o Container Insights em todos os clusters novos por padrão, insira o comando a seguir.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

Se você não usou o comando anterior para habilitar o Container Insights em todos os novos clusters por padrão, insira o comando a seguir para criar um novo cluster com o Container Insights habilitado. Você deve estar executando a versão 1.16.200 ou posterior da AWS CLI para que o comando a seguir funcione.

```
aws ecs create-cluster --cluster-name myCICluster --settings
 "name=containerInsights,value=enabled"
```

Desabilitar o Container Insights em clusters do Amazon ECS

Para desabilitar o Container Insights em um cluster existente do Amazon ECS, insira o comando a seguir.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=disabled
```

Configurar o Container Insights no Amazon ECS usando o AWS Distro for OpenTelemetry

Consulte esta seção para usar o AWS Distro for OpenTelemetry para configurar o CloudWatch Container Insights em um cluster do Amazon ECS. Para obter mais informações sobre o AWS Distro for OpenTelemetry, consulte AWS Distro for OpenTelemetry.

Estas etapas presumem que você já tenha um cluster executando o Amazon ECS. Para obter mais informações sobre como usar o AWS Distro for Open Telemetry com o Amazon ECS e configurar um cluster do Amazon ECS para essa finalidade, consulte Configurar o AWS Distro for Open Telemetry Collector no Amazon Elastic Container Service.

Etapa 1: Criar uma função de tarefa

A primeira etapa é criar uma função de tarefa no cluster que o AWS Distro for Open Telemetry Collector usará.

Para criar uma função de tarefa para o AWS Distro for Open Telemetry

1. Abra o console IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Policies (Políticas) e, em seguida, selecione Create policy (Criar política).
3. Selecione a guia JSON e copie a política a seguir:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Escolha Revisar política.
5. Para nome, insira **`AWSDistroOpenTelemetryPolicy`** e escolha Create policy (Criar política).
6. No painel de navegação à esquerda, escolha Roles (Funções) e Create role (Criar função).
7. Na lista de serviços, escolha Elastic Container Service.
8. Na parte inferior da página, escolha Elastic Container Service e Next: Permissions (Próximo: permissões).
9. Na lista de políticas, procure AWSDistroOpenTelemetryPolicy.
10. Marque a caixa de seleção ao lado de AWSDistroOpenTelemetryPolicy.
11. Escolha Next: Tags (Próximo: etiquetas) e Next: Review (Próximo: revisar).

12. Em Role name (Nome da função), insira **`AWSOpenTelemetryTaskRole`** e escolha Create role (Criar função).

Etapa 2: Criar uma função de execução de tarefa

A próxima etapa é criar uma função de execução de tarefas para o AWS OpenTelemetry Collector.

Para criar uma função de execução de tarefa para o AWS Distro for Open Telemetry

1. Abra o console IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Roles (Funções) e Create role (Criar função).
3. Na lista de serviços, escolha Elastic Container Service.
4. Na parte inferior da página, escolha Elastic Container Service e Next: Permissions (Próximo: permissões).
5. Na lista de políticas, procure AmazonECSTaskExecutionRolePolicy e marque a caixa de seleção ao lado de AmazonECSTaskExecutionRolePolicy.
6. Na lista de políticas, procure CloudWatchLogsFullAccess e marque a caixa de seleção ao lado de CloudWatchLogsFullAccess.
7. Na lista de políticas, procure AmazonSSMReadOnlyAccess e marque a caixa de seleção ao lado de AmazonSSMReadOnlyAccess.
8. Escolha Next: Tags (Próximo: etiquetas) e Next: Review (Próximo: revisar).
9. Em Role name (Nome da função), insira **`AWSOpenTelemetryTaskExecutionRole`** e escolha Create role (Criar função).

Etapa 3: Criar uma definição de tarefa

A próxima etapa é criar uma definição de tarefa.

Para criar uma definição de tarefa para o AWS Distro for Open Telemetry

1. Abra o console em https://console.aws.amazon.com/ecs/v2.
2. No painel de navegação, escolha Task definitions (Definições de tarefas)
3. Escolha Create new task definition (Criar nova definição de tarefa), Create new task definition (Criar nova definição de tarefa).
4. Em Task definition family (Família de definição de tarefa), especifique um nome exclusivo para a definição de tarefa.

5. Configure seus contêineres e escolha Avançar.
6. Em Métricas e logs, selecione Usar coleção de métricas.
7. Escolha Próximo.
8. Escolha Criar.

Para obter mais informações sobre como usar o AWS Open Telemetry Collector com o Amazon ECS, consulte Configurar o AWS Distro for Open Telemetry Collector no Amazon Elastic Container Service.

Etapa 4: Executar a tarefa

A etapa final é executar a tarefa que você criou.

Para executar a tarefa para AWS Distro for OpenTelemetry

1. Abra o console em https://console.aws.amazon.com/ecs/v2.
2. No painel de navegação à esquerda, escolha Task Definitions (Definições de tarefa) e selecione a tarefa que você acabou de criar.
3. Escolha Ações, Implantar, Executar tarefa.
4. Escolha Deploy (Implantar), Run task (Executar tarefa).
5. Na seção Opções de computação, em Cluster existente, escolha o cluster.
6. Escolha Criar.
7. Em seguida, é possível conferir as novas métricas no console do CloudWatch.
8. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
9. No painel de navegação à esquerda, escolha Metrics.

   Você deverá ver um namespace ECS/ContainerInsights. Escolha esse namespace e verá oito métricas.

Implantar o atendente do CloudWatch para coletar métricas no nível de instância do EC2 no Amazon ECS

Para implantar o atendente do CloudWatch para coletar métricas no nível de instância de clusters do Amazon ECS hospedados em uma instância do EC2, use uma configuração de início rápido com uma configuração padrão ou instale o atendente manualmente para poder personalizá-lo.

Ambos os métodos exigem que você já tenha, pelo menos, um cluster do Amazon ECS implantado com um tipo de inicialização do EC2 e que o contêiner do agente do CloudWatch tenha acesso ao Instance Metadata Service (IMDS) do EC2. Para obter mais informações sobre o IMDS, consulte Metadados da instância e dados de usuário.

Esses métodos também presumem que a AWS CLI esteja instalada. Além disso, para executar os comandos nos procedimentos a seguir, é necessário estar conectado em uma conta ou em uma função que tenha as políticas IAMFullAccess e AmazonECS_FullAccess.

Tópicos

- Configuração rápida usando o AWS CloudFormation
- Configuração manual e personalizada

Configuração rápida usando o AWS CloudFormation

Para usar a configuração rápida, insira o comando a seguir a fim de usar o AWS CloudFormation para instalar o atendente. Substitua *cluster-name* e *cluster-region* pelo nome e pela região de seu cluster do Amazon ECS.

Esse comando cria as funções do IAM CWAgentECSTaskRole e CWAgentECSExecutionRole. Se essas funções já existirem em sua conta, use `ParameterKey=CreateIAMRoles,ParameterValue=False` em vez de `ParameterKey=CreateIAMRoles,ParameterValue=True` ao inserir o comando. Caso contrário, o comando falhará.

```
ClusterName=cluster-name
Region=cluster-region
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body file://cwagent-ecs-instance-metric-cfn.json \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

(Alternativa) Usar suas próprias funções do IAM

Se quiser usar suas próprias função de tarefa do ECS e função de execução de tarefa do ECS em vez das funções CWAgentECSTaskRole e CWAgentECSExecutionRole, primeiro verifique se a função que deve ser usada como a função de tarefa do ECS tem a CloudWatchAgentServerPolicy anexada. Além disso, verifique se a função a ser usada como função de execução de tarefa do ECS tem as políticas CloudWatchAgentServerPolicy e AmazonECSTaskExecutionRolePolicy anexadas. Depois, insira o comando a seguir. No comando, substitua *task-role-arn* pelo ARN da função de tarefa do ECS e substitua *execution-role-arn* pelo ARN da função de execução de tarefa do ECS personalizada.

```
ClusterName=cluster-name
Region=cluster-region
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body file://cwagent-ecs-instance-metric-cfn.json \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=TaskRoleArn,ParameterValue=${TaskRoleArn} \
                 ParameterKey=ExecutionRoleArn,ParameterValue=${ExecutionRoleArn} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

Solucionar problemas da configuração rápida

Para verificar o status da pilha do AWS CloudFormation, insira o comando a seguir.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stacks --stack-name CWAgentECS-$ClusterName-$Region --
region $Region
```

Se o `StackStatus` for diferente de `CREATE_COMPLETE` ou de `CREATE_IN_PROGRESS`, verifique os eventos da pilha para localizar o erro. Insira o comando da a seguir.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name CWAgentECS-$ClusterName-$Region
 --region $Region
```

Para conferir o status do serviço daemon `cwagent`, insira o comando a seguir. Na saída, `runningCount` deve ser igual a `desiredCount` na seção `deployment`. Se não for igual, verifique a seção `failures` na saída.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services cwagent-daemon-service --cluster $ClusterName --
region $Region
```

Também é possível usar o console do CloudWatch Logs para conferir o log do atendente. Procure o grupo de logs /ecs/ecs-cwagent-daemon-service.

Excluir a pilha do AWS CloudFormation do atendente do CloudWatch

Se for necessário excluir a pilha do AWS CloudFormation, insira o comando a seguir.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation delete-stack --stack-name CWAgentECS-${ClusterName}-${Region} --
region ${Region}
```

Configuração manual e personalizada

Siga as etapas desta seção para implantar manualmente o atendente do CloudWatch a fim de coletar métricas no nível de instância dos clusters do Amazon ECS hospedados em instâncias do EC2.

Funções e políticas do IAM necessárias

São necessárias duas funções do IAM. É necessário criá-las, caso ainda não existam. Para obter mais informações sobre essas funções, consulte Funções do IAM para tarefas e Função de execução de tarefa do Amazon ECS.

- Uma função de tarefa do ECS, que é usada pelo atendente do CloudWatch para publicar métricas. Se essa função já existir, será necessário verificar se ela tem a política `CloudWatchAgentServerPolicy` anexada.
- Uma função de execução de tarefa do ECS, que é usada pelo atendente do Amazon ECS para executar o atendente do CloudWatch. Se essa função já existir, será necessário verificar se ela tem as políticas `AmazonECSTaskExecutionRolePolicy` e `CloudWatchAgentServerPolicy` anexadas.

Se ainda não tiver essas funções, você poderá usar os comandos a seguir para criá-las e anexar as políticas necessárias. Este primeiro comando cria a função de tarefa do ECS.

```
aws iam create-role --role-name CWAgentECSTaskRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid
\": \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com
\"},\"Action\": \"sts:AssumeRole\"}]}"
```

Após inserir o comando anterior, anote o valor de `Arn` da saída do comando como "TaskRoleArn". Você precisará usá-lo posteriormente ao criar a definição de tarefa. Depois, insira o comando a seguir para anexar as políticas necessárias.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
CloudWatchAgentServerPolicy \
    --role-name CWAgentECSTaskRole
```

Este próximo comando cria a função de execução de tarefa do ECS.

```
aws iam create-role --role-name CWAgentECSExecutionRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid
\": \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com
\"},\"Action\": \"sts:AssumeRole\"}]}"
```

Após inserir o comando anterior, anote o valor de `Arn` da saída do comando como "ExecutionRoleArn". Você precisará usá-lo posteriormente ao criar a definição de tarefa. Depois, insira os comandos a seguir para anexar as políticas necessárias.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
CloudWatchAgentServerPolicy \
    --role-name CWAgentECSExecutionRole

aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy \
    --role-name CWAgentECSExecutionRole
```

Criar a definição de tarefa e iniciar o serviço daemon

Crie uma definição de tarefa e use-a para executar o atendente do CloudWatch como um serviço daemon. Para criar a definição de tarefa, insira o comando a seguir. Nas primeiras linhas, substitua os espaços reservados pelos valores reais da implantação. *`logs-region`* é a região onde o

CloudWatch Logs está localizado, e *cluster-region* é a região onde o cluster está localizado. *task-role-arn* é o ARN da função de execução de tarefa do ECS que você está usando, e *execution-role-arn* é o ARN da função de execução de tarefa do ECS.

```
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
AWSLogsRegion=logs-region
Region=cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cwagent-ecs-instance-metric.json \
    | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
    | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-json
```

Depois, execute o comando a seguir para executar o serviço daemon. Substitua *cluster-name* e *cluster-region* pelo nome e pela região de seu cluster do Amazon ECS.

> **Important**
>
> Remova todas as estratégias do provedor de capacidade antes de executar o comando. Caso contrário, o comando não funcionará.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs create-service \
    --cluster ${ClusterName} \
    --service-name cwagent-daemon-service \
    --task-definition ecs-cwagent-daemon-service \
    --scheduling-strategy DAEMON \
    --region ${Region}
```

Se esta mensagem de erro for exibida, `An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent`, é porque você já criou um serviço daemon chamado `cwagent-daemon-service`. Será necessário excluir esse serviço primeiro, usando o comando a seguir como exemplo.

```
ClusterName=cluster-name
Region=cluster-region
```

```
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
    --force
```

(Opcional) Configuração avançada

Se preferir, você poderá usar o SSM para especificar outras opções de configuração para o atendente do CloudWatch nos clusters do Amazon ECS hospedados nas instâncias do EC2. Essas opções são as seguintes:

- `metrics_collection_interval`? a frequência, em segundos, com a qual o atendente do CloudWatch coleta as métricas. O padrão é 60. O intervalo é de 1 a 172.000.
- `endpoint_override`: (opcional) especifica um endpoint diferente para o qual enviar logs. Você pode querer fazer isso se estiver publicando de um cluster em uma VPC e quiser que os dados de logs vão para um VPC endpoint.

  O valor de `endpoint_override` deve ser uma string que seja um URL.
- `force_flush_interval`: especifica em segundos a quantidade máxima de tempo em que os logs permanecem no buffer da memória antes de serem enviados ao servidor. Não importa a configuração para esse campo, se o tamanho dos logs no buffer alcançar 1 MB, os logs serão enviados imediatamente para o servidor. O valor de padrão é de 5 segundos.
- `region`: por padrão, o atendente publica métricas na mesma região onde a instância de contêiner do Amazon ECS está localizada. Para substituir isso, é possível especificar uma região diferente aqui. Por exemplo, `"region" : "us-east-1"`

Veja a seguir um exemplo de uma configuração personalizada:

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "ecs": {
                "metrics_collection_interval": 30
            }
        },
```

```
        "force_flush_interval": 5
    }
}
```

Para personalizar a configuração do atendente do CloudWatch nos contêineres do Amazon ECS

1. Verifique se a política AmazonSSMReadOnlyAccess está anexada à função de execução de tarefa do Amazon ECS. É possível inserir o comando a seguir para fazer isso. Esse exemplo pressupõe que a função de execução de tarefa do Amazon ECS seja CWAgentECSExecutionRole. Se você estiver usando uma função diferente, substitua esse nome da função no comando a seguir.

   ```
   aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
   AmazonSSMReadOnlyAccess \
           --role-name CWAgentECSExecutionRole
   ```

2. Crie o arquivo de configuração personalizada semelhante ao exemplo anterior. Nomeie esse arquivo como `/tmp/ecs-cwagent-daemon-config.json`.

3. Execute o comando a seguir para colocar essa configuração no Parameter Store. Substitua *`cluster-region`* pela região do cluster do Amazon ECS. Para executar esse comando, é necessário estar conectado a um usuário ou a uma função que tenha a política AmazonSSMFullAccess.

   ```
   Region=cluster-region
   aws ssm put-parameter \
       --name "ecs-cwagent-daemon-service" \
       --type "String" \
       --value "`cat /tmp/ecs-cwagent-daemon-config.json`" \
       --region $Region
   ```

4. Faça download do arquivo de definição de tarefa em um arquivo local, como `/tmp/cwagent-ecs-instance-metric.json`.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/
   cwagent-ecs-instance-metric/cwagent-ecs-instance-metric.json -o /tmp/cwagent-ecs-
   instance-metric.json
   ```

5. Modifique o arquivo de definição de tarefa. Remova a seguinte seção:

```
"environment": [
                {
                    "name": "USE_DEFAULT_CONFIG",
                    "value": "True"
                }
            ],
```

Substitua essa seção pela seguinte:

```
"secrets": [
                {
                    "name": "CW_CONFIG_CONTENT",
                    "valueFrom": "ecs-cwagent-daemon-service"
                }
            ],
```

6. Reinicie o atendente como um serviço daemon seguindo estas etapas:

   a. Execute o seguinte comando .

   ```
   TaskRoleArn=task-role-arn
   ExecutionRoleArn=execution-role-arn
   AWSLogsRegion=logs-region
   Region=cluster-region
   cat /tmp/cwagent-ecs-instance-metric.json \
       | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
   ${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
       | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-
   json
   ```

   b. Execute o comando a seguir para executar o serviço daemon. Substitua *cluster-name* e *cluster-region* pelo nome e pela região de seu cluster do Amazon ECS.

   ```
   ClusterName=cluster-name
   Region=cluster-region
   aws ecs create-service \
       --cluster ${ClusterName} \
       --service-name cwagent-daemon-service \
       --task-definition ecs-cwagent-daemon-service \
       --scheduling-strategy DAEMON \
       --region ${Region}
   ```

Se esta mensagem de erro for exibida, `An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent`, é porque você já criou um serviço daemon chamado `cwagent-daemon-service`. Será necessário excluir esse serviço primeiro, usando o comando a seguir como exemplo.

```
ClusterName=cluster-name
Region=Region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
    --force
```

Implantar o AWS Distro for OpenTelemetry para coletar métricas no nível de instância do EC2 em clusters do Amazon ECS

Realize as etapas desta seção para usar o AWS Distro for OpenTelemetry para coletar métricas no nível de instância do EC2 em clusters do Amazon ECS. Para obter mais informações sobre o AWS Distro for OpenTelemetry, consulte AWS Distro for OpenTelemetry.

Estas etapas presumem que você já tenha um cluster executando o Amazon ECS. Esse cluster deve ser implantado com o tipo de inicialização EC2. Para obter mais informações sobre como usar o AWS Distro for Open Telemetry com o Amazon ECS e configurar um cluster do Amazon ECS para essa finalidade, consulte Configurar o AWS Distro for Open Telemetry Collector no Amazon Elastic Container Service para métricas no nível da instância do EC2 no ECS.

Tópicos

- Configuração rápida usando o AWS CloudFormation
- Configuração manual e personalizada

Configuração rápida usando o AWS CloudFormation

Baixe o arquivo de modelo do AWS CloudFormation para instalar o AWS Distro for OpenTelemetry Colector para o Amazon ECS no EC2. Execute o comando curl a seguir.

```
curl -O https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
deployment-template/ecs/aws-otel-ec2-instance-metrics-daemon-deployment-cfn.yaml
```

Depois de baixar o arquivo de modelo, abra-o e substitua *PATH_TO_CloudFormation_TEMPLATE* pelo caminho onde você salvou o arquivo de modelo. Em seguida, exporte os seguintes parâmetros e execute o comando AWS CloudFormation, conforme mostrado no comando a seguir.

- Cluster_Name: o nome do cluster do Amazon ECS
- AWS_Region: a região para onde os dados serão enviados
- PATH_TO_CloudFormation_TEMPLATE: o caminho onde você salvou o arquivo de modelo do AWS CloudFormation.
- command: para habilitar o AWS Distro for OpenTelemetry Collector para coletar as métricas no nível de instância do Amazon ECS no Amazon EC2, é necessário especificar `--config=/etc/ecs/otel-instance-metrics-config.yaml` para este parâmetro.

```
ClusterName=Cluster_Name
Region=AWS_Region
command=--config=/etc/ecs/otel-instance-metrics-config.yaml
aws cloudformation create-stack --stack-name AOCECS-${ClusterName}-${Region} \
--template-body file://PATH_TO_CloudFormation_TEMPLATE \
--parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
ParameterKey=CreateIAMRoles,ParameterValue=True \
ParameterKey=command,ParameterValue=${command} \
--capabilities CAPABILITY_NAMED_IAM \
--region ${Region}
```

Depois de executar este comando, use o console do Amazon ECS para ver se a tarefa está em execução.

Solucionar problemas da configuração rápida

Para verificar o status da pilha do AWS CloudFormation, insira o comando a seguir.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack --stack-name AOCECS-$ClusterName-$Region --region
 $Region
```

Se o valor de `StackStatus` for diferente de `CREATE_COMPLETE` ou de `CREATE_IN_PROGRESS`, verifique os eventos da pilha para localizar o erro. Insira o comando da a seguir.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name AOCECS-$ClusterName-$Region --
region $Region
```

Para conferir o status do serviço daemon `AOCECS`, insira o comando a seguir. Na saída, é necessário verificar se `runningCount` é igual a `desiredCount` na seção de implantação. Se não for igual, confira a seção de falhas na saída.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services AOCECS-daemon-service --cluster $ClusterName --
region $Region
```

Também é possível usar o console do CloudWatch Logs para conferir o log do atendente. Procure o grupo de logs /aws/ecs/containerinsights/{ClusterName}/performance.

Configuração manual e personalizada

Siga as etapas desta seção para implantar manualmente o AWS Distro for OpenTelemetry a fim de coletar métricas no nível de instância dos clusters do Amazon ECS hospedados em instâncias do Amazon EC2.

Etapa 1: Funções e políticas do IAM necessárias

São necessárias duas funções do IAM. É necessário criá-las, caso ainda não existam. Para obter mais informações sobre essas funções, consulte Criar política do IAM e Criar função do IAM.

Etapa 2: Criar uma definição de tarefa

Crie uma definição de tarefa e use-a para iniciar o atendente do AWS Distro for OpenTelemetry como um serviço daemon.

Para usar o modelo de definição de tarefa para criar a definição de tarefa, siga as instruções em Criar definição de tarefa do ECS EC2 para instância do EC2 com AWS OTel Collector.

Para usar o console do Amazon ECS para criar a definição de tarefa, siga as instruções em Instalar AWS OTel Collector criando definição de rarefas pelo Console AWS para métricas de instância do EC2 no Amazon ECS.

Etapa 3: Iniciar o serviço daemon

Para iniciar o AWS Distro para OpenTeemetry como um serviço daemon, siga as instruções em Executar tarefa no Amazon Elastic Container Service (Amazon ECS) usando o serviço daemon.

(Opcional) Configuração avançada

Se preferir, você poderá usar o SSM para especificar outras opções de configuração para o AWS Distro for OpenTelemetry nos clusters do Amazon ECS hospedados nas instâncias do Amazon EC2. Para obter mais informações sobre como criar um arquivo de configuração, consulte Configuração personalizada do OpenTelemetry. Para obter mais informações sobre as opções que você pode usar no arquivo de configuração, consulte Receptor do AWS Container Insights.

Configurar o FireLens para enviar logs ao CloudWatch Logs

O FireLens para Amazon ECS permite usar parâmetros de definição de tarefa para rotear logs para o Amazon CloudWatch Logs para armazenamento e análise de logs. O FireLens funciona com Fluent Bit e Fluentd. Fornecemos a imagem da AWS for Fluent Bit ou é possível usar sua própria imagem do Fluent Bit ou Fluentd. Criar definições de tarefa do Amazon ECS com uma configuração do FireLens tem suporte usando os AWS SDKs, AWS CLI e AWS Management Console. Para obter mais informações sobre o CloudWatch Logs, consulte O que é o Amazon CloudWatch Logs?.

Há considerações importantes ao usar o FireLens for Amazon ECS. Para obter mais informações, consulte Considerações.

Para encontrar imagens da AWS for Fluent Bit, consulte Usar a imagem da AWS for Fluent Bit.

Para criar uma definição de tarefa que usa uma configuração do FireLens, consulte Como criar uma definição de tarefa que usa uma configuração do FireLens.

Exemplo

O exemplo de definição de tarefa a seguir mostra como especificar uma configuração de log que encaminha logs a um grupo de logs do CloudWatch Logs. Para obter mais informações, consulte O que é o Amazon CloudWatch Logs? no Guia do usuário do Amazon CloudWatch Logs.

Nas opções de configuração de log, especifique o nome do grupo de logs e a região em que ele existe. Para que o Fluent Bit crie o grupo de logs em seu nome, especifique `"auto_create_group":"true"`. Também é possível especificar o ID da tarefa como o prefixo de fluxo de log que auxilia na filtragem. Para obter mais informações, consulte Plugin do Fluent Bit para CloudWatch Logs.

```
{
 "family": "firelens-example-cloudwatch",
 "taskRoleArn": "arn:aws:iam::123456789012:role/ecs_task_iam_role",
 "containerDefinitions": [
  {
   "essential": true,
   "image": "906394416424.dkr.ecr.us-west-2.amazonaws.com/aws-for-fluent-bit:latest",
   "name": "log_router",
   "firelensConfiguration": {
    "type": "fluentbit"
   },
   "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
     "awslogs-group": "firelens-container",
     "awslogs-region": "us-west-2",
     "awslogs-create-group": "true",
     "awslogs-stream-prefix": "firelens"
    }
   },
   "memoryReservation": 50
   },
   {
    "essential": true,
    "image": "nginx",
    "name": "app",
    "logConfiguration": {
     "logDriver":"awsfirelens",
     "options": {
     "Name": "cloudwatch",
     "region": "us-west-2",
     "log_key": "log",
                                  "log_group_name": "/aws/ecs/containerinsights/
$(ecs_cluster)/application",
     "auto_create_group": "true",
     "log_stream_name": "$(ecs_task_id)"
    }
   },
   "memoryReservation": 100
  }
 ]
}
```

Configurar o Container Insights no Amazon EKS e no Kubernetes

Há suporte para o Container Insights nas versões 1.23 e posteriores do Amazon EKS. Há suporte para o método de início rápido da instalação somente nas versões 1.24 e posteriores.

O processo geral para configurar o Container Insights no Amazon EKS ou no Kubernetes é o seguinte:

1. Verifique se você tem os pré-requisitos necessários.
2. Configure o complemento de observabilidade do EKS do Amazon CloudWatch, o agente do CloudWatch ou o AWS Distro para OpenTelemetry em seu cluster para enviar métricas ao CloudWatch.

> **Note**
>
> Para usar o Container Insights com observabilidade aprimorada para o Amazon EKS, você deve usar o complemento de observabilidade do EKS do Amazon CloudWatch ou o agente do CloudWatch. Para obter mais informações sobre esta versão do Container Insights, consulte Container Insights com observabilidade aprimorada para o Amazon EKS.
> Para usar o Container Insights com o Fargate, você deve usar o AWS Distro para OpenTelemetry. O Container Insights com observabilidade aprimorada para o Amazon EKS não é compatível com o Fargate.

> **Note**
>
> Agora, o Container Insights é compatível com nós de processamento do Windows em um cluster do Amazon EKS. Além disso, o Container Insights com observabilidade aprimorada para o Amazon EKS é compatível com o sistema Windows. Para obter informações sobre como habilitar o Container Insights no Windows, consulte Como usar o agente do CloudWatch com observabilidade aprimorada do Container Insights ativada.

Configure o Fluent Bit ou o Fluentd para enviar logs ao CloudWatch Logs. (Isso ficará ativado por padrão se você instalar o complemento de observabilidade do EKS do Amazon CloudWatch.)

Você pode executar essas etapas de uma só vez como parte da configuração de início rápido, se estiver usando o atendente do CloudWatch, ou executá-las de forma separada.

3. (Opcional) Configure o registro do ambiente de gerenciamento do Amazon EKS.
4. (Opcional) Configure o atendente do CloudWatch como um endpoint do StatsD no cluster para enviar métricas do StatsD ao CloudWatch.
5. (Opcional) Habilite logs de acesso do App Mesh Envoy.

Com a versão original do Container Insights, as métricas coletadas e os registros ingeridos são cobrados como métricas personalizadas. Com o Container Insights com observabilidade aprimorada para o Amazon EKS, as métricas e os logs do Container Insights são cobrados por observação em vez de serem cobrados por métrica armazenada ou log ingerido. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

Tópicos

- Verifique os pré-requisitos do
- Como usar o agente do CloudWatch com observabilidade aprimorada do Container Insights ativada
- Usar o AWS Distro for OpenTelemetry
- Enviar logs ao CloudWatch Logs
- Atualizar ou excluir o Container Insights no Amazon EKS e no Kubernetes

Verifique os pré-requisitos do

Antes de instalar o Container Insights no Amazon EKS ou no Kubernetes, verifique os pré-requisitos a seguir. Esses pré-requisitos se aplicam tanto se você usar o atendente do CloudWatch ou o AWS Distro for OpenTelemetry para configurar o Container Insights em clusters do Amazon EKS.

- Você tem um cluster funcional do Amazon EKS ou do Kubernetes com nós anexados em uma das regiões que oferecem suporte ao Container Insights ao Amazon EKS e ao Kubernetes. Para obter a lista de regiões compatíveis, consulte Container Insights.
- Você tem `kubectl` instalado e em execução. Para obter mais informações, consulte Instalar o `kubectl` no Manual do usuário do Amazon EKS.
- Se você estiver usando o Kubernetes em execução na AWS, em vez de usar o Amazon EKS, os seguintes pré-requisitos também serão necessários:

- Certifique-se de que o cluster do Kubernetes habilitou o controle de acesso baseado em funções (RBAC). Para obter mais informações, consulte Usar a autorização de RBAC (em inglês) na Referência do Kubernetes.
- Seu kubelet habilitou o modo de autorização Webhook. Para obter mais informações, consulte Autenticação/autorização do Kubelet (em inglês) na Referência do Kubernetes.

Você também deve conceder permissões do IAM para permitir que seus nós de processamento do Amazon EKS enviem métricas e logs ao CloudWatch. Há duas maneiras de fazer isso:

- Anexe uma política à função do IAM dos nós de processamento. Isso funciona tanto para clusters do Amazon EKS quanto para outros clusters do Kubernetes.
- Utilize uma função do IAM para contas de serviço para o cluster e anexe a política a essa função. Funciona somente para clusters do Amazon EKS.

A primeira opção concede permissões ao CloudWatch para o nó inteiro, enquanto o uso de uma função do IAM para a conta de serviço dá acesso ao CloudWatch somente aos pods do daemonset apropriados.

Anexar uma política à função do IAM de seus nós de processamento

Siga estas etapas para anexar a política à função do IAM dos nós de processamento. Isso funciona tanto para clusters do Amazon EKS como para clusters do Kubernetes fora do Amazon EKS.

Como adicionar a política necessária à função do IAM para os nós de processamento

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. Selecione uma das instâncias do nó de processamento e escolha a função do IAM na descrição.
3. Na página da função do IAM, selecione Attach policies (Anexar políticas).
4. Na lista de políticas, marque a caixa de seleção ao lado de CloudWatchAgentServerPolicy. Se necessário, use a caixa de pesquisa para encontrar essa política.
5. Escolha Anexar políticas.

Se você estiver executando um cluster do Kubernetes fora do Amazon EKS, talvez você não tenha uma função do IAM anexada a seus nós de processamento. Caso contrário, primeiro anexe uma função do IAM à instância e adicione a política conforme explicado nas etapas anteriores. Para obter

mais informações sobre como anexar uma função do IAM a uma instância, consulte Anexar uma função do IAM a uma instância, no Manual do usuário do Amazon EC2 para instâncias do Windows.

Se estiver executando um cluster do Kubernetes fora do Amazon EKS e quiser coletar IDs de volumes do EBS nas métricas. você deverá adicionar outra política à função do IAM anexada à instância. Adicione o seguinte como uma política em linha. Para obter mais informações, consulte Adicionar e remover permissões de identidade do IAM no Manual do usuário do IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Usar uma função de conta de serviço do IAM

Esse método funciona somente em clusters do Amazon EKS.

Para conceder permissão para o CloudWatch usar uma função de conta de serviço do IAM

1. Caso ainda não tenha feito isso, habilite as funções do IAM para contas de serviço no cluster. Para obter mais informações, consulte Habilitar funções do IAM para contas de serviço em seu cluster .
2. Caso ainda não tenha configurado, configure a conta de serviço para usar o perfil do IAM. Para obter mais informações, consulte Configuração de uma conta de serviço do Kubernetes para assumir um perfil do IAM.

   Ao criar a função, anexe a política do IAM CloudWatchAgentServerPolicy à função, além da política que você criar para a função. Além disso, a conta de serviço do Kubernetes associada que está vinculada a essa função deve ser criada no namespace do `amazon-cloudwatch`, no qual os daemonsets do CloudWatch e do Fluent Bit serão implantados nas próximas etapas.

3. Associe a função do IAM a uma conta de serviço no cluster, se ainda não tiver feito isso. Para obter mais informações, consulte Configuração de uma conta de serviço do Kubernetes para assumir um perfil do IAM.

Como usar o agente do CloudWatch com observabilidade aprimorada do Container Insights ativada

Use as instruções em uma das seções a seguir para configurar o Container Insights em um cluster do Amazon EKS ou do Kubernetes usando o agente do CloudWatch. Há suporte para as instruções de início rápido somente nas versões 1.24 e posteriores do Amazon EKS.

> **Note**
>
> Você pode instalar o Container Insights seguindo as instruções em qualquer uma das seções a seguir. Não é necessário seguir todos os três conjuntos de instruções.

Tópicos

- Instalar o complemento Amazon CloudWatch Observability do EKS
- Configuração de início rápido para o Container Insights no Amazon EKS e no Kubernetes
- Configurar o atendente do CloudWatch para coletar métricas do cluster

Instalar o complemento Amazon CloudWatch Observability do EKS

Você pode usar o complemento do EKS da Amazon para instalar o Container Insights com observabilidade aprimorada para o Amazon EKS. O complemento instala o agente do CloudWatch para enviar as métricas de infraestrutura do cluster, instala o Fluent Bit para enviar os logs de contêiner, e também habilita que o CloudWatch Application Signals envie a telemetria de performance para as aplicações.

Quando você usa o complemento do Amazon EKS na versão 1.5.0 ou em versões posteriores, o Container Insights é habilitado nos nós de processamento do Linux e do Windows no cluster. No momento, o Application Signals não é compatível com sistema Windows no Amazon EKS.

O complemento do Amazon EKS não é compatível com clusters que executam o Kubernetes em vez do Amazon EKS.

Para obter mais informações sobre o complemento de observabilidade do EKS do Amazon CloudWatch, consulte Instalar o agente do CloudWatch usando o complemento de observabilidade do EKS do Amazon CloudWatch.

Como instalar o complemento Amazon CloudWatch Observability do EKS

1. Primeiro, configure as permissões necessárias ao anexar a política do IAM CloudWatchAgentServerPolicy aos nós de processamento. Para fazer isso, insira o comando a seguir. Substitua *my-worker-node-role* pelo perfil do IAM usada por seus nós de processamento do Kubernetes.

```
aws iam attach-role-policy \
--role-name my-worker-node-role \
--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

2. Insira o seguinte comando para instalar o complemento:

```
aws eks create-addon --cluster-name my-cluster-name --addon-name amazon-cloudwatch-observability
```

Configuração de início rápido para o Container Insights no Amazon EKS e no Kubernetes

> **Important**
>
> Se estiver instalando o Container Insights em um cluster do Amazon EKS, recomendamos que use o complemento de observabilidade do EKS do Amazon CloudWatch para a instalação em vez de usar as instruções desta seção. Além disso, para recuperar redes com computação acelerada, use o complemento Amazon CloudWatch Observability EKS. Para mais informações e instruções, consulte Instalar o complemento Amazon CloudWatch Observability do EKS.

Para concluir a configuração do Container Insights, siga as instruções de início rápido nesta seção. Se estiver instalando em um cluster do Amazon EKS e usar as instruções desta seção em ou após 6 de novembro de 2023, você instalará o Container Insights com observabilidade aprimorada para o Amazon EKS no cluster.

> **Important**
>
> Antes de executar as etapas desta seção, você deve ter verificado os pré-requisitos, inclusive as permissões do IAM. Para ter mais informações, consulte Verifique os pré-requisitos do .

Como alternativa, você pode seguir as instruções nas duas seções a seguir, Configurar o atendente do CloudWatch para coletar métricas do cluster e Enviar logs ao CloudWatch Logs. Essas seções fornecem mais detalhes sobre como o atendente do CloudWatch funciona com o Amazon EKS e o Kubernetes, mas necessitam que você execute mais etapas de instalação.

Com a versão original do Container Insights, as métricas coletadas e os registros ingeridos são cobrados como métricas personalizadas. Com o Container Insights com observabilidade aprimorada para o Amazon EKS, as métricas e os logs do Container Insights são cobrados por observação em vez de serem cobrados por métrica armazenada ou log ingerido. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

> **Note**
>
> A Amazon agora lançou o Fluent Bit como a solução de log padrão para o Container Insights com ganhos consideráveis de performance. Recomendamos usar o Fluent Bit em vez do Fluentd.

Início rápido ao usar o operador do agente do CloudWatch e o Fluent Bit

Existem duas configurações para Fluent Bit: uma versão otimizada e uma versão que fornece uma experiência mais semelhante ao Fluentd. A configuração de início rápido usa a versão otimizada. Para obter mais detalhes sobre a configuração compatível com o Fluentd, consulte Configurar o Fluent Bit como um DaemonSet para enviar logs ao CloudWatch Logs.

O operador do agente do CloudWatch corresponde a um contêiner adicional instalado em um cluster do Amazon EKS. Ele é modelado de acordo com o Operador do OpenTelemetry para Kubernetes. O operador gerencia o ciclo de vida útil dos recursos do Kubernetes em um cluster. Ele instala o agente do CloudWatch, a ferramenta DCGM Exporter (NVIDIA) e o monitor do AWS Neuron em um cluster do Amazon EKS e os gerencia. O Fluent Bit e o agente do CloudWatch para Windows são instalados diretamente em um cluster do Amazon EKS sem a necessidade de que o operador os gerencie.

Para obter uma solução de autoridade de certificação mais segura e repleta de funcionalidades, o operador do agente do CloudWatch requer o cert-manager, uma solução amplamente adotada para o gerenciamento de certificados TLS no Kubernetes. Usar o cert-manager simplifica o processo de obtenção, renovação, gerenciamento e uso desses certificados. Ele garante que os certificados sejam válidos e atualizados, bem como tenta renovar os certificados em um momento configurado antes da expiração. O cert-manager também facilita a emissão de certificados de diversas fontes com suporte, incluindo o AWS Certificate Manager Private Certificate Authority.

Como implantar o Container Insights usando o início rápido

1. Instale o cert-manager se ele ainda não estiver instalado no cluster. Para obter mais informações, consulte cert-manager Installation.
2. Instale as definições de recursos personalizados (CRD) ao inserir o comando apresentado a seguir.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl
 apply --server-side -f -
```

3. Instale o operador ao inserir o comando apresentado a seguir. Substitua *my-cluster-name* pelo nome do cluster do Amazon EKS ou do Kubernetes e *my-cluster-region* pelo nome da região em que os logs são publicados. Recomendamos usar a mesma região em que o cluster está implantado para reduzir os custos de transferência de dados de saída da AWS.

```
ClusterName=my-cluster-name
RegionName=my-cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 apply -f -
```

   Por exemplo, para implantar o Container Insights no cluster chamado `MyCluster` e publicar os logs e as métricas em Oeste dos EUA (Oregon), insira o comando a seguir.

```
ClusterName='MyCluster'
RegionName='us-west-2'
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
```

```
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 apply -f -
```

Migrar do Container Insights

Se você já tiver o Container Insights configurado em um cluster do Amazon EKS e desejar realizar a migração para o Container Insights com observabilidade aprimorada para o Amazon EKS, consulte Como atualizar para o Container Insights com observabilidade aprimorada para o Amazon EKS.

Excluir o Container Insights

Se você quiser remover o Container Insights depois de usar a configuração de início rápido, insira os comandos apresentados a seguir.

```
ClusterName=my-cluster-name
RegionName=my-cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 delete -f -
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl delete
 -f -
```

Configurar o atendente do CloudWatch para coletar métricas do cluster

> **Important**
>
> Se estiver instalando o Container Insights em um cluster do Amazon EKS, recomendamos que use o complemento de observabilidade do EKS do Amazon CloudWatch para a instalação em vez de usar as instruções desta seção. Para mais informações e instruções, consulte Instalar o complemento Amazon CloudWatch Observability do EKS.

Para configurar o Container Insights para coletar métricas, siga as etapas em Configuração de início rápido para o Container Insights no Amazon EKS e no Kubernetes ou siga as etapas nesta seção. Nas etapas a seguir, você configura o atendente do CloudWatch para ser capaz de coletar métricas dos clusters.

Se estiver instalando em um cluster do Amazon EKS e usar as instruções desta seção em ou após 6 de novembro de 2023, você instalará o Container Insights com observabilidade aprimorada para o Amazon EKS no cluster.

Etapa 1: Criar um namespace para o CloudWatch

Use a seguinte etapa para criar um namespace do Kubernetes chamado `amazon-cloudwatch` para o CloudWatch. Ignore essas etapas se você já tiver criado esse namespace.

Para criar um namespace para o CloudWatch

- Insira o comando da a seguir.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

Etapa 2: Criar uma conta de serviço no cluster

Use as etapas a seguir para criar uma conta de serviço para o atendente do CloudWatch, se ainda não tiver uma.

Para criar uma conta de serviço para o atendente do CloudWatch

- Insira o comando da a seguir.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cwagent/cwagent-serviceaccount.yaml
```

Se você não seguiu as etapas anteriores, mas já tem uma conta de serviço para o atendente do CloudWatch que deseja usar, deve garantir que ela tenha as regras a seguir. Além disso, no restante das etapas da instalação do Container Insights, você deve usar o nome da conta de serviço em vez de `cloudwatch-agent`.

```
rules:
  - apiGroups: [""]
    resources: ["pods", "nodes", "endpoints"]
    verbs: ["list", "watch"]
  - apiGroups: [ "" ]
```

```
    resources: [ "services" ]
    verbs: [ "list", "watch" ]
  - apiGroups: ["apps"]
    resources: ["replicasets", "daemonsets", "deployments", "statefulsets"]
    verbs: ["list", "watch"]
  - apiGroups: ["batch"]
    resources: ["jobs"]
    verbs: ["list", "watch"]
  - apiGroups: [""]
    resources: ["nodes/proxy"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes/stats", "configmaps", "events"]
    verbs: ["create", "get"]
  - apiGroups: [""]
    resources: ["configmaps"]
    resourceNames: ["cwagent-clusterleader"]
    verbs: ["get","update"]
  - nonResourceURLs: ["/metrics"]
    verbs: ["get", "list", "watch"]
```

Etapa 3: Criar um ConfigMap para o atendente do CloudWatch

Siga as etapas a seguir para criar um ConfigMap para o atendente do CloudWatch.

Para criar um ConfigMap para o atendente do CloudWatch

1. Faça download do YAML do ConfigMap para o host do cliente `kubectl` executando o seguinte comando:

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
   container-insights-monitoring/cwagent/cwagent-configmap.yaml
   ```

2. Edite o arquivo YAML obtido por download da seguinte forma:

   - cluster_name: na seção `kubernetes`, substitua `{{cluster_name}}` pelo nome do cluster. Remova os caracteres `{{}}`. Se preferir, caso esteja usando um cluster do Amazon EKS, você poderá excluir o campo `"cluster_name"` e o valor. Se fizer isso, o atendente do CloudWatch detectará o nome do cluster a partir das etiquetas do Amazon EC2.

3. (Opcional) Faça alterações adicionais no ConfigMap com base nos requisitos de monitoramento da seguinte forma:

- metrics_collection_interval: na seção `kubernetes`, você pode especificar com que frequência o atendente coleta métricas. O padrão é 60 segundos. O intervalo de coleta do cadvisor padrão em kubelet é de 15 segundos, portanto, não defina esse valor para menos de 15 segundos.
- endpoint_override: na seção `logs`, você poderá especificar o endpoint do CloudWatch Logs se desejar substituir o endpoint padrão. Você pode querer fazer isso se estiver publicando de um cluster em uma VPC e quiser que os dados vão para um VPC endpoint.
- force_flush_interval: na seção `logs`, você pode especificar o intervalo para agrupar em lote os eventos de log antes que eles sejam publicados no CloudWatch Logs. O padrão é 5 segundos.
- region: por padrão, o atendente publicou métricas para a Região em que o nó de processamento está localizado. Para substituir isso, você pode adicionar um campo `region` na seção `agent`: por exemplo, `"region":"us-west-2"`.
- Seção statsd: se quiser que o atendente do CloudWatch Logs também execute um StatsD em cada nó de processamento do cluster, você poderá adicionar uma seção `statsd` à seção `metrics`, conforme o exemplo a seguir. Para obter informações sobre outras opções do StatsD para essa seção, consulte Recuperar métricas personalizadas com o StatsD .

```
"metrics": {
  "metrics_collected": {
    "statsd": {
      "service_address":":8125"
    }
  }
}
```

Um exemplo completo da seção JSON é o seguinte.

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "kubernetes": {
                "cluster_name": "MyCluster",
                "metrics_collection_interval": 60
            }
```

```
        },
        "force_flush_interval": 5,
        "endpoint_override": "logs.us-east-1.amazonaws.com"
    },
    "metrics": {
        "metrics_collected": {
            "statsd": {
                "service_address": ":8125"
            }
        }
    }
}
```

4. Crie o ConfigMap no cluster executando o comando a seguir.

```
kubectl apply -f cwagent-configmap.yaml
```

Etapa 4: Implantar o atendente do CloudWatch como um DaemonSet

Para concluir a instalação do atendente do CloudWatch e começar a coletar métricas de contêiner, siga as etapas a seguir.

Para implantar o atendente do CloudWatch como um DaemonSet

1. • Para não usar o StatsD no cluster, insira o comando a seguir.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/cwagent/cwagent-
daemonset.yaml
```

   • Para usar o StatsD, siga estas etapas:

      a. Faça download do YAML do DaemonSet para o host do cliente `kubectl` executando o comando a seguir.

```
curl -O  https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-
mode/daemonset/container-insights-monitoring/cwagent/cwagent-daemonset.yaml
```

b. Remova o comentário da seção `port` no arquivo `cwagent-daemonset.yaml` da seguinte forma:

```
ports:
  - containerPort: 8125
    hostPort: 8125
    protocol: UDP
```

c. Implante o atendente do CloudWatch no cluster executando o comando a seguir.

```
kubectl apply -f cwagent-daemonset.yaml
```

d. Implante o agente do CloudWatch nos nós do Windows em seu cluster ao executar o comando apresentado a seguir. O receptor StatsD não é compatível com o agente do CloudWatch no Windows.

```
kubectl apply -f cwagent-daemonset-windows.yaml
```

2. Confirme se o atendente está implantado executando o comando a seguir.

```
kubectl get pods -n amazon-cloudwatch
```

Quando for concluído, o atendente do CloudWatch criará um grupo de logs chamado `/aws/containerinsights/`*`Cluster_Name`*`/performance` e enviará os eventos de log de performance a esse grupo de logs. Se você também configurar o atendente como um listener do StatsD, o atendente também escutará as métricas do StatsD na porta 8125 com o endereço IP do nó no qual o pod do aplicativo está programado.

Solução de problemas

Se o atendente não for implantado corretamente, tente o seguinte:

- Execute o comando a seguir para obter a lista de pods.

```
kubectl get pods -n amazon-cloudwatch
```

- Execute o comando a seguir e verifique os eventos na parte inferior da saída.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- Execute o comando a seguir para verificar os logs.

```
kubectl logs pod-name  -n amazon-cloudwatch
```

Usar o AWS Distro for OpenTelemetry

Você pode configurar o Container Insights para coletar métricas dos clusters do Amazon EKS usando o coletor AWS Distro for OpenTelemetry. Para obter mais informações sobre o AWS Distro for OpenTelemetry, consulte AWS Distro for OpenTelemetry.

> **Important**
>
> Se você instalar usando o AWS Distro para OpenTelemetry, você instala o Container Insights, mas não terá acesso ao Container Insights com observabilidade aprimorada para o Amazon EKS. Você não coletará as métricas detalhadas compatíveis com o Container Insights com observabilidade aprimorada para o Amazon EKS.

A forma de configuração do Container Insights depende se o cluster está hospedado em instâncias do Amazon EC2 ou em AWS Fargate (Fargate).

Clusters do Amazon EKS hospedados no Amazon EC2

Se você ainda não tiver feito isso, verifique se cumpriu os pré-requisitos, inclusive as funções do IAM necessárias. Para ter mais informações, consulte Verifique os pré-requisitos do .

A Amazon fornece um chart do Helm que pode ser usado para configurar o monitoramento do Amazon Elastic Kubernetes Service no Amazon EC2. Esse monitoramento usa coletor AWS Distro for OpenEleetry (ADOT) para métricas e Fluent Bit para logs. Portanto, o chart do Helm é útil para clientes que usam o Amazon EKS no Amazon EC2 e desejam coletar métricas e logs para enviar ao CloudWatch Container Insights. Para obter mais informações sobre o chart do Helm, consulte Gráfico ADOT Helm para EKS em métricas e logs do EC2 para o Amazon CloudWatch Container Insights.

Como alternativa, você também pode usar as instruções no restante desta seção.

Primeiro, implante o coletor do AWS Distro for OpenTelemetry como um DaemonSet inserindo o comando a seguir.

```
curl https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
deployment-template/eks/otel-container-insights-infra.yaml |
```

```
kubectl apply -f -
```

Insira o comando a seguir para confirmar se o coletor está em execução.

```
kubectl get pods -l name=aws-otel-eks-ci -n aws-otel-eks
```

Se a saída desse comando incluir vários pods no estado `Running`, o coletor está em execução e coletando métricas do cluster. O coletor cria um grupo de logs chamado `aws/containerinsights/`*`cluster-name`*`/performance` e envia os eventos de log de performance para ele.

Para obter informações sobre como visualizar as métricas do Container Insights no CloudWatch, consulte Visualizar métricas do Container Insights.

A AWS também forneceu documentação sobre o GitHub para esse cenário. Se quiser personalizar as métricas e os logs publicados pelo Container Insights, consulte https://aws-otel.github.io/docs/getting-started/container-insights/eks-infra.

Clusters do Amazon EKS hospedados no Fargate

Para obter instruções sobre como configurar e implantar um ADOT Collector para coletar métricas do sistema de cargas de trabalho implantadas em um cluster do Amazon EKS no Fargate e enviá-las para o CloudWatch Container Insights, consulte Container Insights EKS Fargate na documentação do AWS Distro for OpenTelemetry.

Enviar logs ao CloudWatch Logs

Para enviar logs de seus contêineres ao Amazon CloudWatch Logs, você pode usar Fluent Bit ou Fluentd. Para obter mais informações, consulte Fluent Bit e Fluentd.

Se você ainda não estiver usando o Fluentd, recomendamos usar o Fluent Bit por estes motivos:

- O Fluent Bit tem um espaço de recursos menor e é mais eficiente em termos de recursos com uso de memória e CPU do que o Fluentd. Para obter uma comparação mais detalhada, consulte Comparação de performance Fluent Bit e Fluentd.
- A imagem do Fluent Bit é desenvolvida e mantida pela AWS. Isso dá à AWS a capacidade de adotar novos recursos de imagem do Fluent Bit e responder a problemas muito mais rapidamente.

Tópicos

- Comparação de performance Fluent Bit e Fluentd
- Configurar o Fluent Bit como um DaemonSet para enviar logs ao CloudWatch Logs
- (Opcional) Configurar o Fluentd como um DaemonSet para enviar logs ao CloudWatch Logs
- (Opcional) Configurar o registro do ambiente de gerenciamento do Amazon EKS
- (Opcional) Habilitar logs de acesso do App Mesh Envoy
- (Opcional) Habilite o recurso Use_Kubelet para clusters grandes

Comparação de performance Fluent Bit e Fluentd

As tabelas a seguir mostram a vantagem de performance que o Fluent Bit tem sobre o Fluentd em uso de memória e CPU. Os números a seguir são apenas para referência e podem se alterados de acordo com o ambiente.

Logs por segundo	Uso da CPU pelo Fluentd	Uso da CPU pelo Fluent Bit com configuração compatível com o Fluentd	Uso da CPU pelo Fluent Bit com configuração otimizada
100	0,35 vCPU	0,02 vCPU	0,02 vCPU
1.000	0,32 vCPU	0,14 vCPU	0,11 vCPU
5.000	0,85 vCPU	0,48 vCPU	0,30 vCPU
10.000	0,94 vCPU	0,60 vCPU	0,39 vCPU

Logs por segundo	Uso de memória do Fluentd	Uso de memória U pelo Fluent Bit com configuração compatível com o Fluentd	Uso de memória pelo Fluent Bit com configuração otimizada
100	153 MB	46 MB	37 MB
1.000	270 MB	45 MB	40 MB

Logs por segundo	Uso de memória do Fluentd	Uso de memória U pelo Fluent Bit com configuração compatível com o Fluentd	Uso de memória pelo Fluent Bit com configuração otimizada
5.000	320 MB	55 MB	45 MB
10.000	375 MB	92 MB	75 MB

Configurar o Fluent Bit como um DaemonSet para enviar logs ao CloudWatch Logs

As seções a seguir ajudam a implantar o Fluent Bit para enviar logs de contêineres ao CloudWatch Logs.

Tópicos

- O que se difere se você já estiver usando Fluentd
- Configurar o Fluent Bit
- Suporte a logs de várias linhas
- (Opcional) Reduzir o volume de log do Fluent Bit
- Solução de problemas
- Painel

O que se difere se você já estiver usando Fluentd

Se você já estiver usando o Fluentd para enviar logs de contêineres ao CloudWatch Logs, leia esta seção para ver as diferenças entre o Fluentd e o Fluent Bit. Se ainda não estiver usando o Fluentd com o Container Insights, você pode pular para Configurar o Fluent Bit.

Fornecemos duas configurações padrão para o Fluent Bit:

- Configuração otimizada do Fluent Bit: uma configuração alinhada às práticas recomendadas do Fluent Bit.
- Configuração compatível com Fluentd: uma configuração alinhada ao comportamento Fluentd o máximo possível.

A lista a seguir explica as diferenças entre o Fluentd e cada configuração do Fluent Bit em detalhes.

- Diferenças nos nomes de fluxo de logs: se você usar a configuração otimizada do Fluent Bit, os nomes de fluxo de logs serão diferentes.

  Em `/aws/containerinsights/Cluster_Name/application`

  - A configuração otimizada do Fluent Bit envia logs para *kubernetes-nodeName*-`application.var.log.containers.`*kubernetes-podName*_*kubernetes-namespace*_*kubernetes-container-name*-*kubernetes-containerID*
  - O Fluentd envia logs para *kubernetes-podName*_*kubernetes-namespace*_*kubernetes-containerName*_*kubernetes-containerID*

  Em `/aws/containerinsights/Cluster_Name/host`

  - A configuração otimizada do Fluent Bit envia logs para *kubernetes-nodeName*.*host-log-file*
  - O Fluentd envia logs para *host-log-file*-*Kubernetes-NodePrivateIp*

  Em `/aws/containerinsights/Cluster_Name/dataplane`

  - A configuração otimizada do Fluent Bit envia logs para *kubernetes-nodeName*.*dataplaneServiceLog*
  - O Fluentd envia logs para *dataplaneServiceLog*-*Kubernetes-nodeName*
- Os arquivos de log `kube-proxy` e `aws-node` = que o Container Insights grava estão em locais diferentes. Na configuração do Fluentd, eles estão em `/aws/containerinsights/Cluster_Name/application`. Na configuração otimizada do Fluent Bit, eles estão em `/aws/containerinsights/Cluster_Name/dataplane`.
- A maioria dos metadados, como `pod_name` e `namespace_name` são os mesmos no Fluent Bit e no Fluentd, mas os seguintes são diferentes.
  - A configuração otimizada do Fluent Bit usa `docker_id`, e o Fluentd usa `Docker.container_id`.
  - Nenhuma das duas configurações do Fluent Bit usa os metadados a seguir. Eles estão presentes apenas no Fluentd: `container_image_id`, `master_url`, `namespace_id` e `namespace_labels`.

Configurar o Fluent Bit

Para configurar o Fluent Bit para coletar logs de seus contêineres, siga as etapas em Configuração de início rápido para o Container Insights no Amazon EKS e no Kubernetes ou siga as etapas nesta seção.

Com qualquer dos dois métodos, a função do IAM que está anexada aos nós do cluster deve ter permissões suficientes. Para obter mais informações sobre as permissões necessárias para executar um cluster do Amazon EKS, consultePolíticas, funções, e permissões do Amazon EKS IAM no Manual do usuário do Amazon EKS.

Nas etapas a seguir, você configura o Fluent Bit como um daemonSet para enviar logs ao CloudWatch Logs. Ao concluir esta etapa, o Fluent Bit criará os grupos de log a seguir, caso eles ainda não existam.

> **Important**
>
> Se você já tiver o Fluentd configurado no Container Insights e o daemonSet do Fluentd não estiver sendo executado conforme o esperado (isso poderá acontecer se você usar o runtime `containerd`), desinstale-o antes de instalar o Fluent Bit para evitar que o Fluent Bit processe as mensagens de log de erros do Fluentd. Caso contrário, você deverá desinstalar o Fluentd imediatamente após ter instalado o Fluent Bit com êxito. A desinstalação do Fluentd após a instalação do Fluent Bit garante a continuidade do registro em logs durante esse processo de migração. Apenas um dentre o Fluent Bit e o Fluentd é necessário para enviar logs ao CloudWatch Logs.

Nome do grupo de logs	Origem do log
`/aws/containerinsights/` *`Cluster_Name`* `/application`	Todos os arquivos de log em `/var/log/containers`
`/aws/containerinsights/` *`Cluster_Name`* `/host`	Logs de `/var/log/dmesg` , `/var/log/secure` e `/var/log/messages`
`/aws/containerinsights/` *`Cluster_Name`* `/dataplane`	Os logs no `/var/log/journal` para `kubelet.service` , `kubeproxy.service` e `docker.service` .

Para instalar o Fluent Bit para enviar logs de contêineres ao CloudWatch Logs

1. Se você ainda não tem um namespace chamado `amazon-cloudwatch`, crie um inserindo este comando:

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

2. Execute o comando a seguir para criar um ConfigMap chamado `cluster-info` com o nome do cluster e a região para enviar logs. Substitua *`cluster-name`* e *`cluster-region`* pelo nome e pela região de seu cluster.

```
ClusterName=cluster-name
RegionName=cluster-region
FluentBitHttpPort='2020'
FluentBitReadFromHead='Off'
[[ ${FluentBitReadFromHead} = 'On' ]] && FluentBitReadFromTail='Off'||
 FluentBitReadFromTail='On'
[[ -z ${FluentBitHttpPort} ]] && FluentBitHttpServer='Off' ||
 FluentBitHttpServer='On'
kubectl create configmap fluent-bit-cluster-info \
--from-literal=cluster.name=${ClusterName} \
--from-literal=http.server=${FluentBitHttpServer} \
--from-literal=http.port=${FluentBitHttpPort} \
--from-literal=read.head=${FluentBitReadFromHead} \
--from-literal=read.tail=${FluentBitReadFromTail} \
--from-literal=logs.region=${RegionName} -n amazon-cloudwatch
```

   Neste comando, o `FluentBitHttpServer` para monitorar métricas de plugin é ativado por padrão. Para desativá-lo, altere a terceira linha no comando para `FluentBitHttpPort=''` (string vazia) no comando.

   Também por padrão, o Fluent Bit lê arquivos de log a partir do final e capturará somente novos logs depois de implantado. Caso queira o oposto, defina `FluentBitReadFromHead='On'`, e ele coletará todos os logs no sistema de arquivos.

3. Baixe e implante o DaemonSet do Fluent Bit no cluster executando os comandos a seguir.
   - Se você quiser obter a configuração otimizada do Fluent Bit para computadores com o sistema Linux, execute o comando apresentado a seguir.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-
bit.yaml
```

- Se você quiser obter a configuração otimizada do Fluent Bit para computadores com o sistema Windows, execute o comando apresentado a seguir.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-bit-
windows.yaml
```

- Se você estiver usando computadores com o sistema Linux e desejar uma configuração do Fluent Bit mais semelhante ao Fluentd, execute o comando apresentado a seguir.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-bit-
compatible.yaml
```

> **Important**
>
> Por padrão, a configuração do daemonset do Fluent Bit define o nível de log como INFO, o que pode resultar em maiores custos de ingestão do CloudWatch Logs. Se você quiser reduzir o volume e os custos de ingestão de logs, altere o nível de log para ERROR. Para obter mais informações sobre como reduzir o volume de log, consulte (Opcional) Reduzir o volume de log do Fluent Bit.

4. Valide a implantação inserindo o comando a seguir. Cada nó deve ter um pod chamado fluent-bit-*.

```
kubectl get pods -n amazon-cloudwatch
```

As etapas acima criarão os seguintes recursos no cluster:

- Uma conta de serviço chamada `Fluent-Bit` no namespace `amazon-cloudwatch`. Essa conta de serviço é usada para executar o DaemonSet do Fluent Bit. Para obter mais informações, consulte Gerenciar contas de serviço (em inglês) na Referência do Kubernetes.
- Uma função do cluster chamada `Fluent-Bit-role` no namespace `amazon-cloudwatch`. Essa função do cluster concede permissões `get`, `list` e `watch` em logs de pod para a conta de serviço `Fluent-Bit`. Para obter mais informações, consulte Visão geral da API (em inglês) na Referência do Kubernetes.
- Um ConfigMap chamado `Fluent-Bit-config` no namespace `amazon-cloudwatch`. Esse ConfigMap contém a configuração a ser usada pelo Fluent Bit. Para obter mais informações, consulte Configurar um pod para usar um ConfigMap na documentação do Kubernetes Tasks.

Se pretende verificar a configuração do Fluent Bit, siga estas etapas.

Verifique a configuração do Fluent Bit

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. Certifique-se de que você está na região na qual implantou o Fluent Bit.
4. Confira a lista de grupos de logs na região. Você deve ver o seguinte:
   - `/aws/containerinsights/`*`Cluster_Name`*`/application`
   - `/aws/containerinsights/`*`Cluster_Name`*`/host`
   - `/aws/containerinsights/`*`Cluster_Name`*`/dataplane`
5. Navegue até um desses grupos de log e marque Last Event Time (Hora do último evento) para os fluxos de log. Se for recente em relação à implantação do Fluent Bit, a instalação será verificada.

   Pode haver um pequeno atraso na criação do grupo de logs `/dataplane`. Isso é normal, pois esses grupos de log só são criados quando o Fluent Bit começa a enviar logs a esse grupo de logs.

Suporte a logs de várias linhas

Para obter informações sobre como usar o Fluent Bit com logs de várias linhas, consulte as seções a seguir da documentação do Fluent Bit:

- Análise de várias linhas
- Várias linhas e contêineres (v1.8)
- Núcleo de várias linhas (v1.8)
- Sempre use várias linhas multilinha na entrada final

(Opcional) Reduzir o volume de log do Fluent Bit

Por padrão, enviamos logs de aplicação do Fluent Bit e metadados do Kubernetes ao CloudWatch. Para reduzir o volume de dados que estão sendo enviados ao CloudWatch, você pode impedir que uma ou ambas as fontes de dados sejam enviadas ao CloudWatch.

Para interromper os logs de aplicação do Fluent Bit, remova a seção a seguir do arquivo `Fluent-Bit.yaml`.

```
[INPUT]
        Name                tail
        Tag                 application.*
        Path                /var/log/containers/fluent-bit*
        Parser              docker
        DB                  /fluent-bit/state/flb_log.db
        Mem_Buf_Limit       5MB
        Skip_Long_Lines     On
        Refresh_Interval    10
```

Para remover os metadados do Kubernetes, a fim de que não sejam anexados aos eventos de log enviados ao CloudWatch, adicione uma linha à seção `application-log.conf` do arquivo `Fluent-Bit.yaml`. Substitua *<Metadata_1>* e os campos semelhantes pelos identificadores de metadados reais.

```
application-log.conf: |
    [FILTER]
        Name                nest
        Match               application.*
        Operation           lift
        Nested_under        kubernetes
        Add_prefix          Kube.

    [FILTER]
        Name                modify
        Match               application.*
```

```
        Remove              Kube.<Metadata_1>
        Remove              Kube.<Metadata_2>
        Remove              Kube.<Metadata_3>

    [FILTER]
        Name                nest
        Match               application.*
        Operation           nest
        Wildcard            Kube.*
        Nested_under        kubernetes
        Remove_prefix       Kube.
```

Solução de problemas

Caso não veja esses grupos de log e esteja procurando na região correta, confira os logs para os pods do daemonSet do Fluentd para procurar o erro.

Execute o comando a seguir para certificar-se de que o status seja `Running`.

```
kubectl get pods -n amazon-cloudwatch
```

Se os logs tiverem erros relacionados às permissões do IAM, verifique a função do IAM que está anexada aos nós do cluster. Para obter mais informações sobre as permissões necessárias para executar um cluster do Amazon EKS, consulte Políticas, funções, e permissões do Amazon EKS IAM no Manual do usuário do Amazon EKS.

Se o status do pod for `CreateContainerConfigError`, obtenha o erro exato executando o comando a seguir.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

Painel

É possível criar um painel para monitorar métricas de cada plugin em execução. Você pode visualizar dados para bytes de entrada e saída e para taxas de processamento de registros, bem como erros de saída e taxas de repetição/falha. Para visualizar essas métricas, será necessário instalar o atendente do CloudWatch com a coleção de métricas do Prometheus para clusters do Amazon EKS e do Kubernetes. Consulte Instalar o atendente do CloudWatch com a coleção de métricas do Prometheus em clusters do Amazon EKS e do Kubernetes para obter mais informações sobre como configurar o painel.

> **Note**
>
> Antes de configurar este painel, é necessário configurar as métricas do Container Insights para o Prometheus. Para ter mais informações, consulte Monitoramento de métricas do Container Insights Prometheus.

Para criar um painel para métricas do Prometheus do Fluent Bits

1. Crie variáveis de ambiente, substituindo os valores à direita nas linhas a seguir para corresponder a sua implantação.

```
DASHBOARD_NAME=your_cw_dashboard_name
REGION_NAME=your_metric_region_such_as_us-west-1
CLUSTER_NAME=your_kubernetes_cluster_name
```

2. Crie o painel executando o comando a seguir.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/fluent-bit/cw_dashboard_fluent_bit.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| xargs -0 aws cloudwatch put-dashboard --dashboard-name ${DASHBOARD_NAME} --
dashboard-body
```

(Opcional) Configurar o Fluentd como um DaemonSet para enviar logs ao CloudWatch Logs

> **Warning**
>
> O suporte ao Container Insights para o Fluentd está agora em modo de manutenção, o que significa que a AWS não fornecerá outras atualizações do Fluentd e que estamos planejando descontinuá-lo em um futuro próximo. Além disso, a configuração atual do Fluentd para o Container Insights está usando uma versão antiga da imagem do Fluentd `fluent/fluentd-kubernetes-daemonset:v1.10.3-debian-cloudwatch-1.0` que não tem os últimos patches de melhorias e segurança. Para obter a imagem do Fluentd mais recente com suporte na comunidade de código aberto, consulte fluentd-kubernetes-daemonset.

É altamente recomendável migrar para usar o FluentBit com o Container Insights sempre que possível. Usar o FluentBit como encaminhador de log para o Container Insights proporciona ganhos consideráveis de performance.
Para obter mais informações, consulte Configurar o Fluent Bit como um DaemonSet para enviar logs ao CloudWatch Logs e O que se difere se você já estiver usando Fluentd.

Para configurar o Fluentd para coletar logs de seus contêineres, siga as etapas em Configuração de início rápido para o Container Insights no Amazon EKS e no Kubernetes ou siga as etapas nesta seção. Nas etapas a seguir, você configura o Fluentd como um DaemonSet para enviar logs ao CloudWatch Logs. Ao concluir esta etapa, o Fluentd criará os grupos de log a seguir, caso eles ainda não existam.

Nome do grupo de logs	Origem do log
`/aws/containerinsights/ Cluster_Name /application`	Todos os arquivos de log em `/var/log/containers`
`/aws/containerinsights/ Cluster_Name /host`	Logs de `/var/log/dmesg` , `/var/log/secure` e `/var/log/messages`
`/aws/containerinsights/ Cluster_Name /dataplane`	Os logs no `/var/log/journal` para `kubelet.service` , `kubeproxy.service` e `docker.service` .

Etapa 1: Criar um namespace para o CloudWatch

Use a seguinte etapa para criar um namespace do Kubernetes chamado `amazon-cloudwatch` para o CloudWatch. Ignore essas etapas se você já tiver criado esse namespace.

Para criar um namespace para o CloudWatch

- Insira o comando da a seguir.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

Etapa 2: instalar o Fluentd

Inicie esse processo fazendo download do Fluentd. Ao concluir essas etapas, a implantação criará os seguintes recursos no cluster:

- Uma conta de serviço chamada `fluentd` no namespace `amazon-cloudwatch`. Esst conta de serviço é usada para executar o DaemonSet do Fluentd. Para obter mais informações, consulte Gerenciar contas de serviço (em inglês) na Referência do Kubernetes.
- Uma função do cluster chamada `fluentd` no namespace `amazon-cloudwatch`. Essa função do cluster concede permissões `get`, `list` e `watch` em logs de pod para a conta de serviço `fluentd`. Para obter mais informações, consulte Visão geral da API (em inglês) na Referência do Kubernetes.
- Um ConfigMap chamado `fluentd-config` no namespace `amazon-cloudwatch`. Este ConfigMap contém a configuração a ser usada pelo Fluentd. Para obter mais informações, consulte Configurar um pod para usar um ConfigMap na documentação do Kubernetes Tasks.

Para instalar o Fluentd

1. Crie um ConfigMap chamado `cluster-info` com o nome do cluster e a região da AWS à qual os logs serão enviados. Execute o comando a seguir atualizando os espaços reservados com os nomes do cluster e da região.

   ```
   kubectl create configmap cluster-info \
   --from-literal=cluster.name=cluster_name \
   --from-literal=logs.region=region_name -n amazon-cloudwatch
   ```

2. Faça download e implante o DaemonSet do Fluentd no cluster executando o comando a seguir. Verifique se está usando a imagem do contêiner com a arquitetura correta. O exemplo de manifesto de só funciona em instâncias x86 e entrará em `CrashLoopBackOff` se você tiver instâncias Advanced RISC Machine (ARM) em seu cluster. O DaemonSet do Fluentd não tem uma imagem do Docker oficial de várias arquiteturas que permita usar uma etiqueta para várias imagens subjacentes e deixar o runtime do contêiner escolher o certo. A imagem de ARM do Fluentd usa uma etiqueta diferente com um sufixo `arm64`.

   ```
   kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
   container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
   daemonset/container-insights-monitoring/fluentd/fluentd.yaml
   ```

> **Note**
>
> Devido a uma alteração recente para otimizar a configuração do Fluentd e minimizar o impacto das solicitações da API do Fluentd nos endpoints da API do Kubernetes, a opção "Watch" para filtros do Kubernetes foi desabilitada por padrão. Para obter mais detalhes, consulte fluent-plugin-kubernetes_metadata_filter.

3. Valide a implantação executando o comando a seguir. Cada nó deve ter um pod chamado `fluentd-cloudwatch-*`.

```
kubectl get pods -n amazon-cloudwatch
```

Etapa 3: Verificar a configuração do Fluentd

Para verificar a configuração do Fluentd, use as etapas a seguir.

Para verificar a configuração do Fluentd para o Container Insights

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs. Certifique-se de que você esteja na região na qual implantou o Fluentd para seus contêineres.

   Na lista de grupos de log na região, você deve ver o seguinte:

   - `/aws/containerinsights/Cluster_Name/application`
   - `/aws/containerinsights/Cluster_Name/host`
   - `/aws/containerinsights/Cluster_Name/dataplane`

   Caso você veja esses grupos de log, a configuração do Fluentd estará verificada.

Suporte a logs de várias linhas

Em 19 de agosto de 2019, adicionamos suporte a logs de várias linhas para os logs coletados pelo Fluentd.

Por padrão, o iniciador da entrada do log de várias linhas é qualquer caractere sem espaço em branco. Isso significa que todas as linhas do log que comecem com um caractere que não contenha espaço em branco são consideradas como uma nova entrada do log de várias linhas.

Se seus próprios logs de aplicação usarem um iniciador de várias linhas diferente, será possível oferecer suporte a eles fazendo duas alterações no arquivo `fluentd.yaml`.

Primeiro, exclua-os do suporte padrão de várias linhas adicionando os nomes de caminho dos arquivos de log a um campo `exclude_path` na seção `containers` de `fluentd.yaml`. Veja um exemplo a seguir.

```
<source>
  @type tail
  @id in_tail_container_logs
  @label @containers
  path /var/log/containers/*.log
  exclude_path ["full_pathname_of_log_file*", "full_pathname_of_log_file2*"]
```

Adicione um bloco para seus arquivos de log ao arquivo `fluentd.yaml`. O exemplo abaixo é usado para o arquivo de log do agente do CloudWatch, que usa uma expressão regular de carimbo de data/hora como o iniciador de várias linhas. Você pode copiar esse bloco e adicioná-lo ao `fluentd.yaml`. Altere as linhas indicadas para refletir o nome do arquivo de log da aplicação e o iniciador de várias linhas que você deseja usar.

```
<source>
  @type tail
  @id in_tail_cwagent_logs
  @label @cwagentlogs
  path /var/log/containers/cloudwatch-agent*
  pos_file /var/log/cloudwatch-agent.log.pos
  tag *
  read_from_head true
 <parse>
    @type json
    time_format %Y-%m-%dT%H:%M:%S.%NZ
 </parse>
</source>
```

```
<label @cwagentlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_cwagent
  </filter>

  <filter **>
    @type record_transformer
    @id filter_cwagent_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
 </filter>

  <filter **>
    @type concat
    key log
    multiline_start_regexp /^\d{4}[-/]\d{1,2}[-/]\d{1,2}/
    separator ""
    flush_interval 5
    timeout_label @NORMAL
 </filter>

 <match **>
    @type relabel
    @label @NORMAL
 </match>
</label>
```

(Opcional) Redução do volume de log do Fluentd

Por padrão, enviamos logs de aplicação do Fluentd e metadados do Kubernetes ao CloudWatch. Para reduzir o volume de dados que estão sendo enviados ao CloudWatch, você pode impedir que uma ou ambas as fontes de dados sejam enviadas ao CloudWatch.

Para interromper os logs da aplicação Fluentd, remova a seção a seguir do arquivo `fluentd.yaml`.

```
<source>
  @type tail
  @id in_tail_fluentd_logs
  @label @fluentdlogs
  path /var/log/containers/fluentd*
```

```
  pos_file /var/log/fluentd.log.pos
  tag *
  read_from_head true
   <parse>
   @type json
   time_format %Y-%m-%dT%H:%M:%S.%NZ
  </parse>
</source>

<label @fluentdlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_fluentd
  </filter>

  <filter **>
    @type record_transformer
    @id filter_fluentd_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
   </filter>

   <match **>
     @type relabel
     @label @NORMAL
   </match>
</label>
```

Para remover os metadados do Kubernetes, a fim de que não sejam anexados aos eventos de log enviados ao CloudWatch, adicione uma linha à seção `record_transformer` do arquivo `fluentd.yaml`. Na origem do log em que você deseja remover esses metadados, adicione a seguinte linha.

```
remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
```

Por exemplo:

```

<filter **>
  @type record_transformer
```

```
  @id filter_containers_stream_transformer
  <record>
    stream_name ${tag_parts[3]}
  </record>
  remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
</filter>
```

Solução de problemas

Caso não veja esses grupos de log e esteja procurando na região correta, verifique os logs para os pods do DaemonSet do Fluentd para procurar o erro.

Execute o comando a seguir para certificar-se de que o status seja `Running`.

```
kubectl get pods -n amazon-cloudwatch
```

Nos resultados do comando anterior, observe o nome do pod que começa com `fluentd-cloudwatch`. Use o nome desse pod no comando a seguir.

```
kubectl logs pod_name -n amazon-cloudwatch
```

Se os logs tiverem erros relacionados às permissões do IAM, verifique a função do IAM anexada aos nós do cluster. Para obter mais informações sobre as permissões necessárias para executar um cluster do Amazon EKS, consulte[Políticas, funções, e permissões do Amazon EKS IAM](#) no Manual do usuário do Amazon EKS.

Se o status do pod for `CreateContainerConfigError`, obtenha o erro exato executando o comando a seguir.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

Se o status do pod for`CrashLoopBackOff`, verifique se a arquitetura da imagem do contêiner do Fluentd é a mesma do nó quando você instalou o Fluentd. Se o cluster tiver nós x86 e ARM64, será possível usar um rótulo kubernetes.io/arch para colocar as imagens no nó correto. Para obter mais informações, consulte [kuberntes.io/arch](#).

(Opcional) Configurar o registro do ambiente de gerenciamento do Amazon EKS

Se estiver usando o Amazon EKS, opcionalmente, você poderá habilitar o registro do ambiente de gerenciamento do Amazon EKS para fornecer logs de diagnóstico e auditoria diretamente do

ambiente de gerenciamento do Amazon EKS para o CloudWatch Logs. Para obter mais informações, consulte Amazon EKS Control Plane Logging (Registro em log do plano de controle do Amazon EKS).

(Opcional) Habilitar logs de acesso do App Mesh Envoy

É possível configurar o Container Insights Fluentd para enviar logs de acesso do App Mesh Envoy ao CloudWatch Logs. Para obter mais informações, consulte Logging (Registro em logs).

Como fazer com que os logs de acesso do Envoy sejam enviados ao CloudWatch Logs

1. Configure o Fluentd no cluster. Para ter mais informações, consulte (Opcional) Configurar o Fluentd como um DaemonSet para enviar logs ao CloudWatch Logs.
2. Configurar logs de acesso do Envoy para seus nós virtuais. Para obter instruções, consulte Logging (Registro em logs). Configure o caminho de log como **`/dev/stdout`** em cada nó virtual.

Ao concluir, os logs de acesso do Envoy são enviados para o grupo de logs `/aws/containerinsights/`*`Cluster_Name`*`/application`.

(Opcional) Habilite o recurso Use_Kubelet para clusters grandes

Por padrão, o recurso Use_Kubelet está desabilitado no plugin FluentBit Kubernetes. A habilitação desse recurso pode reduzir o tráfego para o servidor de API e mitigar o problema de gargalo do servidor de API. Recomendamos a habilitação desse recurso para clusters grandes.

Para habilitar Use_Kubelet, primeiro adicione os nós e as permissões de nós/proxy à configuração clusterRole.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: fluent-bit-role
rules:
  - nonResourceURLs:
      - /metrics
    verbs:
      - get
  - apiGroups: [""]
    resources:
      - namespaces
```

```
      - pods
      - pods/logs
      - nodes
      - nodes/proxy
    verbs: ["get", "list", "watch"]
```

Na configuração do DaemonSet, esse recurso precisa de acesso à rede host. A versão da imagem para `amazon/aws-for-fluent-bit` deve ser 2.12.0 ou posterior, ou a versão da imagem de bits fluentes deve ser 1.7.2 ou posterior.

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluent-bit
  namespace: amazon-cloudwatch
  labels:
    k8s-app: fluent-bit
    version: v1
    kubernetes.io/cluster-service: "true"
spec:
  selector:
    matchLabels:
      k8s-app: fluent-bit
  template:
    metadata:
      labels:
        k8s-app: fluent-bit
        version: v1
        kubernetes.io/cluster-service: "true"
    spec:
      containers:
      - name: fluent-bit
        image: amazon/aws-for-fluent-bit:2.19.0
        imagePullPolicy: Always
        env:
            - name: AWS_REGION
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: logs.region
            - name: CLUSTER_NAME
              valueFrom:
                configMapKeyRef:
```

```
                name: fluent-bit-cluster-info
                key: cluster.name
        - name: HTTP_SERVER
          valueFrom:
            configMapKeyRef:
              name: fluent-bit-cluster-info
              key: http.server
        - name: HTTP_PORT
          valueFrom:
            configMapKeyRef:
              name: fluent-bit-cluster-info
              key: http.port
        - name: READ_FROM_HEAD
          valueFrom:
            configMapKeyRef:
              name: fluent-bit-cluster-info
              key: read.head
        - name: READ_FROM_TAIL
          valueFrom:
            configMapKeyRef:
              name: fluent-bit-cluster-info
              key: read.tail
        - name: HOST_NAME
          valueFrom:
            fieldRef:
              fieldPath: spec.nodeName
        - name: HOSTNAME
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.name
        - name: CI_VERSION
          value: "k8s/1.3.8"
      resources:
        limits:
          memory: 200Mi
        requests:
          cpu: 500m
          memory: 100Mi
      volumeMounts:
      # Please don't change below read-only permissions
      - name: fluentbitstate
        mountPath: /var/fluent-bit/state
      - name: varlog
```

```
          mountPath: /var/log
          readOnly: true
        - name: varlibdockercontainers
          mountPath: /var/lib/docker/containers
          readOnly: true
        - name: fluent-bit-config
          mountPath: /fluent-bit/etc/
        - name: runlogjournal
          mountPath: /run/log/journal
          readOnly: true
        - name: dmesg
          mountPath: /var/log/dmesg
          readOnly: true
      terminationGracePeriodSeconds: 10
      hostNetwork: true
      dnsPolicy: ClusterFirstWithHostNet
      volumes:
      - name: fluentbitstate
        hostPath:
          path: /var/fluent-bit/state
      - name: varlog
        hostPath:
          path: /var/log
      - name: varlibdockercontainers
        hostPath:
          path: /var/lib/docker/containers
      - name: fluent-bit-config
        configMap:
          name: fluent-bit-config
      - name: runlogjournal
        hostPath:
          path: /run/log/journal
      - name: dmesg
        hostPath:
          path: /var/log/dmesg
      serviceAccountName: fluent-bit
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
        effect: NoSchedule
      - operator: "Exists"
        effect: "NoExecute"
      - operator: "Exists"
```

```
        effect: "NoSchedule"
```

A configuração do Kubernetes Plugin deve ser semelhante à seguinte:

```
[FILTER]
        Name                kubernetes
        Match               application.*
        Kube_URL            https://kubernetes.default.svc:443
        Kube_Tag_Prefix     application.var.log.containers.
        Merge_Log           On
        Merge_Log_Key       log_processed
        K8S-Logging.Parser  On
        K8S-Logging.Exclude Off
        Labels              Off
        Annotations         Off
        Use_Kubelet         On
        Kubelet_Port        10250
        Buffer_Size         0
```

Atualizar ou excluir o Container Insights no Amazon EKS e no Kubernetes

Siga as etapas destas seções para atualizar a imagem de contêiner do atendente do CloudWatch ou para remover o Container Insights de um cluster do Amazon EKS ou do Kubernetes.

Tópicos

- Como atualizar para o Container Insights com observabilidade aprimorada para o Amazon EKS
- Atualizar a imagem do contêiner do atendente do CloudWatch
- Exclusão do agente do CloudWatch e do Fluent Bit para o Container Insights

Como atualizar para o Container Insights com observabilidade aprimorada para o Amazon EKS

> **Important**
>
> Se você estiver atualizando ou instalando o Container Insights em um cluster do Amazon EKS, recomendamos usar o complemento Observability do Amazon CloudWatch para o EKS para a instalação, em vez de usar as instruções apresentadas nesta seção. Além disso, para recuperar as métricas de computação acelerada, é necessário usar o complemento Observability do Amazon CloudWatch para o EKS. Para mais informações e instruções, consulte Instalar o complemento Amazon CloudWatch Observability do EKS.

O Container Insights com observabilidade aprimorada para o Amazon EKS é a versão mais recente do Container Insights. Ele coleta métricas detalhadas de clusters que executam o Amazon EKS e oferece painéis de controle selecionados e imediatamente utilizáveis para detalhar a telemetria de aplicações e infraestrutura. Para obter mais informações sobre esta versão do Container Insights, consulte Container Insights com observabilidade aprimorada para o Amazon EKS.

Se você tiver instalado a versão original do Container Insights em um cluster do Amazon EKS e quiser atualizá-lo para a versão mais recente com observabilidade aprimorada, siga as instruções desta seção.

> **Important**
>
> Antes de executar as etapas desta seção, você deve ter verificado os pré-requisitos, incluindo o cert-manager. Para ter mais informações, consulte Início rápido ao usar o operador do agente do CloudWatch e o Fluent Bit.

Atualizar um cluster do Amazon EKS para o Container Insights com observabilidade aprimorada para o Amazon EKS

1. Instale o operador do agente do CloudWatch ao inserir o comando apresentado a seguir. Substitua *my-cluster-name* pelo nome do cluster do Amazon EKS ou do Kubernetes e *my-cluster-region* pelo nome da região em que os logs são publicados. Recomendamos usar a mesma região em que o cluster está implantado para reduzir os custos de transferência de dados de saída da AWS.

   ```
   ClusterName=my-cluster-name
   RegionName=my-cluster-region
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
   container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
   {{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
    apply -f -
   ```

   Se você notar uma falha causada por recursos conflitantes, é provável que já tenha o agente do CloudWatch e o Fluent Bit com os componentes associados, como o ServiceAccount, o ClusterRole e o ClusterRoleBinding instalados no cluster. Quando o operador do agente do CloudWatch tenta instalar o agente do CloudWatch e os componentes associados, se detectar quaisquer alterações no conteúdo, por padrão, apresentará falhas na instalação ou na atualização para evitar a substituição do estado dos recursos no cluster. Recomendamos excluir

qualquer agente do CloudWatch existente com a configuração para o Container Insights que tenha sido instalado anteriormente no cluster e, em seguida, instalar o operador do agente do CloudWatch.

2. (Opcional) Para aplicar uma configuração personalizada do Fluent Bit existente, é necessário atualizar o configmap associado ao daemonset do Fluent Bit. O operador do agente do CloudWatch fornece uma configuração padrão para o Fluent Bit, e é possível substituir ou modificar essa configuração conforme necessário. Para aplicar uma configuração personalizada, siga as etapas apresentadas a seguir.

   a. Abra a configuração existente ao inserir o comando apresentado a seguir.

   ```
   kubectl edit cm fluent-bit-config -n amazon-cloudwatch
   ```

   b. Faça as alterações no arquivo e, em seguida, insira `:wq` para salvar o arquivo e sair do modo de edição.

   c. Reinicie o Fluent Bit ao inserir o comando apresentado a seguir.

   ```
   kubectl rollout restart fluent-bit -n amazon-cloudwatch
   ```

Atualizar a imagem do contêiner do atendente do CloudWatch

> **Important**
>
> Se você estiver atualizando ou instalando o Container Insights em um cluster do Amazon EKS, recomendamos usar o complemento Observability do Amazon CloudWatch para o EKS para a instalação, em vez de usar as instruções apresentadas nesta seção. Além disso, para recuperar métricas de computação acelerada, é necessário usar o complemento Observability do Amazon CloudWatch para o EKS ou o operador do agente do CloudWatch. Para mais informações e instruções, consulte Instalar o complemento Amazon CloudWatch Observability do EKS.

Se você precisar atualizar a imagem do contêiner para a versão mais recente, use as etapas nesta seção.

Para atualizar a imagem de contêiner

1. Verifique se a Definição de Recursos do Cliente (CRD) `amazoncloudwatchagent` já existe ao inserir o comando apresentado a seguir.

   ```
   kubectl get crds amazoncloudwatchagents.cloudwatch.aws.amazon.com -n amazon-
   cloudwatch
   ```

   Se esse comando retornar um erro informando que o CRD está ausente, o cluster não tem o Container Insights com observabilidade aprimorada para o Amazon EKS configurado com o operador do agente do CloudWatch. Nesse caso, consulte Como atualizar para o Container Insights com observabilidade aprimorada para o Amazon EKS.
2. Aplique o arquivo `cwagent-version.yaml` mais recente inserindo o comando a seguir.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/main/k8s-quickstart/cwagent-version.yaml | kubectl apply -f -
   ```

Exclusão do agente do CloudWatch e do Fluent Bit para o Container Insights

Se você instalou o Container Insights usando a instalação do complemento de observabilidade do EKS para o Amazon CloudWatch, você pode excluir o Container Insights e o agente do CloudWatch inserindo o seguinte comando:

> **Note**
>
> Agora, o complemento do Amazon EKS é compatível com o Container Insights em nós de processamento do Windows. Se você excluir o complemento do Amazon EKS, o Container Insights para Windows também será excluído.

```
aws eks delete-addon –cluster-name my-cluster –addon-name amazon-cloudwatch-
observability
```

Caso contrário, para excluir todos os recursos relacionados ao agente do CloudWatch e ao Fluent Bit, insira o comando apresentado a seguir. Neste comando, *`My_Cluster_Name`* corresponde ao nome do cluster do Amazon EKS ou do Kubernetes, e *`My_Region`* corresponde ao nome da região na qual os logs são publicados.

```
ClusterName=My_Cluster_Name
RegionName=My-Region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 delete -f -
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl delete
 -f -
```

Visualizar métricas do Container Insights

Depois que o Container Insights for configurado e estiver coletando métricas, você poderá visualizar essas métricas no console do CloudWatch.

Para que as métricas do Container Insights sejam exibidas no seu painel, você deve concluir a configuração do Container Insights. Para ter mais informações, consulte Configurar o Container Insights.

Esse procedimento explica como visualizar as métricas que o Container Insights gera automaticamente a partir dos dados de log coletados. O restante desta seção explica como analisar seus dados mais profundamente e usar o CloudWatch Logs Insights para ver mais métricas em mais níveis de detalhamento.

Para visualizar métricas do Container Insights

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights, Container Insights.
3. Na caixa suspensa sob Container Insights, escolha Monitoramento de performance.
4. Use as caixas suspensas perto do canto superior para selecionar o tipo de recurso a ser exibido, bem como o recurso específico.

Você pode definir um alarme do CloudWatch em qualquer métrica que o Container Insights coletar. Para mais informações, consulte Usar alarmes do Amazon CloudWatch

> **Note**
>
> Se você já configurou o CloudWatch Application Insights para monitorar suas aplicações em contêiner, o painel do Application Insights aparecerá abaixo do painel do Container

Insights. Se você ainda não ativou o Application Insights, poderá fazê-lo escolhendo Auto-configure Application Insights (Configurar automaticamente o Application Insights abaixo da visualização de performance no painel Container Insights.
Para obter mais informações sobre o Application Insights e aplicações em contêiner, consulte Habilitar o Application Insights para monitoramento de recursos do Amazon ECS e do Amazon EKS.

Visualizar os principais colaboradores

Para algumas das exibições no monitoramento de performance do Container Insights, também é possível ver os principais colaboradores por memória ou CPU, ou os recursos ativos mais recentes. Essa informação está disponível quando você seleciona, na caixa suspensa próxima ao alto da página, um dos seguintes painéis:

- Serviços do ECS
- Tarefas do ECS
- Namespaces do EKS
- Serviços do EKS
- Pods do EKS

Quando você estiver visualizando um desses tipos de recursos, a parte inferior da página exibirá uma tabela classificada inicialmente por uso da CPU. É possível alterá-la para classificar por uso de memória ou atividade recente. Para ver mais sobre uma das linhas na tabela, é possível marcar a caixa de seleção ao lado dessa linha e escolher Actions (Ações) e uma das opções do menu Actions.

Usar o CloudWatch Logs Insights para visualizar dados do Container Insights

O Container Insights coleta métricas com eventos de log de performance usando formato de métrica incorporado. Os logs são armazenados no CloudWatch Logs. O CloudWatch gera várias métricas automaticamente a partir dos logs. Você pode visualizá-las no console do CloudWatch. Também é possível fazer uma análise mais profunda dos dados de performance coletados usando consultas do CloudWatch Logs Insights.

Para obter mais informações sobre o CloudWatch Logs Insights, consulte Analisar dados de log com o CloudWatch Logs Insights. Para obter mais informações sobre os campos de log que podem ser

usados em consultas, consulte Eventos do log de performance do Container Insights para Amazon EKS e Kubernetes.

Para usar o CloudWatch Logs Insights para consultar os dados de métricas de contêiner

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Insights.

   Próximo da parte superior da tela está o editor de consultas. Quando você abre o CloudWatch Logs Insights pela primeira vez, essa caixa contém uma consulta padrão que retorna os 20 eventos de log mais recentes.
3. Na caixa acima do editor de consultas, selecione um dos grupos de log do Container Insights para consultar. Para o que o exemplo de consultas a seguir funcione, o nome do grupo de logs deve terminar com performance.

   Quando você seleciona um grupo de logs, o CloudWatch Logs Insights automaticamente detecta campos nos dados no grupo de logs e os exibe em Discovered fields (Campos detectados) no painel á direta. Ele também exibe um gráfico de barras de eventos de log neste grupo de logs com o passar do tempo. Esse gráfico de barras mostra a distribuição de eventos no grupo de logs correspondente à consulta e ao intervalo de tempo, e não apenas os eventos exibidos na tabela.
4. No editor de consultas, substitua a consulta padrão pela consulta a seguir e selecione Run query (Executar consulta).

   ```
   STATS avg(node_cpu_utilization) as avg_node_cpu_utilization by NodeName
   | SORT avg_node_cpu_utilization DESC
   ```

   Essa consulta mostra uma lista de nós, classificados por utilização média da CPU do nó.
5. Para tentar outro exemplo, substitua essa consulta por outra consulta e selecione Run query (Executar consulta). Mais consultas de exemplo são listadas posteriormente nesta página.

   ```
   STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by
    PodName
   | SORT avg_number_of_container_restarts DESC
   ```

   Essa consulta exibe uma lista dos pods classificados pelo número médio de reinicializações do contêiner.

6. Se você quiser tentar outra consulta, poderá usar campos de inclusão na lista à direita da tela. Para obter mais informações sobre a sintaxe de consulta, leia Sintaxe de consulta do CloudWatch Logs Insights.

Como visualizar as listas de seus recursos

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Resources (Recursos).
3. A exibição padrão é uma lista dos recursos que estão sendo monitorados pelo Container Insights e os alarmes definidos nesses recursos. Para visualizar um mapa visual dos recursos, escolha Map view (Visualização do mapa).
4. Na visualização do mapa, deixe o cursor sobre qualquer recurso no mapa para visualizar métricas básicas sobre esse recurso. Você pode escolher qualquer recurso para visualizar gráficos mais detalhados sobre o recurso.

Caso de uso: visualizar métricas em nível de tarefa em contêineres do Amazon ECS

O exemplo a seguir ilustra como usar o CloudWatch Logs Insights para aprofundar os logs do Container Insights. Para obter mais exemplos, consulte o blog Introducing Amazon CloudWatch Container Insights for Amazon ECS.

O Container Insights não gera automaticamente métricas no nível de detalhamento da tarefa. A consulta a seguir exibe métricas no nível da tarefa para uso da CPU e da memória.

```
stats avg(CpuUtilized) as CPU, avg(MemoryUtilized) as Mem by TaskId, ContainerName
| sort Mem, CPU desc
```

Outros exemplos de consultas do Container Insights

Lista de seus pods, classificados por número médio de reinicializações de contêiner

```
STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by PodName
| SORT avg_number_of_container_restarts DESC
```

Pods solicitados versus pods em execução

```
fields @timestamp, @message
| sort @timestamp desc
```

```
| filter Type="Pod"
| stats min(pod_number_of_containers) as requested,
 min(pod_number_of_running_containers) as running, ceil(avg(pod_number_of_containers-
pod_number_of_running_containers)) as pods_missing by kubernetes.pod_name
| sort pods_missing desc
```

Contagem de falhas no nó do cluster

```
stats avg(cluster_failed_node_count) as CountOfNodeFailures
| filter Type="Cluster"
| sort @timestamp desc
```

Erros de log do aplicativo por nome do contêiner

```
stats count() as countoferrors by kubernetes.container_name
| filter stream="stderr"
| sort countoferrors desc
```

Métricas coletadas pelo Container Insights

O Container Insights coleta um conjunto de métricas para o Amazon ECS e para o AWS Fargate no Amazon ECS, e um outro conjunto para o Amazon EKS, para o AWS Fargate no Amazon EKS e para o Kubernetes.

As métricas não estarão visíveis até que as tarefas do contêiner estejam em execução por algum tempo.

Tópicos

- Métricas Amazon ECS Container Insights
- Métricas do Amazon EKS e do Kubernetes Container Insights

Métricas Amazon ECS Container Insights

A tabela a seguir lista as métricas e as dimensões que o Container Insights coleta para Amazon ECS. Essas métricas estão no namespace `ECS/ContainerInsights`. Para ter mais informações, consulte Indicadores.

Se você não vir as métricas do Container Insights no seu console, certifique-se de que você tenha concluído a configuração do Container Insights. As métricas não serão exibidas até que o Container

Insights tenha sido configurado completamente. Para ter mais informações, consulte Configurar o Container Insights.

As métricas a seguir estarão disponíveis após a conclusão das etapas em Configurar o Container Insights no Amazon ECS para métricas no nível de cluster e no nível de serviço

Nome da métrica	Dimensões	Descrição
`ContainerInstanceCount`	ClusterName	O número de instâncias do EC2 que executam o atendente do Amazon ECS registrado com um cluster. Essa métrica é coletada apenas para instâncias de contêineres que estão executando tarefas do Amazon ECS no cluster. Ela não é coletada para instâncias de contêineres vazios que não têm nenhuma tarefa do Amazon ECS. Unidade: Contagem

`CpuUtilized`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		As unidades de CPU utilizadas por tarefas no recurso especificado pela definição de dimensão que você está usando.
		
		Essa métrica é coletada apenas para tarefas que têm uma reserva de CPU definida em sua definição de tarefa.

Nome da métrica	Dimensões	Descrição
		Unidade: nenhuma

`CpuReserved`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		As unidades de CPU reservadas por tarefas no recurso especificado pelo conjunto de dimensões que você está usando.
		
		Essa métrica é coletada apenas para tarefas que têm uma reserva de CPU definida em sua definição de tarefa.
		
		Unidade: nenhuma

`DeploymentCount`	`ServiceName` , `ClusterName`	O número de implantações em um serviço do Amazon ECS. Unidade: Contagem
`DesiredTaskCount`	`ServiceName` , `ClusterName`	O número de tarefas desejadas para um serviço do Amazon ECS. Unidade: Contagem

Nome da métrica	Dimensões	Descrição

`EBSFilesystemSize`	`VolumeName` , `TaskDefinitionFamily` , `ClusterName`
	
	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
		A quantidade total, em gigabytes (GB), do armazenamento do sistema de arquivos do Amazon EBS alocada aos recursos especificados pelas dimensões que você está usando.
		
		Esta métrica só está disponível para tarefas executadas na infraestrutura do Amazon ECS executadas no Fargate usando a versão da plataforma `1.4.0` ou instâncias do Amazon EC2 usando a versão do agente de contêiner `1.79.0` ou posterior.
		
		Unidade: gigabytes (GB)

Nome da métrica	Dimensões	Descrição

`EBSFilesystemUtilized`	`VolumeName` , `TaskDefinitionFamily` , `ClusterName`
	
	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
		A quantidade total, em gigabytes (GB), do armazenamento do sistema de arquivos do Amazon EBS usada pelos recursos especificados pelas dimensões que você está usando.
		
		Esta métrica só está disponível para tarefas executadas na infraestrutura do Amazon ECS executadas no Fargate usando a versão da plataforma `1.4.0` ou instâncias do Amazon EC2 usando a versão do agente de contêiner `1.79.0` ou posterior.
		
		Para tarefas executadas no Fargate, o Fargate reserva espaço no disco que é usado somente pelo Fargate. Não há custo associado ao espaço que Fargate usa, mas você verá esse armazenamento adicional usando ferramentas como `df`.
		
		Unidade: gigabytes (GB)

Nome da métrica	Dimensões	Descrição

`EphemeralStorageReserved` [1]	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		O número de bytes reservados no armazenamento efêmero do recurso, especificado pelas dimensões que você está usando. O armazenamento efêmero é usado para o sistema de arquivos raiz do contêiner e para qualquer volume de host de montagem por associação definido na imagem do contêiner e na definição da tarefa. A quantidade de armazenamento efêmero não pode ser alterada em uma tarefa em execução.
		
		Essa métrica só está disponível para tarefas executadas na plataforma Fargate Linux versão 1.4.0 ou posterior.
		
		Unidade: gigabytes (GB)

Nome da métrica	Dimensões	Descrição

`EphemeralStorageUtilized` [1]	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		O número de bytes usados no armazenamento efêmero do recurso, especificado pelas dimensões que você está usando. O armazenamento efêmero é usado para o sistema de arquivos raiz do contêiner e para qualquer volume de host de montagem por associação definido na imagem do contêiner e na definição da tarefa. A quantidade de armazenamento efêmero não pode ser alterada em uma tarefa em execução.
		
		Essa métrica só está disponível para tarefas executadas na plataforma Fargate Linux versão 1.4.0 ou posterior.
		
		Unidade: gigabytes (GB)

Nome da métrica	Dimensões	Descrição
MemoryUtilized	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	A memória que está sendo usada pelas tarefas no recurso especificado pela definição de dimensões que você está usando. Essa métrica é coletada apenas para tarefas que têm uma reserva de memória definida em sua definição de tarefa. Unidade: megabytes
MemoryReserved	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	A memória reservada por tarefas no recurso especificado pelo conjunto de dimensões que você está usando. Essa métrica é coletada apenas para tarefas que têm uma reserva de memória definida em sua definição de tarefa. Unidade: megabytes

Nome da métrica	Dimensões	Descrição
NetworkRxBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	O número de bytes recebidos pelo recurso especificado por dimensões que você está usando. Essa métrica é obtida com o runtime do Docker. Essa métrica está disponível apenas para contêineres em tarefas que usam os modos de rede awsvpc ou bridge. Unidade: bytes/segundo
NetworkTxBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	O número de bytes enviados pelo recurso especificado pelas dimensões que você está usando. Essa métrica é obtida com o runtime do Docker. Essa métrica está disponível apenas para contêineres em tarefas que usam os modos de rede awsvpc ou bridge. Unidade: bytes/segundo

Nome da métrica	Dimensões	Descrição
PendingTaskCount	ServiceName , ClusterName	O número de tarefas que estão atualmente no estado PENDING. Unidade: Contagem
RunningTaskCount	ServiceName , ClusterName	O número de tarefas que estão atualmente no estado RUNNING. Unidade: Contagem
ServiceCount	ClusterName	O número de serviços no cluster. Unidade: Contagem
StorageReadBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	O número de bytes lidos do armazenamento na instância do recurso que é especificado pelas dimensões que você está usando. Isso não inclui bytes de leitura para seus dispositivos de armazenamento. Essa métrica é obtida com o runtime do Docker. Unidade: bytes

Nome da métrica	Dimensões	Descrição

`StorageWriteBytes`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		O número de bytes gravados para armazenamento no recurso especificado pelas dimensões que você está usando. Essa métrica é obtida com o runtime do Docker.
		
		Unidade: bytes

`TaskCount`	`ClusterName`	O número de tarefas em execução no cluster. Unidade: Contagem
`TaskSetCount`	`ServiceName` , `ClusterName`	O número de conjuntos de tarefas no serviço. Unidade: Contagem

> **Note**
>
> As métricas `EphemeralStorageReserved` e `EphemeralStorageUtilized` só estão disponíveis para tarefas que são executadas na plataforma Fargate Linux versão 1.4.0 ou posterior.
> O Fargate reserva espaço no disco. Esse espaço é usado apenas pelo Fargate. Você não é cobrado por isso. Ele não é mostrado nessas métricas. Porém, você pode ver esse armazenamento adicional em outras ferramentas, como o `df`.

As métricas a seguir estarão disponíveis após a conclusão das etapas em Implantar o atendente do CloudWatch para coletar métricas no nível de instância do EC2 no Amazon ECS

Nome da métrica	Dimensões	Descrição
instance_cpu_limit	ClusterName	O número máximo de unidades de CPU que podem ser atribuídas a uma única instância do EC2 no cluster. Unidade: nenhuma
instance_cpu_reserved_capacity	ClusterName InstanceId, ContainerInstanceId, ClusterName	A porcentagem de CPU que está sendo reservada em uma única instância do EC2 no cluster. Unidade: Percentual
instance_cpu_usage_total	ClusterName	O número de unidades de CPU que está sendo usada em uma única instância do EC2 no cluster. Unidade: nenhuma
instance_cpu_utilization	ClusterName InstanceId, ContainerInstanceId, ClusterName	A porcentagem total de unidades de CPU que estão sendo usadas em uma única instância do EC2 no cluster. Unidade: Percentual
instance_filesystem_utilization	ClusterName InstanceId, ContainerInstanceId, ClusterName	A porcentagem total da capacidade do sistema de arquivos de que está sendo

Nome da métrica	Dimensões	Descrição
		usada em uma única instância do EC2 no cluster. Unidade: Percentual
`instance_memory_limit`	`ClusterName`	A quantidade máxima de memória, em bytes, que pode ser atribuída a uma única instância do EC2 nesse cluster. Unidade: bytes

`instance_memory_reserved_capacity`	`ClusterName`
	
	`InstanceId`, `ContainerInstanceId`, `ClusterName`
		A porcentagem de memória que está sendo reservada em uma única instância do EC2 no cluster.
		
		Unidade: Percentual

`instance_memory_utilization`	`ClusterName`
	
	`InstanceId`, `ContainerInstanceId`, `ClusterName`
		A porcentagem total de memória que está sendo usada em uma única instância do EC2 no cluster.
		
		Unidade: Percentual

Nome da métrica	Dimensões	Descrição
`instance_memory_working_set`	`ClusterName`	A quantidade de memória, em bytes, que está sendo usada em uma única instância do EC2 no cluster. Unidade: bytes
`instance_network_total_bytes`	`ClusterName`	O número total de bytes por segundo transmitidos e recebidos pela rede um uma única instância do EC2 no cluster. Unidade: bytes/segundo
`instance_number_of_running_tasks`	`ClusterName`	O número de tarefas em execução em uma única instância do EC2 no cluster. Unidade: Contagem

Métricas do Amazon EKS e do Kubernetes Container Insights

As tabelas a seguir listam as métricas e as dimensões que o Container Insights coleta para o Amazon EKS e Kubernetes. Essas métricas estão no namespace `ContainerInsights`. Para ter mais informações, consulte Indicadores.

Se você não vir as métricas do Container Insights no seu console, certifique-se de que você tenha concluído a configuração do Container Insights. As métricas não serão exibidas até que o Container

Insights tenha sido configurado completamente. Para ter mais informações, consulte Configurar o Container Insights.

Se você estiver usando a versão 1.5.0 ou versões posteriores do complemento do Amazon EKS ou a versão 1.300035.0 do agente do CloudWatch, a maioria das métricas listadas na tabela a seguir será coletada para nós nos sistemas Linux e Windows. Consulte a coluna Nome da métrica da tabela para visualizar quais métricas não são coletadas para o Windows.

Com a versão original do Container Insights, as métricas são cobradas como métricas personalizadas. Com o Container Insights com capacidade de observabilidade aprimorada para o Amazon EKS, as métricas do Container Insights são cobradas por observação, em vez de serem cobradas por métrica armazenada ou log ingerido. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

> **Note**
>
> No Windows, métricas de rede, como `pod_network_rx_bytes` e `pod_network_tx_bytes`, não são coletadas para a hospedagem de contêineres de processos.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
`cluster_failed_node_count`	`ClusterName`		O número de nós do operador com falha no cluster. Um nó é considerado com falha quando apresenta qualquer condição de nó. Para obter mais informações, consulte Condições na

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			documentação do Kubernetes.
`cluster_node_count`	`ClusterName`		O número total de nós do operador no cluster.

`namespace_number_of_running_pods`	`Namespace` `ClusterName`
	
	`ClusterName`
			O número de pods em execução por namespace no recurso especificado pelas dimensões que você está usando.

`node_cpu_limit`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	O número máximo de unidades de CPU que pode ser atribuído a um único nó neste cluster.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_cpu_reserved_capacity`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			A porcentagem de unidades de CPU reservadas para componentes do nó, como kubelet, kube-proxy e Docker.
			
			Fórmula: `node_cpu_request / node_cpu_limit`
			
			Note
			`node_cpu_request` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes nos eventos de log de

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			performance para Amazon EKS e Kubernetes.
node_cpu_usage_total	ClusterName	ClusterName , InstanceId , NodeName	O número de unidades da CPU que está sendo usado nos nós do cluster.
node_cpu_utilization	NodeName, ClusterName , InstanceId ClusterName		A porcentagem total de unidades de CPU que está sendo usada nos nós do cluster. Fórmula: node_cpu_usage_total / node_cpu_limit

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_filesystem_utilization`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			A porcentagem total da capacidade do sistema de arquivos que está sendo usado nos nós do cluster.
			
			Fórmula: `node_filesystem_usage / node_filesystem_capacity`
			
			Note
			`node_filesystem_usage` e `node_filesystem_capacity` não são relatados diretamente como métricas, mas são campos em eventos de log de performance. Para ter mais informaçõ

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			es, consulte Campos relevantes nos eventos de log de performance para Amazon EKS e Kubernetes.
`node_memory_limit`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	A quantidade máxima de memória, em bytes, que pode ser atribuída a um único nó neste cluster.

`node_filesystem_inodes`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS. Não está disponível no Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			O número total de inodes (usados e não usados) em um nó.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_filesystem_inodes_free`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS. Não está disponível no Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			O número de inodes não utilizados em um nó.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_memory_reserved_capacity`	`NodeName`, `ClusterName`, `InstanceId`
	
	`ClusterName`
			A porcentagem de memória que está sendo usada no momento nos nós do cluster.
			
			Fórmula: `node_memory_request / node_memory_limit`
			
			Note
			`node_memory_request` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes nos eventos

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			de log de performance para Amazon EKS e Kubernetes.
node_memory_utilization	NodeName, ClusterName , InstanceId ClusterName		A porcentagem de memória que está sendo usada no momento pelo nó ou nós. É o percentual de uso de memória de nó dividido pela limitação de memória de nó. Fórmula: node_memory_working_set / node_memory_limit .
node_memory_working_set	ClusterName	ClusterName , InstanceId , NodeName	A quantidade de memória, em bytes, sendo usada no conjunto de trabalho dos nós no cluster.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_network_total_bytes`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			O número total de bytes transmitidos e recebidos por segundo pela rede por nó em um cluster.
			
			Fórmula: `node_network_rx_bytes + node_network_tx_bytes`
			
			Note
			`node_network_rx_bytes` e `node_network_tx_bytes` não são relatados diretamente como métricas, mas são campos em eventos de log de performance. Para ter mais informaçõ es, consulte

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			Campos relevantes nos eventos de log de performance para Amazon EKS e Kubernetes.

`node_number_of_running_containers`	`NodeName`, `ClusterName`, `InstanceId`
	
	`ClusterName`
			O número de contêineres em execução por nó em um cluster.

`node_number_of_running_pods`	`NodeName`, `ClusterName`, `InstanceId`
	
	`ClusterName`
			O número de pods em execução por nó em um cluster.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_status_allocatable_pods`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			O número de pods que podem ser atribuídos a um nó com base em seus recursos alocáveis, que é definido como o restante da capacidade de um nó depois de contabilizar as reservas de daemons do sistema e os limites de remoção rígidos.

`node_status_capacity_pods`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			O número de pods que podem ser atribuídos a um nó com base em sua capacidade.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_status_condition_ready`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indica se a condição de status do nó `Ready` é verdadeira.

`node_status_condition_memory_pressure`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indica se a condição de status do nó `MemoryPressure` é verdadeira.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_status_condition_pid_pressure`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indica se a condição de status do nó `PIDPressure` é verdadeira.

`node_status_condition_disk_pressure`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indica se a condição de status do nó `OutOfDisk` é verdadeira.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_status_condition_unknown`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Indica se alguma das condições de status do nó é Desconhecida.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_interface_network_rx_dropped`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			O número de pacotes que foram recebidos e posteriormente descartados por uma interface de rede no nó.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_interface_network_tx_dropped`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			O número de pacotes que deveriam ser transmitidos, mas foram descartados por uma interface de rede no nó.

`node_diskio_io_service_bytes_total`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS. Não está disponível no Windows.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			O número total de bytes transferidos por todas as operações de E/S no nó.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`node_disk io_io_ser viced_total`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS. Não está disponível no Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			O número total de operações de E/S no nó.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_cpu_reserved_capacity`	`PodName`, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
		
		`ClusterName` , `Namespace` , `Service`
			A capacidade da CPU reservada por pod em um cluster.
			
			Fórmula: `pod_cpu_request / node_cpu_limit`
			
			Note
			`pod_cpu_request` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes nos eventos de log de performance para

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_cpu_utilization`	`PodName`, Namespace, `ClusterName`
	
	Namespace, `ClusterName`
	
	Serviço, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	A porcentagem de unidades de CPU que estão sendo usadas por pods.
			
			Fórmula: `pod_cpu_usage_total / node_cpu_limit`
			
			Note
			`pod_cpu_usage_total` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes nos eventos de log de performan

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			ce para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_cpu_utilization_over_pod_limit`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	Serviço, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	O percentual das unidades de CPU que estão sendo usadas por pods com relação ao limite de pods.
			
			Fórmula: `pod_cpu_usage_total / pod_cpu_limit`
			
			Note
			`pod_cpu_usage_total` e `pod_cpu_limit` não são relatados diretamente como métricas, mas são campos em eventos de log de performance. Para ter mais informações, consulte Campos relevantes

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			nos eventos de log de performance para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_memory_reserved_capacity`	`PodName`, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
		
		`ClusterName` , `Namespace` , `Service`
			A porcentagem de memória reservada para pods.
			
			Fórmula: `pod_memory_request / node_memory_limit`
			
			Note
			`pod_memory_request` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes nos eventos de log de performan

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			ce para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_memory_utilization`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	Serviço, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName`, `Namespace`, `PodName`, `FullPodName`	A porcentagem de memória que está sendo usada atualmente pelo pod ou pods.
			
			Fórmula: `pod_memory_working_set / node_memory_limit`
			
			Note
			`pod_memory_working_set` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			nos eventos de log de performance para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_memory_utilization_over_pod_limit`	`PodName`, Namespace, `ClusterName`
	
	Namespace, `ClusterName`
	
	Serviço, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	O percentual de memória que está sendo usada por pods com relação ao limite de pods. Se qualquer contêiner no pod não tiver um limite de memória definido, essa métrica não aparecerá.
			
			Fórmula: `pod_memory_working_set / pod_memory_limit`
			
			Note
			`pod_memory_working_set` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			Para ter mais informações, consulte Campos relevantes nos eventos de log de performance para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_network_rx_bytes`	`PodName`, Namespace, `ClusterName`
	
	Namespace, `ClusterName`
	
	Serviço, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	O número de bytes que estão sendo recebidos por segundo na rede pelo pod.
			
			Fórmula: `sum(pod_interface_network_rx_bytes)`
			
			Note
			`pod_interface_network_rx_bytes` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			nos eventos de log de performance para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_network_tx_bytes`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	Serviço, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	O número de bytes que estão sendo transmitidos por segundo na rede pelo pod.
			
			Fórmula: `sum(pod_interface_network_tx_bytes)`
			
			Note
			`pod_interface_network_tx_bytes` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			nos eventos de log de performance para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_cpu_request`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			As solicitações da CPU para o pod.
			
			Fórmula: `sum(container_cpu_request)`
			
			Note
			`pod_cpu_request` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes nos eventos de log de performance para

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_memory_request`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			As solicitações de memória para o pod.
			
			Fórmula: `sum(container_memory_request)`
			
			Note
			`pod_memory_request` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes nos eventos de log de performance para

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_cpu_limit`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			O limite de CPU definido para os contêineres no pod. Se algum contêiner no pod não tiver um limite de CPU definido, essa métrica não será exibida.
			
			Fórmula: `sum(container_cpu_limit)`
			
			Note
			`pod_cpu_limit` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			nos eventos de log de performance para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_memory_limit`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			O limite de memória definido para os contêineres no pod. Se qualquer contêiner no pod não tiver um limite de memória definido, essa métrica não aparecerá.
			
			Fórmula: `sum(container_memory_limit)`
			
			Note
			`pod_cpu_limit` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			nos eventos de log de performance para Amazon EKS e Kubernetes.

`pod_status_failed`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que todos os contêineres no pod foram encerrados, e pelo menos um contêiner foi encerrado com um status diferente de zero ou foi encerrado pelo sistema.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_status_ready`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que todos os contêineres no pod estão prontos, tendo atingido a condição `ContainerReady` .

`pod_status_running`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que todos os contêineres no pod estão em execução.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_status_scheduled`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que o pod foi agendado para um nó.

`pod_status_unknown`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que o status do pod não pode ser obtido.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_status_pending`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que o pod foi aceito pelo cluster, mas um ou mais contêineres ainda não estão prontos.

`pod_status_succeeded`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica que todos os contêineres no pod foram encerrados com êxito e não serão reiniciados.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_number_of_containers`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Relata o número de contêineres definidos na especificação do pod.

`pod_number_of_running_containers`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Relata o número de contêineres no pod que estão atualmente no estado `Running`.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_container_status_terminated`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Relata o número de contêineres no pod que estão no estado `Terminated` .

`pod_container_status_running`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Relata o número de contêineres no pod que estão no estado `Running`.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_container_status_waiting`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Relata o número de contêineres no pod que estão no estado `Waiting`.

`pod_interface_network_rx_dropped`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			O número de pacotes que foram recebidos e posteriormente descartados em uma interface de rede para o pod.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`pod_interface_network_tx_dropped`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			O número de pacotes que deveriam ser transmitidos, mas foram descartados para o pod.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`container_cpu_utilization`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			A porcentagem de unidades de CPU que estão sendo usadas pelo contêiner.
			
			Fórmula: `container_cpu_usage_total / node_cpu_limit`
			
			Note
			`container_cpu_utilization` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes nos eventos de log de

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			performance para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`container_cpu_utilization_over_container_limit`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			A porcentagem de unidades de CPU que estão sendo usadas pelo contêiner em relação ao limite do contêiner. Se o contêiner não tiver um limite de CPU definido, essa métrica não será exibida.
			
			Fórmula: `container_cpu_usage_total / container_cpu_limit`
			
			Note
			`container_cpu_utilization_over_container_limit` não é relatado diretamente como uma métrica, mas é um campo

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			em eventos de log de performance. Para ter mais informaçõ es, consulte Campos relevantes nos eventos de log de performan ce para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`container_memory_utilization`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			A porcentagem de unidades de memória que está sendo usada pelo contêiner.
			
			Fórmula: `container_memory_working_set / node_memory_limit`
			
			Note
			`container_memory_utilization` não é relatado diretamente como uma métrica, mas é um campo em eventos de log de performance. Para ter mais informações, consulte Campos relevantes

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			nos eventos de log de performance para Amazon EKS e Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`container_memory_utilization_over_container_limit`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			A porcentagem de unidades de memória que está sendo usada pelo contêiner em relação ao limite do contêiner. Se o contêiner não tiver um limite de memória definido, essa métrica não será exibida.
			
			Fórmula: `container_memory_working_set / container_memory_limit`
			
			Note
			`container_memory_utilization_over_container_limit` não é relatado diretamente como uma métrica, mas é um campo

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
			em eventos de log de performance. Para ter mais informações, consulte Campos relevantes nos eventos de log de performance para Amazon EKS e Kubernetes.

`container_memory_failures_total`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS. Não está disponível no Windows.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			O número de falhas de alocação de memória que ocorreram no contêiner.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição
pod_number_of_container_restarts	PodName, Namespace , ClusterName		O número total de reinicializações de contêineres em um pod.
service_number_of_running_pods	Serviço, Namespace , ClusterName ClusterName		O número de pods que executam o serviço ou os serviços no cluster.
replicas_desired Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS		ClusterName PodName, Namespace , ClusterName	O número de pods desejados para uma workload, conforme definido na especificação da workload.
replicas_ready Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS		ClusterName PodName, Namespace , ClusterName	O número de pods de uma workload que atingiram o status de “prontos”.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`status_replicas_available`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			O número de pods para uma workload que estão disponíveis. Um pod está disponível quando estiver pronto para o `minReadySeconds` definido na especificação da workload.

`status_replicas_unavailable`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			O número de pods para uma workload que não estão disponíveis. Um pod está disponível quando estiver pronto para o `minReadySeconds` definido na especificação da workload. Os pods não estarão disponíveis se não atenderem a esse critério.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`apiserver_storage_objects`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName`, `resource`
			O número de objetos armazenados no etcd no momento da última verificação.

`apiserver_request_total`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName`, `code`, `verb`
			O número total de solicitações de API para o servidor de API do Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`apiserver_request_duration_seconds`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , verb`
			Latência de resposta para solicitações de API para o servidor de API do Kubernetes.

`apiserver_admission_controller_admission_duration_seconds`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , operation`
			Latência do controlador de admissão em segundos. Um controlador de admissão é um código que intercepta solicitações para o servidor de API do Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`rest_client_request_duration_seconds`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName`, `operation`
			Latência de resposta observada pelos clientes que chamam o servidor da API do Kubernetes. Essa métrica é experimental e pode mudar em versões futuras do Kubernetes.

`rest_client_requests_total`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName`, `code`, `method`
			O número total de solicitações de API para o servidor de API do Kubernetes feitas por clientes. Essa métrica é experimental e pode mudar em versões futuras do Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`etcd_request_duration_seconds`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , operation`
			Latência de resposta das chamadas de API para o Etcd. Essa métrica é experimental e pode mudar em versões futuras do Kubernetes.

`apiserver_storage_size_bytes`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , endpoint`
			Tamanho do arquivo de banco de dados de armazenamento alocado fisicamente em bytes. Essa métrica é experimental e pode mudar em versões futuras do Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`apiserver_longrunning_requests`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , resource`
			O número de solicitações ativas de longa duração para o servidor de API do Kubernetes.

`apiserver_current_inflight_requests`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , request_kind`
			O número de solicitações que estão sendo processadas pelo servidor de API do Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`apiserver_admission_webhook_admission_duration_seconds`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , name`
			Latência do webhook de admissão em segundos. Os webhooks de admissão são retornos de chamada HTTP que recebem solicitações de admissão e realizam alguma ação com elas.

`apiserver_admission_step_admission_duration_seconds`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , operation`
			Latência da subetapa de admissão em segundos.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`apiserver_requested_deprecated_apis`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , group`
			Número de solicitações para APIs obsoletas no servidor de API do Kubernetes.

`apiserver_request_total_5XX`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , code, verb`
			Número de solicitações ao servidor de API do Kubernetes que foram respondidas com um código de resposta HTTP 5XX.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`apiserver_storage_list_duration_seconds`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , resource`
			Latência de resposta da listagem de objetos do Etcd. Essa métrica é experimental e pode mudar em versões futuras do Kubernetes.

`apiserver_current_inqueue_requests`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , request_kind`
			O número de solicitações em fila enfileiradas pelo servidor de API do Kubernetes. Essa métrica é experimental e pode mudar em versões futuras do Kubernetes.

Nome da métrica	Dimensões com qualquer versão do Container Insights	Dimensões adicionais com o Container Insights com observabilidade aprimorada para o Amazon EKS	Descrição

`apiserver_flowcontrol_rejected_requests_total`
 	
Essa métrica está disponível somente com o Container Insights com observabilidade aprimorada para o Amazon EKS
		`ClusterName`
		
		`ClusterName , reason`
			Número de solicitações rejeitadas pelo subsistema API Priority and Fairness. Essa métrica é experimental e pode mudar em versões futuras do Kubernetes.

Métricas da GPU NVIDIA

A partir da versão `1.300034.0` do agente do CloudWatch, o Container Insights com observabilidade aprimorada para o Amazon EKS coleta métricas da GPU NVIDIA de workloads do EKS por padrão. O agente do CloudWatch deve ser instalado usando o complemento Observability do CloudWatch para o EKS na versão `v1.3.0-eksbuild.1` ou em versões posteriores. Para ter mais informações, consulte Instalar o agente do CloudWatch usando o complemento de observabilidade do EKS do Amazon CloudWatch. Essas métricas de GPU NVIDIA coletadas estão listadas na tabela desta seção.

Para que o Container Insights colete métricas de GPU NVIDIA, você deve atender aos seguintes pré-requisitos:

- Você deve usar o Container Insights com observabilidade aprimorada para o Amazon EKS, com o complemento Observability do Amazon CloudWatch para o EKS na versão `v1.3.0-eksbuild.1` ou em versões posteriores.

- O plug-in de dispositivo NVIDIA para Kubernetes deve estar instalado no cluster.
- O kit de ferramentas de contêiner NVIDIA deve ser instalado nos nós do cluster. Por exemplo, as AMIs aceleradas otimizadas do Amazon EKS são criadas com os componentes necessários.

Você pode optar por não coletar métricas de GPU NVIDIA definindo a opção `accelerated_compute_metrics` no arquivo de configuração do agente CloudWatch como `false`. Para obter mais informações e um exemplo de configuração de exclusão, consulte (Opcional) Configuração adicional.

Nome da métrica	Dimensões	Descrição

`container_gpu_memory_total`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		O tamanho total do buffer de quadros, em bytes, nas GPUs alocadas ao contêiner.

`container_gpu_memory_used`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		O bytes do buffer de quadros usados nas GPUs alocadas ao contêiner.

Nome da métrica	Dimensões	Descrição
container_gpu_memory_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	A porcentagem do buffer de quadros usada das GPUs alocadas ao contêiner.
container_gpu_power_draw	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	O uso de energia em watts das GPUs alocadas ao contêiner.

Nome da métrica	Dimensões	Descrição
container_gpu_temperature	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	A temperatura em graus Celsius das GPUs alocadas ao contêiner.
container_gpu_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	A porcentagem de utilização das GPUs alocadas ao contêiner.
node_gpu_memory_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	O tamanho total do buffer de quadros, em bytes, nas GPUs alocadas ao nó.

Nome da métrica	Dimensões	Descrição
node_gpu_memory_used	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Os bytes do buffer de quadros usados nas GPUs alocadas ao nó.
node_gpu_memory_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	A porcentagem de buffer de quadros usado nas GPUs alocadas ao nó.
node_gpu_power_draw	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	O uso de energia em watts das GPUs alocadas ao nó.
node_gpu_temperature	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	A temperatura em graus Celsius das GPUs alocadas ao nó.

Nome da métrica	Dimensões	Descrição
node_gpu_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	A porcentagem de utilização das GPUs alocadas ao nó.
pod_gpu_memory_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	O tamanho total do buffer de quadros, em bytes, nas GPUs alocadas ao pod.

Nome da métrica	Dimensões	Descrição
pod_gpu_memory_used	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Os bytes do buffer de quadros usados nas GPUs alocadas ao pod.
pod_gpu_memory_utilization	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	A porcentagem de buffer de quadros usada das GPUs alocadas ao pod.

Nome da métrica	Dimensões	Descrição
pod_gpu_power_draw	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	O uso de energia em watts das GPUs alocadas ao pod.
pod_gpu_temperature	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	A temperatura em graus Celsius das GPUs alocadas ao pod.

Nome da métrica	Dimensões	Descrição
pod_gpu_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	A porcentagem de utilização das GPUs alocadas ao pod.

Métricas do AWS Neuron para o AWS Trainium e para o AWS Inferentia

A partir da versão `1.300036.0` do agente do CloudWatch, o Container Insights com observabilidade aprimorada para o Amazon EKS coleta métricas de computação acelerada dos aceleradores AWS Trainium e AWS Inferentia por padrão. O agente do CloudWatch deve ser instalado usando o complemento Observability do CloudWatch para o EKS na versão `v1.5.0-eksbuild.1` ou em versões posteriores. Para obter mais informações sobre o complemento, consulte Instalar o agente do CloudWatch usando o complemento de observabilidade do EKS do Amazon CloudWatch. Para obter mais informações sobre o AWS Trainium, consulte AWS Trainium. Para obter mais informações sobre o AWS Inferentia, consulte AWS Inferentia.

Para que o Container Insights colete métricas do AWS Neuron, você deve atender aos seguintes pré-requisitos:

- Você deve usar o Container Insights com observabilidade aprimorada para o Amazon EKS, com o complemento Observability do Amazon CloudWatch para o EKS na versão `v1.5.0-eksbuild.1` ou em versões posteriores.
- O driver Neuron deve estar instalado nos nós do cluster.
- O plug-in do dispositivo Neuron deve estar instalado no cluster. Por exemplo, as AMIs aceleradas otimizadas do Amazon EKS são criadas com os componentes necessários.

As métricas que são coletadas estão listadas na tabela desta seção. As métricas são coletadas para o AWS Trainium, o AWS Inferentia e o AWS Inferentia2.

O agente do CloudWatch coleta essas métricas do monitor do Neuron e realiza a correlação necessária de recursos do Kubernetes para fornecer métricas nos níveis de pod e de contêiner

Nome da métrica	Dimensões	Descrição
container_neuroncore_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	Utilização do NeuronCore, durante o período de captura do NeuronCore que está alocado para o contêiner. Unidade: Percentual
container_neuroncore_memory_usage_constants	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para constantes durante o treinamento pelo NeuronCore que está alocado para o contêiner (ou ponderações durante a inferência). Unidade: bytes
container_neuroncore_memory	ClusterName ClusterName , Namespace , PodName, ContainerName	A quantidade de memória do dispositivo usada para o código executável dos modelos pelo

Nome da métrica	Dimensões	Descrição
_usage_model_code	ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	NeuronCore que está alocado para o contêiner. Unidade: bytes
container_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para o rascunho compartilhado dos modelos pelo NeuronCore que está alocado para o contêiner. Esta região de memória está reservada para os modelos. Unidade: bytes
container_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para o runtime do Neuron pelo NeuronCore que está alocado para o contêiner. Unidade: bytes

Nome da métrica	Dimensões	Descrição
container_neuroncore_memory_usage_tensors	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para tensores pelo NeuronCore que está alocado para o contêiner. Unidade: bytes
container_neuroncore_memory_usage_total	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	A quantidade total de memória usada pelo NeuronCore que está alocado para o contêiner. Unidade: bytes

Nome da métrica	Dimensões	Descrição
container_neurondevice_hw_ecc_events_total	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice	O número de eventos do sistema ECC corrigidos e não corrigidos para a SRAM no chip e para a memória do dispositivo Neuron no nó. Unidade: Contagem
pod_neuroncore_utilization	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	A utilização do NeuronCore durante o período capturado do NeuronCore que está alocado para o pod. Unidade: Percentual

Nome da métrica	Dimensões	Descrição
pod_neuroncore_memory_usage_constants	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para constantes durante o treinamento pelo NeuronCore que está alocado para o pod (ou ponderações durante a inferência). Unidade: bytes
pod_neuroncore_memory_usage_model_code	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para o código executável dos modelos pelo NeuronCore que está alocado para o pod. Unidade: bytes

Nome da métrica	Dimensões	Descrição
pod_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para o rascunho compartilhado dos modelos pelo NeuronCore que está alocado para o pod. Esta região de memória está reservada para os modelos. Unidade: bytes
pod_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para o runtime do Neuron pelo NeuronCore que está alocado para o pod. Unidade: bytes

Nome da métrica	Dimensões	Descrição
pod_neuroncore_memory_usage_tensors	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para tensores pelo NeuronCore que está alocado para o pod. Unidade: bytes
pod_neuroncore_memory_usage_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	A quantidade total de memória usada pelo NeuronCore que está alocado para o pod. Unidade: bytes

Nome da métrica	Dimensões	Descrição
pod_neurondevice_hw_ecc_events_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice	O número de eventos do sistema ECC corrigidos e não corrigidos para a SRAM no chip e para a memória do dispositivo Neuron que está alocado para um pod. Unidade: bytes
node_neuroncore_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	A utilização do NeuronCore durante o período capturado do NeuronCore que está alocado para o nó. Unidade: Percentual
node_neuroncore_memory_usage_constants	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para constantes durante o treinamento pelo NeuronCore que está alocado para o nó (ou ponderações durante a inferência). Unidade: bytes

Nome da métrica	Dimensões	Descrição
node_neuroncore_memory_usage_model_code	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para o código executável dos modelos pelo NeuronCore que está alocado para o nó. Unidade: bytes
node_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para o rascunho compartilhado dos modelos pelo NeuronCore que está alocado para o nó. Esta é uma região de memória reservada para os modelos. Unidade: bytes
node_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para o runtime do Neuron pelo NeuronCore que está alocado para o nó. Unidade: bytes
node_neuroncore_memory_usage_tensors	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	A quantidade de memória do dispositivo usada para tensores pelo NeuronCore que está alocado para o nó. Unidade: bytes

Nome da métrica	Dimensões	Descrição
node_neuroncore_memory_usage_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	A quantidade total de memória usada pelo NeuronCore que está alocado para o nó. Unidade: bytes
node_neuron_execution_errors_total	ClusterName ClusterName , InstanceId , NodeName	O número total de erros de execução no nó. Isso é calculado pelo agente do CloudWatch ao agregar os erros dos seguintes tipos: generic, numerical , transient , model, runtime e hardware. Unidade: Contagem
node_neurondevice_runtime_memory_used_bytes	ClusterName ClusterName , InstanceId , NodeName	O uso total de memória do dispositivo Neuron em bytes no nó. Unidade: bytes
node_neuron_execution_latency	ClusterName ClusterName , InstanceId , NodeName	Em segundos, a latência para uma execução no nó medida pelo runtime do Neuron. Unidade: segundos
node_neurondevice_hw_ecc_events_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , NodeName, NeuronDevice	O número de eventos do sistema ECC corrigidos e não corrigidos para a SRAM no chip e para a memória do dispositivo Neuron no nó. Unidade: Contagem

Métricas do AWS Elastic Fabric Adapter (EFA)

A partir da versão `1.300037.0` do agente do CloudWatch, o Container Insights com observabilidade aprimorada para o Amazon EKS coleta métricas do AWS Elastic Fabric Adapter (EFA) de clusters do Amazon EKS em instâncias do Linux. O agente do CloudWatch deve ser instalado usando o complemento Observability do CloudWatch para o EKS na versão `v1.5.2-eksbuild.1` ou em versões posteriores. Para obter mais informações sobre o complemento, consulte Instalar o agente do CloudWatch usando o complemento de observabilidade do EKS do Amazon CloudWatch. Para obter mais informações sobre o AWS Elastic Fabric Adapter, consulte Elastic Fabric Adapter.

Para que o Container Insights colete métricas do AWS Elastic Fabric Adapter, você deve atender aos seguintes pré-requisitos:

- Você deve usar o Container Insights com observabilidade aprimorada para o Amazon EKS, com o complemento Observability do Amazon CloudWatch para o EKS na versão `v1.5.2-eksbuild.1` ou em versões posteriores.
- O plug-in do dispositivo EFA deve estar instalado no cluster. Para obter mais informações, consulte aws-efa-k8s-device-plugin no GitHub.

As métricas que são coletadas estão listadas na tabela apresentada a seguir.

Nome da métrica	Dimensões	Descrição

`container_efa_rx_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		O número de bytes por segundo que são recebidos pelos dispositivos EFA que estão alocados para o contêiner.
		
		Unidade: bytes/segundo

`container_efa_tx_bytes`	`ClusterName`	O número de bytes por segundo que são transmitidos pelos dispositi

Nome da métrica	Dimensões	Descrição

	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		vos EFA que estão alocados para o contêiner.
		
		Unidade: bytes/segundo

`container_efa_rx_dropped`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		O número de pacotes que foram recebidos e, em seguida, descartad os pelos dispositivos EFA que estão alocados para o contêiner.
		
		Unidade: contagem/segundo

`container_efa_rdma_read_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		O número de bytes por segundo recebidos usando operações de leitura de acesso remoto direto à memória pelos dispositivos EFA que estão alocados para o contêiner.
		
		Unidade: bytes/segundo

Nome da métrica	Dimensões	Descrição
container_efa_rdma_write_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	O número de bytes por segundo transmitidos usando operações de leitura de acesso remoto direto à memória pelos dispositivos EFA que estão alocados para o contêiner. Unidade: bytes/segundo
container_efa_rdma_write_recv_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	O número de bytes por segundo recebidos durante operações de gravação de acesso remoto direto à memória pelos dispositivos EFA que estão alocados para o contêiner. Unidade: bytes/segundo

Nome da métrica	Dimensões	Descrição
pod_efa_rx_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	O número de bytes por segundo que são recebidos pelos dispositivos EFA que estão alocados para o pod. Unidade: bytes/segundo
pod_efa_tx_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	O número de bytes por segundo que são transmitidos pelos dispositivos EFA que estão alocados para o pod. Unidade: bytes/segundo

Nome da métrica	Dimensões	Descrição
pod_efa_rx_dropped	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	O número de pacotes que foram recebidos e, em seguida, descartados pelos dispositivos EFA alocados para o pod. Unidade: contagem/segundo
pod_efa_rdma_read_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	O número de bytes por segundo recebidos usando operações de leitura de acesso remoto direto à memória pelos dispositivos EFA que estão alocados para o pod. Unidade: bytes/segundo

Nome da métrica	Dimensões	Descrição
pod_efa_rdma_write_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	O número de bytes por segundo transmitidos usando operações de leitura de acesso remoto direto à memória pelos dispositivos EFA que estão alocados para o pod. Unidade: bytes/segundo
pod_efa_rdma_write_recv_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	O número de bytes por segundo recebidos durante operações de gravação de acesso remoto direto à memória pelos dispositivos EFA que estão alocados para o pod. Unidade: bytes/segundo

Nome da métrica	Dimensões	Descrição
node_efa_rx_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	O número de bytes por segundo que são recebidos pelos dispositivos EFA que estão alocados para o nó. Unidade: bytes/segundo
node_efa_tx_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	O número de bytes por segundo que são transmitidos pelos dispositivos EFA que estão alocados para o nó. Unidade: bytes/segundo
node_efa_rx_dropped	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	O número de pacotes que foram recebidos e, em seguida, descartad os pelos dispositivos EFA que estão alocados para o nó. Unidade: contagem/segundo
node_efa_rdma_read_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	O número de bytes por segundo recebidos usando operações de leitura de acesso remoto direto à memória pelos dispositivos EFA que estão alocados para o nó. Unidade: bytes/segundo

Nome da métrica	Dimensões	Descrição
pod_efa_rdma_write_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	O número de bytes por segundo transmitidos usando operações de leitura de acesso remoto direto à memória pelos dispositivos EFA que estão alocados para o pod. Unidade: bytes/segundo
node_efa_rdma_write_recv_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	O número de bytes por segundo recebidos durante operações de gravação de acesso remoto direto à memória pelos dispositivos EFA que estão alocados para o nó. Unidade: bytes/segundo

Referência do log de performance do Container Insights

Esta seção contém informações de referência sobre como o Container Insights usa eventos de log de performance para coletar métricas. Quando você implanta o Container Insights, ele cria automaticamente um grupo de logs para os eventos do log de performance. Você não precisa criar esse grupo de logs sozinho.

Tópicos

- Eventos de log de performance do Container Insights para Amazon ECS
- Eventos do log de performance do Container Insights para Amazon EKS e Kubernetes
- Campos relevantes nos eventos de log de performance para Amazon EKS e Kubernetes

Eventos de log de performance do Container Insights para Amazon ECS

Veja a seguir exemplos dos eventos de log de performance que o Container Insights coleta do Amazon ECS.

Esses logs estão no CloudWatch Logs, em um grupo de logs chamado `/aws/ecs/containerinsights/`*`CLUSTER_NAME`*`/performance`. Dentro desse grupo de logs, cada instância de contêiner terá um fluxo de logs chamado `AgentTelemetry-`*`CONTAINER_INSTANCE_ID`*.

É possível consultar esses logs usando consultas como `{ $.Type = "Container" }` para visualizar todos os eventos do log de contêiner.

Tipo: Container

```
{
 "Version":"0",
 "Type":"Container",
 "ContainerName":"sleep",
 "TaskId":"7ac4dfba69214411b4783a3b8189c9ba",
 "TaskDefinitionFamily":"sleep360",
 "TaskDefinitionRevision":"1",
 "ContainerInstanceId":"0d7650e6dec34c1a9200f72098071e8f",
 "EC2InstanceId":"i-0c470579dbcdbd2f3",
 "ClusterName":"MyCluster",
 "Image":"busybox",
 "ContainerKnownStatus":"RUNNING",
 "Timestamp":1623963900000,
 "CpuUtilized":0.0,
 "CpuReserved":10.0,
 "MemoryUtilized":0,
 "MemoryReserved":10,
 "StorageReadBytes":0,
 "StorageWriteBytes":0,
 "NetworkRxBytes":0,
 "NetworkRxDropped":0,
 "NetworkRxErrors":0,
 "NetworkRxPackets":14,
 "NetworkTxBytes":0,
 "NetworkTxDropped":0,
 "NetworkTxErrors":0,
 "NetworkTxPackets":0
}
```

Tipo: tarefa

```
{
    "Version": "0",
    "Type": "Task",
```

```
    "TaskId": "7ac4dfba69214411b4783a3b8189c9ba",
    "TaskDefinitionFamily": "sleep360",
    "TaskDefinitionRevision": "1",
    "ContainerInstanceId": "0d7650e6dec34c1a9200f72098071e8f",
    "EC2InstanceId": "i-0c470579dbcdbd2f3",
    "ClusterName": "MyCluster",
    "AccountID": "637146863587",
    "Region": "us-west-2",
    "AvailabilityZone": "us-west-2b",
    "KnownStatus": "RUNNING",
    "LaunchType": "EC2",
    "PullStartedAt": 1623963608201,
    "PullStoppedAt": 1623963610065,
    "CreatedAt": 1623963607094,
    "StartedAt": 1623963610382,
    "Timestamp": 1623963900000,
    "CpuUtilized": 0.0,
    "CpuReserved": 10.0,
    "MemoryUtilized": 0,
    "MemoryReserved": 10,
    "StorageReadBytes": 0,
    "StorageWriteBytes": 0,
    "NetworkRxBytes": 0,
    "NetworkRxDropped": 0,
    "NetworkRxErrors": 0,
    "NetworkRxPackets": 14,
    "NetworkTxBytes": 0,
    "NetworkTxDropped": 0,
    "NetworkTxErrors": 0,
    "NetworkTxPackets": 0,
    "EBSFilesystemUtilized": 10,
    "EBSFilesystemSize": 20,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "CpuUtilized",
                    "Unit": "None"
                },
                {
                    "Name": "CpuReserved",
                    "Unit": "None"
                },
```

```
                {
                    "Name": "MemoryUtilized",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "MemoryReserved",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "StorageReadBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "StorageWriteBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "NetworkRxBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "NetworkTxBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "EBSFilesystemSize",
                    "Unit": "Gigabytes"
                },
                {
                    "Name": "EBSFilesystemUtilzed",
                    "Unit": "Gigabytes"
                }
            ],
            "Dimensions": [
                ["ClusterName"],
                [
                    "ClusterName",
                    "TaskDefinitionFamily"
                ]
            ]
        }
    ]
}
```

Tipo: Service

```
{
    "Version": "0",
    "Type": "Service",
    "ServiceName": "myCIService",
    "ClusterName": "myCICluster",
    "Timestamp": 1561586460000,
    "DesiredTaskCount": 2,
    "RunningTaskCount": 2,
    "PendingTaskCount": 0,
    "DeploymentCount": 1,
    "TaskSetCount": 0,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "DesiredTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "RunningTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "PendingTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "DeploymentCount",
                    "Unit": "Count"
                },
                {
                    "Name": "TaskSetCount",
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
                [
                    "ServiceName",
                    "ClusterName"
                ]
            ]
```

```
        }
    ]
}
```

Tipo: volume

```
{
    "Version": "0",
    "Type": "Volume",
    "TaskDefinitionFamily": "myCITaskDef",
    "TaskId": "7ac4dfba69214411b4783a3b8189c9ba",
    "ClusterName": "myCICluster",
    "ServiceName": "myCIService",
    "VolumeId": "vol-1233436545ff708cb",
    "InstanceId": "i-0c470579dbcdbd2f3",
    "LaunchType": "EC2",
    "VolumeName": "MyVolumeName",
    "EBSFilesystemUtilized": 10,
    "EBSFilesystemSize": 20,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "EBSFilesystemSize",
                    "Unit": "Gigabytes"
                },
                {
                    "Name": "EBSFilesystemUtilzed",
                    "Unit": "Gigabytes"
                }
            ],
            "Dimensions": [
                ["ClusterName"],
                [
                    "VolumeName",
                    "TaskDefinitionFamily",
                    "ClusterName"
                ],
                [
                    "ServiceName",
                    "ClusterName"
                ]
```

```
            ]
        }
    ]
}
```

Tipo: Cluster

```
{
    "Version": "0",
    "Type": "Cluster",
    "ClusterName": "myCICluster",
    "Timestamp": 1561587300000,
    "TaskCount": 5,
    "ContainerInstanceCount": 5,
    "ServiceCount": 2,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "TaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ContainerInstanceCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ServiceCount",
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
                [
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Eventos do log de performance do Container Insights para Amazon EKS e Kubernetes

Veja a seguir exemplos dos eventos de log de performance que o Container Insights coleta dos clusters do Amazon EKS e do Kubernetes.

Tipo: Node


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "node_network_total_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "node_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_reserved_capacity"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_pods"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_containers"
        }
      ],
      "Dimensions": [
```

```
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "node_network_total_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "node_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_reserved_capacity"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_pods"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_containers"
        },
        {
          "Name": "node_cpu_usage_total"
        },
        {
          "Name": "node_cpu_limit"
        },
```

```
        {
          "Unit": "Bytes",
          "Name": "node_memory_working_set"
        },
        {
          "Unit": "Bytes",
          "Name": "node_memory_limit"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "/proc",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567096682364",
  "Type": "Node",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_cpu_limit": 4000,
  "node_cpu_request": 1130,
  "node_cpu_reserved_capacity": 28.249999999999996,
  "node_cpu_usage_system": 33.794636630852764,
  "node_cpu_usage_total": 136.47852169244098,
  "node_cpu_usage_user": 71.67075111567326,
  "node_cpu_utilization": 3.4119630423110245,
  "node_memory_cache": 3103297536,
  "node_memory_failcnt": 0,
  "node_memory_hierarchical_pgfault": 0,
  "node_memory_hierarchical_pgmajfault": 0,
```

```
  "node_memory_limit": 16624865280,
  "node_memory_mapped_file": 406646784,
  "node_memory_max_usage": 4230746112,
  "node_memory_pgfault": 0,
  "node_memory_pgmajfault": 0,
  "node_memory_request": 1115684864,
  "node_memory_reserved_capacity": 6.7109407818311055,
  "node_memory_rss": 798146560,
  "node_memory_swap": 0,
  "node_memory_usage": 3901444096,
  "node_memory_utilization": 6.601302600149552,
  "node_memory_working_set": 1097457664,
  "node_network_rx_bytes": 35918.392817386324,
  "node_network_rx_dropped": 0,
  "node_network_rx_errors": 0,
  "node_network_rx_packets": 157.67565245448117,
  "node_network_total_bytes": 68264.20276554905,
  "node_network_tx_bytes": 32345.80994816272,
  "node_network_tx_dropped": 0,
  "node_network_tx_errors": 0,
  "node_network_tx_packets": 154.21455923431654,
  "node_number_of_running_containers": 16,
  "node_number_of_running_pods": 13
}
```

Tipo: NodeFS

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_filesystem_utilization"
        }
      ],
      "Dimensions": [
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
```

```
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097939726",
  "Type": "NodeFS",
  "Version": "0",
  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_filesystem_available": 17298395136,
  "node_filesystem_capacity": 21462233088,
  "node_filesystem_inodes": 10484720,
  "node_filesystem_inodes_free": 10367158,
  "node_filesystem_usage": 4163837952,
  "node_filesystem_utilization": 19.400767547940255
}
```

Tipo: NodeDiskIO

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
```

```
  "Sources": [
    "cadvisor"
  ],
  "Timestamp": "1567096928131",
  "Type": "NodeDiskIO",
  "Version": "0",
  "device": "/dev/nvme0n1",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_diskio_io_service_bytes_async": 9750.505814277016,
  "node_diskio_io_service_bytes_read": 0,
  "node_diskio_io_service_bytes_sync": 230.6174506688036,
  "node_diskio_io_service_bytes_total": 9981.123264945818,
  "node_diskio_io_service_bytes_write": 9981.123264945818,
  "node_diskio_io_serviced_async": 1.153087253344018,
  "node_diskio_io_serviced_read": 0,
  "node_diskio_io_serviced_sync": 0.03603397666700056,
  "node_diskio_io_serviced_total": 1.1891212300110185,
  "node_diskio_io_serviced_write": 1.1891212300110185
}
```

Tipo: NodeNet

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567096928131",
  "Type": "NodeNet",
  "Version": "0",
  "interface": "eni972f6bfa9a0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_interface_network_rx_bytes": 3163.008420864309,
```

```
  "node_interface_network_rx_dropped": 0,
  "node_interface_network_rx_errors": 0,
  "node_interface_network_rx_packets": 16.575629266820258,
  "node_interface_network_total_bytes": 3518.3935157426017,
  "node_interface_network_tx_bytes": 355.385094878293,
  "node_interface_network_tx_dropped": 0,
  "node_interface_network_tx_errors": 0,
  "node_interface_network_tx_packets": 3.9997714100370625
}
```

Tipo: Pod

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_rx_bytes"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_tx_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "pod_cpu_utilization_over_pod_limit"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization_over_pod_limit"
        }
      ],
```

```
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ],
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
        [
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_reserved_capacity"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
```

```
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "pod_number_of_container_restarts"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "Pod",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
```

```
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_cpu_limit": 200,
  "pod_cpu_request": 200,
  "pod_cpu_reserved_capacity": 5,
  "pod_cpu_usage_system": 1.4504841104992765,
  "pod_cpu_usage_total": 5.817016867430125,
  "pod_cpu_usage_user": 1.1281543081661038,
  "pod_cpu_utilization": 0.14542542168575312,
  "pod_cpu_utilization_over_pod_limit": 2.9085084337150624,
  "pod_memory_cache": 8192,
  "pod_memory_failcnt": 0,
  "pod_memory_hierarchical_pgfault": 0,
  "pod_memory_hierarchical_pgmajfault": 0,
  "pod_memory_limit": 104857600,
  "pod_memory_mapped_file": 0,
  "pod_memory_max_usage": 25268224,
  "pod_memory_pgfault": 0,
  "pod_memory_pgmajfault": 0,
  "pod_memory_request": 104857600,
  "pod_memory_reserved_capacity": 0.6307275170893897,
  "pod_memory_rss": 22777856,
  "pod_memory_swap": 0,
  "pod_memory_usage": 25141248,
  "pod_memory_utilization": 0.10988455961791709,
  "pod_memory_utilization_over_pod_limit": 17.421875,
  "pod_memory_working_set": 18268160,
  "pod_network_rx_bytes": 9880.697124714186,
  "pod_network_rx_dropped": 0,
  "pod_network_rx_errors": 0,
  "pod_network_rx_packets": 107.80005532263283,
  "pod_network_total_bytes": 10158.829201483635,
  "pod_network_tx_bytes": 278.13207676944796,
  "pod_network_tx_dropped": 0,
  "pod_network_tx_errors": 0,
  "pod_network_tx_packets": 1.146027574644318,
  "pod_number_of_container_restarts": 0,
  "pod_number_of_containers": 1,
  "pod_number_of_running_containers": 1,
  "pod_status": "Running"
```

```
}
```

Tipo: PodNet

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "PodNet",
  "Version": "0",
  "interface": "eth0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_interface_network_rx_bytes": 9880.697124714186,
  "pod_interface_network_rx_dropped": 0,
  "pod_interface_network_rx_errors": 0,
  "pod_interface_network_rx_packets": 107.80005532263283,
```

```
  "pod_interface_network_total_bytes": 10158.829201483635,
  "pod_interface_network_tx_bytes": 278.13207676944796,
  "pod_interface_network_tx_dropped": 0,
  "pod_interface_network_tx_errors": 0,
  "pod_interface_network_tx_packets": 1.146027574644318
}
```

Tipo: Container

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-NodeGroup-
sample",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "Container",
  "Version": "0",
  "container_cpu_limit": 200,
  "container_cpu_request": 200,
  "container_cpu_usage_system": 1.87958283771964,
  "container_cpu_usage_total": 6.159993652997942,
  "container_cpu_usage_user": 1.6707403001952357,
  "container_cpu_utilization": 0.15399984132494854,
  "container_memory_cache": 8192,
  "container_memory_failcnt": 0,
  "container_memory_hierarchical_pgfault": 0,
  "container_memory_hierarchical_pgmajfault": 0,
  "container_memory_limit": 104857600,
  "container_memory_mapped_file": 0,
  "container_memory_max_usage": 24580096,
  "container_memory_pgfault": 0,
  "container_memory_pgmajfault": 0,
  "container_memory_request": 104857600,
```

```
  "container_memory_rss": 22736896,
  "container_memory_swap": 0,
  "container_memory_usage": 24453120,
  "container_memory_utilization": 0.10574541028701798,
  "container_memory_working_set": 17580032,
  "container_status": "Running",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id":
 "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "number_of_container_restarts": 0
}
```

Tipo: ContainerFS

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
```

```
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "ContainerFS",
  "Version": "0",

  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id":
 "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  }
}
```

Tipo: Cluster

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
```

```
          "Name": "cluster_node_count"
        },
        {
          "Unit": "Count",
          "Name": "cluster_failed_node_count"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "Cluster",
  "Version": "0",
  "cluster_failed_node_count": 0,
  "cluster_node_count": 3
}
```

Tipo: ClusterService

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "service_number_of_running_pods"
        }
      ],
      "Dimensions": [
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
```

```
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "ClusterService",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch",
    "service_name": "cloudwatch-agent-statsd"
  },
  "service_number_of_running_pods": 1
}
```

Tipo: ClusterNamespace

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "namespace_number_of_running_pods"
        }
      ],
      "Dimensions": [
        [
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
```

```
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097594160",
  "Type": "ClusterNamespace",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch"
  },
  "namespace_number_of_running_pods": 7
}
```

Campos relevantes nos eventos de log de performance para Amazon EKS e Kubernetes

Para Amazon EKS e Kubernetes, o atendente do CloudWatch armazenado em contêineres emite dados como eventos de log de performance. Isso permite ao CloudWatch ingerir e armazenar dados de alta cardinalidade. O CloudWatch usa os dados nos eventos de log de performance para criar métricas agregadas do CloudWatch no nível do cluster, do nó e do pod, sem a necessidade de perder detalhes refinados.

A tabela a seguir lista os campos nesses eventos de log de performance que são relevantes à coleção de dados de métrica do Container Insights. Você pode usar o CloudWatch Logs Insights para consultar qualquer um desses campos para coletar dados ou investigar problemas. Para obter mais informações, consulte Analisar dados de log com o CloudWatch Logs Insights.

Tipo	Campo de log	Origem	Fórmula ou observações
Pod	`pod_cpu_utilization`	Calculado	Fórmula: `pod_cpu_usage_total / node_cpu_limit`

Tipo	Campo de log	Origem	Fórmula ou observações
Pod	`pod_cpu_usage_total` O `pod_cpu_usage_total` é informado em milinúcleos.	cadvisor	
Pod	`pod_cpu_limit`	Calculado	Fórmula: `sum(container_cpu_limit)` `sum(container_cpu_limit)` inclui pods já concluídos. Se qualquer contêiner no pod não tiver um limite de CPU definido, esse campo não aparecerá no evento de log. Isso inclui contêineres de inicialização.

Tipo	Campo de log	Origem	Fórmula ou observações
Pod	`pod_cpu_request`	Calculado	Fórmula: `sum(container_cpu_request)` Não é garantido que `container_cpu_request` esteja definido. Somente os que estão definidos são incluídos na soma.
Pod	`pod_cpu_utilization_over_pod_limit`	Calculado	Fórmula: `pod_cpu_usage_total / pod_cpu_limit`
Pod	`pod_cpu_reserved_capacity`	Calculado	Fórmula: `pod_cpu_request / node_cpu_limit`

Tipo	Campo de log	Origem	Fórmula ou observações
Pod	`pod_memory_utilization`	Calculado	Fórmula: `pod_memory_working_set / node_memory_limit` É a porcentagem de uso de memória do pod sobre a limitação de memória do nó.
Pod	`pod_memory_working_set`	cadvisor	
Pod	`pod_memory_limit`	Calculado	Fórmula: `sum(container_memory_limit)` Se qualquer contêiner no pod não tiver um limite de memória definido, esse campo não aparecerá no evento de log. Isso inclui contêineres de inicialização.

Tipo	Campo de log	Origem	Fórmula ou observações
Pod	`pod_memory_request`	Calculado	Fórmula: `sum(container_memory_request)` Não é garantido que `container_memory_request` esteja definido. Somente os que estão definidos são incluídos na soma.

Tipo	Campo de log	Origem	Fórmula ou observações
Pod	`pod_memory_utilization_over_pod_limit`	Calculado	Fórmula: `pod_memory_working_set / pod_memory_limit` Se qualquer contêiner no pod não tiver um limite de memória definido, esse campo não aparecerá no evento de log. Isso inclui contêineres de inicialização.
Pod	`pod_memory_reserved_capacity`	Calculado	Fórmula: `pod_memory_request / node_memory_limit`

Tipo	Campo de log	Origem	Fórmula ou observações
Pod	`pod_network_tx_bytes`	Calculado	Fórmula: `sum(pod_interface_network_tx_bytes)` Esses dados estão disponíveis para todas as interfaces de rede por pod. O atendente do CloudWatch calcula o total e adiciona regras de extração de métrica.
Pod	`pod_network_rx_bytes`	Calculado	Fórmula: `sum(pod_interface_network_rx_bytes)`
Pod	`pod_network_total_bytes`	Calculado	Fórmula: `pod_network_rx_bytes + pod_network_tx_bytes`

Tipo	Campo de log	Origem	Fórmula ou observações
PodNet	`pod_interface_network_rx_bytes`	cadvisor	Esses dados são bytes de rx de rede por segundo de uma interface de rede de pod.
PodNet	`pod_interface_network_tx_bytes`	cadvisor	Esses dados são bytes de tx de rede por segundo de uma interface de rede de pod.
Contêiner	`container_cpu_usage_total`	cadvisor	
Contêiner	`container_cpu_limit`	cadvisor	Não há garantia de estar definido. Não é emitido se não está definido.
Contêiner	`container_cpu_request`	cadvisor	Não há garantia de estar definido. Não é emitido se não está definido.
Contêiner	`container_memory_working_set`	cadvisor	

Tipo	Campo de log	Origem	Fórmula ou observações
Contêiner	`container_memory_limit`	pod	Não há garantia de estar definido. Não é emitido se não está definido.
Contêiner	`container_memory_request`	pod	Não há garantia de estar definido. Não é emitido se não está definido.
Nó	`node_cpu_utilization`	Calculado	Fórmula: `node_cpu_usage_total / node_cpu_limit`
Nó	`node_cpu_usage_total`	cadvisor	
Nó	`node_cpu_limit`	/proc	

Tipo	Campo de log	Origem	Fórmula ou observações
Nó	node_cpu_request	Calculado	Fórmula: sum(pod_cpu_request) Para cronjobs, node_cpu_request também inclui solicitações de pods concluídos. Isso pode levar a um alto valor para node_cpu_reserved_capacity .
Nó	node_cpu_reserved_capacity	Calculado	Fórmula: node_cpu_request / node_cpu_limit
Nó	node_memory_utilization	Calculado	Fórmula: node_memory_working_set / node_memory_limit
Nó	node_memory_working_set	cadvisor	
Nó	node_memory_limit	/proc	

Tipo	Campo de log	Origem	Fórmula ou observações
Nó	node_memory_request	Calculado	Fórmula: sum(pod_memory_request)
Nó	node_memory_reserved_capacity	Calculado	Fórmula: node_memory_request / node_memory_limit
Nó	node_network_rx_bytes	Calculado	Fórmula: sum(node_interface_network_rx_bytes)
Nó	node_network_tx_bytes	Calculado	Fórmula: sum(node_interface_network_tx_bytes)
Nó	node_network_total_bytes	Calculado	Fórmula: node_network_rx_bytes + node_network_tx_bytes
Nó	node_number_of_running_pods	Lista de pods	

Tipo	Campo de log	Origem	Fórmula ou observações
Nó	`node_number_of_running_containers`	Lista de pods	
NodeNet	`node_interface_network_rx_bytes`	cadvisor	Esses dados são bytes de tx de rede por segundo de uma interface de rede do nó de operador.
NodeNet	`node_interface_network_tx_bytes`	cadvisor	Esses dados são bytes de tx de rede por segundo de uma interface de rede do nó de operador.
NodeFS	`node_filesystem_capacity`	cadvisor	
NodeFS	`node_filesystem_usage`	cadvisor	

Tipo	Campo de log	Origem	Fórmula ou observações
NodeFS	`node_filesystem_utilization`	Calculado	Fórmula: `node_filesystem_usage / node_filesystem_capacity` Esses dados estão disponíveis por nome do dispositivo.
Cluster	`cluster_failed_node_count`	Servidor da API	
Cluster	`cluster_node_count`	Servidor da API	
Serviço	`service_number_of_running_pods`	Servidor da API	
`Namespace`	`namespace_number_of_running_pods`	Servidor da API	

Exemplos de cálculo de métricas

Esta seção inclui exemplos que mostram como alguns dos valores na tabela anterior são calculados.

Suponha que você tenha um cluster no estado a seguir.

```
Node1
   node_cpu_limit = 4
   node_cpu_usage_total = 3
```

```
   Pod1
     pod_cpu_usage_total = 2

     Container1
        container_cpu_limit = 1
        container_cpu_request = 1
        container_cpu_usage_total = 0.8

     Container2
        container_cpu_limit = null
        container_cpu_request = null
        container_cpu_usage_total = 1.2

   Pod2
     pod_cpu_usage_total = 0.4

     Container3
        container_cpu_limit = 1
        container_cpu_request = 0.5
        container_cpu_usage_total = 0.4

Node2
   node_cpu_limit = 8
   node_cpu_usage_total = 1.5

   Pod3
     pod_cpu_usage_total = 1

     Container4
        container_cpu_limit = 2
        container_cpu_request = 2
        container_cpu_usage_total = 1
```

A tabela a seguir mostra como as métricas de CPU do pod são calculadas usando esses dados.

Métrica	Fórmula	Pod1	Pod2	Pod3
`pod_cpu_utilization`	`pod_cpu_usage_total / node_cpu_limit`	2/4 = 50%	0,4/4 = 10%	1/8 = 12,5%

Métrica	Fórmula	Pod1	Pod2	Pod3
`pod_cpu_utilization_over_pod_limit`	`pod_cpu_usage_total / sum(container_cpu_limit)`	N/D, pois o limite de CPU para `Container 2` não está definido	0,4/1 = 40%	1/2 = 50%
`pod_cpu_reserved_capacity`	`sum(container_cpu_request) / node_cpu_limit`	(1 + 0)/4 = 25%	0,5/4 = 12,5%	2/8 = 25%

A tabela a seguir mostra como as métricas de CPU do nó são calculadas usando esses dados.

Métrica	Fórmula	Node1	Node2
`node_cpu_utilization`	`node_cpu_usage_total / node_cpu_limit`	3/4 = 75%	1,5/8 = 18,75%
`node_cpu_reserved_capacity`	`sum(pod_cpu_request) / node_cpu_limit`	1,5/4 = 37,5%	2/8 = 25%

Monitoramento de métricas do Container Insights Prometheus

O monitoramento do CloudWatch Container Insights para Prometheus automatiza a detecção de métricas do Prometheus de sistemas e workloads em contêineres. O Prometheus é um toolkit de código aberto para alertas e monitoramento de sistemas. Para obter mais informações, consulte What is Prometheus? na documentação do Prometheus.

A detecção de métricas do Prometheus é compatível com clusters do Amazon Elastic Container Service, Amazon Elastic Kubernetes Service e Kubernetes em execução em instâncias do Amazon EC2. São coletados os tipos de métrica contador, medidor e resumo do Prometheus. O suporte para métricas de histograma está planejado para um lançamento futuro.

Para clusters do Amazon ECS e do Amazon EKS, há suporte para os tipos de inicialização do EC2 e do Fargate. O Container Insights coleta automaticamente métricas de várias workloads, e é possível configurá-lo de modo a coletar métricas de qualquer workload.

Você pode adotar o Prometheus como um método de código aberto e padrão aberto para ingerir métricas personalizadas no CloudWatch. O atendente do CloudWatch com suporte ao Prometheus detecta e coleta métricas do Prometheus para monitorar, solucionar problemas e criar alarmes sobre a degradação na performance e falhas das aplicações mais rapidamente. Isso também reduz o número de ferramentas de monitoramento necessárias para melhorar a observabilidade.

O suporte do Container Insights Prometheus envolve o pagamento de métricas e logs conforme o uso, incluindo coleta, armazenamento e análise. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

Painéis pré-criados para algumas workloads

A solução Container Insights Prometheus contém painéis pré-criados para as workloads bastante utilizadas listadas nesta seção. Para obter exemplos de configurações dessas workloads, consulte (Opcional) Configurar amostra de workloads do Amazon ECS em contêineres para teste de métrica do Prometheuse (Opcional) Configurar workloads de exemplo do Amazon EKS em contêineres para teste de métrica do Prometheus.

Também é possível configurar o Container Insights para coletar métricas do Prometheus de outros serviços e aplicações em contêineres, editando o arquivo de configuração do atendente.

Workloads com painéis pré-criados para clusters do Amazon EKS e do Kubernetes em execução em instâncias do Amazon EC2:

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy

Workloads com painéis pré-criados para clusters do Amazon ECS:

- AWS App Mesh
- Java/JMX

- NGINX
- NGINX Plus

Instalar e configurar a coleta de métricas do Prometheus em cluster do Amazon ECS

Para coletar métricas do Prometheus de clusters do Amazon ECS, é possível usar o atendente do CloudWatch como coletor ou usar o coletor do AWS Distro for OpenTelemetry. Para obter informações sobre como usar o coletor do AWS Distro for OpenTelemetry, consulte https://aws-otel.github.io/docs/getting-started/container-insights/ecs-prometheus.

As seções a seguir explicam como usar o atendente do CloudWatch como coletor para recuperar métricas do Prometheus. Você instala o atendente do CloudWatch com o monitoramento Prometheus em clusters que executam o Amazon ECS e, opcionalmente, pode configurar o atendente para extrair outros destinos. Estas seções também fornecem tutoriais opcionais para configurar workloads de amostra para usar testes com monitoramento Prometheus.

O Container Insights no Amazon ECS oferece suporte às seguintes combinações de tipo de inicialização e modo de rede para métricas do Prometheus:

Tipo de inicialização do Amazon ECS	Modos de rede compatíveis
EC2 (Linux)	bridge, host e awsvpc
Fargate	awsvpc

Requisitos para grupo de segurança de VPC

As regras de entrada dos grupos de segurança para as workloads do Prometheus devem abrir as portas do Prometheus para o atendente do CloudWatch para extrair as métricas Prometheus pelo IP privado.

As regras de saída do grupo de segurança do atendente do CloudWatch devem permitir que o atendente do CloudWatch se conecte à porta das workloads do Prometheus por IP privado.

Tópicos

- Instalar o atendente do CloudWatch com a coleção de métricas do Prometheus em clusters do Amazon ECS
- Extrair outras fontes do Prometheus e importar essas métricas

- (Opcional) Configurar amostra de workloads do Amazon ECS em contêineres para teste de métrica do Prometheus

Instalar o atendente do CloudWatch com a coleção de métricas do Prometheus em clusters do Amazon ECS

Esta seção explica como configurar o atendente do CloudWatch com monitoramento do Prometheus em um cluster que está executando o Amazon ECS. Depois que você fizer isso, o atendente automaticamente extrairá e importará métricas para as seguintes workloads em execução nesse cluster.

- AWS App Mesh
- Java/JMX

Também é possível configurar o atendente para extrair e importar métricas de outras workloads e origens do Prometheus.

Configurar funções do IAM

Você precisa de duas funções do IAM para a definição de tarefa do atendente do CloudWatch. Se você especificar **`CreateIAMRoles=True`** na pilha AWS CloudFormation para que o Container Insights crie essas funções para você, as funções serão criadas com as permissões corretas. Caso queira criá-las ou usar funções existentes, as funções e permissões a seguir são necessárias.

- Função de tarefa do ECS do atendente do CloudWatch: o contêiner do atendente do CloudWatch usa essa função. Ela deve incluir a política CloudWatchAgentServerPolicy e uma política gerenciada pelo cliente que contenha as seguintes permissões somente para leitura:
  - `ec2:DescribeInstances`
  - `ecs:ListTasks`
  - `ecs:ListServices`
  - `ecs:DescribeContainerInstances`
  - `ecs:DescribeServices`
  - `ecs:DescribeTasks`
  - `ecs:DescribeTaskDefinition`
- Função de execução de tarefa do ECS do atendente do CloudWatch: essa é a função necessária para o Amazon ECS iniciar e executar os contêineres. Verifique se sua função de execução de

tarefa tem as políticas AmazonSSMReadOnlyAccess, AmazonECSTaskExecutionRolePolicy e CloudWatchAgentServerPolicy anexadas. Se precisar armazenar dados mais sigilosos para uso do Amazon ECS, consulte Especificar dados sigilosos.

Instale o atendente do CloudWatch com o monitoramento do Prometheus usando AWS CloudFormation

Você pode usar AWS CloudFormation para instalar o atendente do CloudWatch com o monitoramento do Prometheus para clusters do Amazon ECS. A lista a seguir exibe os parâmetros que você usará no modelo AWS CloudFormation.

- ECSClusterName: especifica o cluster do Amazon ECS de destino.
- CreateIAMRoles: especifique **`True`** para criar novas funções para a função de tarefa do Amazon ECS e para a função de execução de tarefas do Amazon ECS. Especifique **`False`** para reutilizar funções existentes.
- TaskRoleName: se você especificou **`True`** em CreateIAMRoles, isso especifica o nome a ser usado para a função de tarefa do Amazon ECS. Se você especificou **`False`** em CreateIAMRoles, isso especifica a função existente a ser usada para a função de tarefa do Amazon ECS.
- ExecutionRoleName: se você especificou **`True`** em CreateIAMRoles, isso especifica o nome a ser usado para a função de execução de tarefa do Amazon ECS. Se você especificou **`False`** em CreateIAMRoles, isso especifica a função existente a ser usada para a função de execução de tarefa do Amazon ECS.
- ECSNetworkMode: se estiver usando o tipo de inicialização do EC2, especifique o modo de rede aqui. Deve ser **`bridge`** ou **`host`**.
- ECSLaunchType: especifique **`fargate`** ou **`EC2`**.
- SecurityGroupID: se o ECSNetworkMode for **`awsvpc`**, especifique o ID do grupo de segurança aqui.
- SubnetID: se o ECSNetworkMode for **`awsvpc`**, especifique o ID da sub-rede aqui.

Exemplos de comando

Esta seção contém exemplos de comando do AWS CloudFormation para instalar o Container Insights com o monitoramento do Prometheus em vários cenários.

Criar uma pilha do AWS CloudFormation para um cluster do Amazon ECS no modo de rede de ponte

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_NETWORK_MODE=bridge
export CREATE_IAM_ROLES=True
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Criar uma pilha do AWS CloudFormation para um cluster do Amazon ECS no modo de rede de host

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_NETWORK_MODE=host
export CREATE_IAM_ROLES=True
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name


curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
```

```
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Criar uma pilha do AWS CloudFormation para um cluster do Amazon ECS no modo de rede awsvpc

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_LAUNCH_TYPE=EC2
export CREATE_IAM_ROLES=True
export ECS_CLUSTER_SECURITY_GROUP=your_security_group_eg_sg-xxxxxxxxxx
export ECS_CLUSTER_SUBNET=your_subnet_eg_subnet-xxxxxxxxxx
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-${ECS_LAUNCH_TYPE}-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSLaunchType,ParameterValue=${ECS_LAUNCH_TYPE} \
                 ParameterKey=SecurityGroupID,ParameterValue=
${ECS_CLUSTER_SECURITY_GROUP} \
                 ParameterKey=SubnetID,ParameterValue=${ECS_CLUSTER_SUBNET} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Criar uma pilha do AWS CloudFormation para um cluster do Fargate no modo de rede awsvpc

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_LAUNCH_TYPE=FARGATE
export CREATE_IAM_ROLES=True
export ECS_CLUSTER_SECURITY_GROUP=your_security_group_eg_sg-xxxxxxxxxx
export ECS_CLUSTER_SUBNET=your_subnet_eg_subnet-xxxxxxxxxx
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-${ECS_LAUNCH_TYPE}-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSLaunchType,ParameterValue=${ECS_LAUNCH_TYPE} \
                 ParameterKey=SecurityGroupID,ParameterValue=
${ECS_CLUSTER_SECURITY_GROUP} \
                 ParameterKey=SubnetID,ParameterValue=${ECS_CLUSTER_SUBNET} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Recursos da AWS criados pela pilha do AWS CloudFormation

A tabela a seguir lista os recursos da AWS que são criados quando você usa o AWS CloudFormation para configurar o Container Insights com o monitoramentodo Prometheus em um cluster do Amazon ECS.

Tipo de recurso	Nome do recurso	Comentários
AWS::SSM::Parameter	AmazonCloudWatch-CWAgentConfig-$*ECS_CLUSTER_NAME* -$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	Este é o atendente do CloudWatch com a definição padrão do formato de métrica incorporado de App Mesh e Java/JMX.
AWS::SSM::Parameter	AmazonCloudWatch-Prometheus ConfigName-$*ECS_CLUSTER_NAME* -$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	Esta é a configuração de extração do Prometheus.
AWS::IAM::Perfil	$ECS_TASK_ROLE_NAME.	A função de tarefa do Amazon ECS. Isso somente é criado se você especific ou **`True`** em `CREATE_IAM_ROLES` .
AWS::IAM::Perfil	${ECS_EXECUTION_ROLE_NAME}	A função de execução de tarefa do Amazon ECS. Isso somente é criado se você especificou **`True`** em `CREATE_IAM_ROLES` .
AWS::ECS::TaskDefinition	cwagent-prometheus-$*ECS_CLUSTER_NAME* -$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	
AWS::ECS::Service	cwagent-prometheus-replica-service-$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	

Excluir a pilha do AWS CloudFormation para o atendente do CloudWatch com monitoramento do Prometheus

Para excluir o atendente do CloudWatch de um cluster do Amazon ECS, insira estes comandos.

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export CLOUDFORMATION_STACK_NAME=your_cloudformation_stack_name
```

```
aws cloudformation delete-stack \
--stack-name ${CLOUDFORMATION_STACK_NAME} \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Extrair outras fontes do Prometheus e importar essas métricas

O atendente do CloudWatch com monitoramento Prometheus precisa de duas configurações para extrair as métricas do Prometheus. Uma serve para as configurações padrão do Prometheus, conforme documentado em <scrape_config> na documentação do Prometheus. A outra é para a configuração do atendente do CloudWatch.

Para clusters do Amazon ECS, as configurações são integradas ao Parameter Store do AWS Systems Manager pelos segredos na definição da tarefa do Amazon ECS:

- O segredo `PROMETHEUS_CONFIG_CONTENT` é para a configuração de extração do Prometheus.
- O segredo `CW_CONFIG_CONTENT` é para a configuração do atendente do CloudWatch.

Para extrair outras origens de métricas do Prometheus e importar essas métricas para o CloudWatch, modifique a configuração de extração do Prometheus e a configuração do atendente do CloudWatch e implante novamente o atendente com a configuração atualizada.

Requisitos para grupo de segurança de VPC

As regras de entrada dos grupos de segurança para as workloads do Prometheus devem abrir as portas do Prometheus para o atendente do CloudWatch para extrair as métricas Prometheus pelo IP privado.

As regras de saída do grupo de segurança do atendente do CloudWatch devem permitir que o atendente do CloudWatch se conecte à porta das workloads do Prometheus por IP privado.

Configuração de extração do Prometheus

O atendente do CloudWatch oferece suporte às configurações de extração padrão do Prometheus, conforme documentado em <scrape_config> na documentação do Prometheus. É possível editar essa seção para atualizar as configurações que já estão nesse arquivo e adicionar outros destinos de extração do Prometheus. Por padrão, um exemplo de arquivo de configuração contém as seguintes linhas de configuração global:

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
```

- scrape_interval: define a frequência da adição de destinos de extração de conteúdo.
- scrape_timeout: define quanto tempo aguardar até a expiração de uma solicitação de extração de conteúdo.

Também é possível definir valores diferentes para essas configurações no nível do trabalho, a fim de substituir as configurações globais.

Trabalhos de extração do Prometheus

Os arquivos YAML do atendente do CloudWatch já têm alguns trabalhos padrão de extração configurados. Por exemplo, nos arquivos YAML para o Amazon ECS, como `cwagent-ecs-prometheus-metric-for-bridge-host.yaml`, os trabalhos de extração padrão são configurados na seção `ecs_service_discovery`.

```
"ecs_service_discovery": {
                  "sd_frequency": "1m",
                  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
                  "docker_label": {
                  },
                  "task_definition_list": [
                    {
                      "sd_job_name": "ecs-appmesh-colors",
                      "sd_metrics_ports": "9901",
                      "sd_task_definition_arn_pattern": ".*:task-definition\/.*-
ColorTeller-(white):[0-9]+",
                      "sd_metrics_path": "/stats/prometheus"
                    },
                    {
                      "sd_job_name": "ecs-appmesh-gateway",
                      "sd_metrics_ports": "9901",
                      "sd_task_definition_arn_pattern": ".*:task-definition/.*-
ColorGateway:[0-9]+",
                      "sd_metrics_path": "/stats/prometheus"
                    }
                  ]
                }
```

Cada um desses destinos padrão é extraído e as métricas são enviadas ao CloudWatch em eventos de log usando o formato de métricas incorporado. Para ter mais informações, consulte Incorporação de métricas em logs.

Os eventos de log dos clusters do Amazon ECS são armazenados no grupo de logs /aws/ecs/containerinsights/***cluster_name***/prometheus.

Cada trabalho de extração está contido em um stream de log diferente nesse grupo de logs.

Para adicionar um novo destino de extração, adicione uma nova entrada à seção `task_definition_list` na seção `ecs_service_discovery` do arquivo YAML e reinicie o atendente. Para obter um exemplo desse processo, consulte Tutorial para adicionar um novo destino de extração do Prometheus: métricas do servidor de API do Prometheus.

Configuração do atendente do CloudWatch para o Prometheus

O arquivo de configuração do atendente do CloudWatch tem uma seção `prometheus` na seção `metrics_collected` para a configuração de extração do Prometheus. Contém as seguintes opções de configuração:

- cluster_name: especifica o nome do cluster a ser adicionado como um rótulo no evento de log. Esse campo é opcional. Se você omitir, o atendente poderá detectar o nome do cluster do Amazon ECS.
- log_group_name: especifica o nome do grupo de log para as métricas do Prometheus extraídas. Esse campo é opcional. Se você omitir, o CloudWatch usará /aws/ecs/containerinsights/***cluster_name***/prometheus para logs de clusters do Amazon ECS.
- prometheus_config_path: especifica o caminho do arquivo de configuração de extração do Prometheus. Se o valor desse campo começar com `env:`, o conteúdo do arquivo de configuração de extração do Prometheus será recuperado da variável de ambiente do contêiner. Não altere esse campo.
- ecs_service_discovery: é a seção para especificar as configurações das funções de detecção automática de destino do Amazon ECS Prometheus. Dois modos são compatíveis para detectar os destinos do Prometheus: detecção baseada no rótulo do docker do contêiner ou detecção baseada na expressão regular do ARN da definição de tarefa do Amazon ECS. Você pode usar os dois modos junto, e o atendente do CloudWatch eliminará a duplicação dos destinos detectados com base em: {private_ip}:{port}/{metrics_path}.

  A seção `ecs_service_discovery` pode conter os seguintes campos:

- `sd_frequency` é a frequência para detectar os exportadores Prometheus. Especifique um número e um sufixo de unidade. Por exemplo, `1m` uma vez por minuto ou `30s` uma vez a cada 30 segundos. Os sufixos de unidade válidos são: `ns`, `us`, `ms`, `s`, `m` e `h`.

  Esse campo é opcional. O padrão é 60 segundos (1 minuto).

- `sd_target_cluster` é o nome do cluster do Amazon ECS de destino para detecção automática. Esse campo é opcional. O padrão é o nome do cluster do Amazon ECS em que o atendente do CloudWatch está instalado.
- `sd_cluster_region` é a região do cluster do Amazon ECS de destino. Esse campo é opcional. O padrão é a região do cluster do Amazon ECS em que o atendente do CloudWatch está instalado.
- `sd_result_file` é o caminho do arquivo YAML para os resultados de destino do Prometheus. A configuração de extração do Prometheus referenciará esse arquivo.
- `docker_label` é uma seção opcional que você pode usar para especificar a configuração para detecção de serviço baseada em rótulos do docker. Se você omitir essa seção, a detecção baseada em rótulos do docker não será usada. A seção pode conter os seguintes campos:
  - `sd_port_label` é o nome do rótulo do docker do contêiner que especifica a porta do contêiner para métricas do Prometheus. O valor padrão é `ECS_PROMETHEUS_EXPORTER_PORT`. Se o contêiner não tiver esse rótulo do docker, o atendente do CloudWatch o ignorará.
  - `sd_metrics_path_label` é o nome do rótulo do docker do contêiner que especifica o caminho das métricas do Prometheus. O valor padrão é `ECS_PROMETHEUS_METRICS_PATH`. Se o contêiner não tiver esse rótulo do docker, o agente assumirá o caminho padrão `/metrics`.
  - `sd_job_name_label` é o nome do rótulo do docker do contêiner que especifica o nome do trabalho de extração do Prometheus. O valor padrão é `job`. Se o contêiner não tiver esse rótulo do docker, o atendente do CloudWatch usará o nome do trabalho na configuração de extração do Prometheus.
- `task_definition_list` é uma seção opcional que você pode usar para especificar a configuração para detectção de serviço baseada em definição de tarefa. Se você omitir essa seção, a detecção baseada em definição de tarefa não será usada. A seção pode conter os seguintes campos:
  - `sd_task_definition_arn_pattern` é o padrão a ser usado para especificar as definições de tarefa do Amazon ECS a serem detectadas. Essa é uma expressão regular.

  - `sd_metrics_ports` lista a containerPort para as métricas do Prometheus. Separe as containerPorts com ponto e vírgula.
  - `sd_container_name_pattern` especifica os nomes de contêiner de tarefas do Amazon ECS. Essa é uma expressão regular.
  - `sd_metrics_path` especifica o caminho da métrica do Prometheus. Se você omitir isso, o atendente assumirá o caminho padrão `/metrics`
  - `sd_job_name` especifica o nome do trabalho de extração do Prometheus. Se você omitir esse campo, o atendente do CloudWatch usará o nome do trabalho na configuração de extração do Prometheus.
- `service_name_list_for_tasks` é uma seção opcional que você pode usar para especificar a configuração para detecção de serviço baseada em definição de tarefa. Ao omitir esta seção, a detecção baseada em nome de serviço não será utilizada. A seção pode conter os seguintes campos:
  - `sd_service_name_pattern` é o padrão a ser usado para especificar o serviço do Amazon ECS em que as tarefas serão detectadas. Essa é uma expressão regular.
  - `sd_metrics_ports` lista a `containerPort` para as métricas do Prometheus. Separar múltiplas `containerPorts` com ponto e vírgula.
  - `sd_container_name_pattern` especifica os nomes de contêiner de tarefas do Amazon ECS. Essa é uma expressão regular.
  - `sd_metrics_path` especifica o caminho das métricas do Prometheus. Se você omitir isso, o atendente entenderá que o caminho padrão é `/metrics`.
  - `sd_job_name` especifica o nome do trabalho de extração do Prometheus. Se você omitir esse campo, o atendente do CloudWatch usará o nome do trabalho na configuração de extração do Prometheus.
- metric_declaration: são seções que especificam a matriz de logs com formato de métrica incorporado a ser gerada. Há seções `metric_declaration` para cada destino do Prometheus do qual o atendente do CloudWatch importa por padrão. Essas seções incluem os seguintes campos:
  - `label_matcher` é uma expressão regular que confere o valor dos rótulos listados em `source_labels`. As métricas correspondentes são disponibilizadas para inclusão no formato de métrica incorporado enviado ao CloudWatch.

    Se você tiver vários rótulos especificados em `source_labels`, recomendamos não utilizar os caracteres ^ ou $ na expressão regular para `label_matcher`.

- `source_labels` especifica o valor dos rótulos verificados pela linha `label_matcher`.
- `label_separator` especifica o separador a ser usado na linha `label_matcher` se vários `source_labels` forem especificados. O padrão é `;`. É possível ver esse padrão usado na linha `label_matcher` no exemplo a seguir.
- `metric_selectors` é uma expressão regular que especifica as métricas a serem coletadas e enviadas ao CloudWatch.
- `dimensions` é a lista de rótulos a serem usados como dimensões do CloudWatch para cada métrica selecionada.

Veja o exemplo de `metric_declaration` a seguir.

```
"metric_declaration": [
  {
     "source_labels":[ "Service", "Namespace"],
     "label_matcher":"(.*node-exporter.*|.*kube-dns.*);kube-system$",
     "dimensions":[
        ["Service", "Namespace"]
     ],
     "metric_selectors":[
        "^coredns_dns_request_type_count_total$"
     ]
  }
]
```

Esse exemplo configura uma seção de formato de métrica incorporada a ser enviada como um evento de log se as seguintes condições forem atendidas:

- O valor de `Service` contém `node-exporter` ou `kube-dns`.
- O valor de `Namespace` é `kube-system`.
- A métrica do Prometheus `coredns_dns_request_type_count_total` contém rótulos `Namespace` e `Service`.

O evento de log enviado inclui a seguinte seção destacada:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
```

```
            {
               "Name":"coredns_dns_request_type_count_total"
            }
         ],
         "Dimensions":[
            [
               "Namespace",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "Namespace":"kube-system",
   "Service":"kube-dns",
   "coredns_dns_request_type_count_total":2562,
   "eks_amazonaws_com_component":"kube-dns",
   "instance":"192.168.61.254:9153",
   "job":"kubernetes-service-endpoints",
   ...
}
```

Guia detalhado para detecção automática em clusters do Amazon ECS

O Prometheus fornece dezenas de mecanismos dinâmicos de detecção de serviços, conforme descrito em <scrape_config>. Porém, não há detecção de serviço integrada para o Amazon ECS. O atendente do CloudWatch adiciona esse mecanismo.

Quando a detecção de serviço do Amazon ECS Prometheus é habilitada, o atendente do CloudWatch faz periodicamente as seguintes chamadas de API aos frontends do Amazon ECS e do Amazon EC2 para recuperar os metadados das tarefas do ECS em execução no cluster do ECS de destino.

```
EC2:DescribeInstances
ECS:ListTasks
ECS:ListServices
ECS:DescribeContainerInstances
ECS:DescribeServices
ECS:DescribeTasks
ECS:DescribeTaskDefinition
```

Os metadados são usados pelo atendente do CloudWatch para verificar os destinos do Prometheus dentro do cluster do ECS. O atendente do CloudWatch é compatível com três modos de detecção de serviço:

- Detecção de serviço baseada em rótulos do docker
- Detecção de serviço baseado em expressão regular do ARN da definição de tarefa do ECS
- Detecção de serviço baseada em expressão regular do nome do serviço do ECS

Todos os modos podem ser usados conjuntamente. O atendente do CloudWatch elimina a duplicação dos destinos detectados com base em: `{private_ip}:{port}/{metrics_path}`.

Todos os destinos detectados são gravados em um arquivo de resultado especificado pela confiiguração `sd_result_file` dentro do contêiner do atendente do CloudWatch. Veja a seguir um exemplo de arquivo de resultado:

```
- targets:
  - 10.6.1.95:32785
  labels:
    __metrics_path__: /metrics
    ECS_PROMETHEUS_EXPORTER_PORT: "9406"
    ECS_PROMETHEUS_JOB_NAME: demo-jar-ec2-bridge-dynamic
    ECS_PROMETHEUS_METRICS_PATH: /metrics
    InstanceType: t3.medium
    LaunchType: EC2
    SubnetId: subnet-123456789012
    TaskDefinitionFamily: demo-jar-ec2-bridge-dynamic-port
    TaskGroup: family:demo-jar-ec2-bridge-dynamic-port
    TaskRevision: "7"
    VpcId: vpc-01234567890
    container_name: demo-jar-ec2-bridge-dynamic-port
    job: demo-jar-ec2-bridge-dynamic
- targets:
  - 10.6.3.193:9404
  labels:
    __metrics_path__: /metrics
    ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_B: "9404"
    ECS_PROMETHEUS_JOB_NAME: demo-tomcat-ec2-bridge-mapped-port
    ECS_PROMETHEUS_METRICS_PATH: /metrics
    InstanceType: t3.medium
    LaunchType: EC2
    SubnetId: subnet-123456789012
```

```
    TaskDefinitionFamily: demo-tomcat-ec2-bridge-mapped-port
    TaskGroup: family:demo-jar-tomcat-bridge-mapped-port
    TaskRevision: "12"
    VpcId: vpc-01234567890
    container_name: demo-tomcat-ec2-bridge-mapped-port
    job: demo-tomcat-ec2-bridge-mapped-port
```

É possível integrar diretamente esse arquivo de resultados à detecção de serviços baseada em arquivo do Prometheus. Para obter mais informações sobre a detecção de serviços baseada em arquivos do Prometheus, consulte <file_sd_config>.

Suponha que o arquivo resultante seja gravado em `/tmp/cwagent_ecs_auto_sd.yaml`. A seguinte configuração de extração do Prometheus o consumirá.

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: cwagent-ecs-file-sd-config
    sample_limit: 10000
    file_sd_configs:
      - files: [ "/tmp/cwagent_ecs_auto_sd.yaml" ]
```

O atendente do CloudWatch também acrescenta os seguintes rótulos adicionais aos destinos descobertos.

- `container_name`
- `TaskDefinitionFamily`
- `TaskRevision`
- `TaskGroup`
- `StartedBy`
- `LaunchType`
- `job`
- `__metrics_path__`
- Rótulos do Docker

Quando o cluster tiver o tipo de inicialização do EC2, os três rótulos a seguir serão adicionados.

- `InstanceType`
- `VpcId`
- `SubnetId`

> **Note**
>
> Os rótulos do Docker que não correspondem à expressão regular `[a-zA-Z_][a-zA-Z0-9_]*` são filtrados. Isso corresponde às convenções do Prometheus listadas em `label_name`em Arquivo de configuração na documentação do Prometheus.

Exemplos de configuração de detecção de serviços do ECS

Esta seção contém exemplos que demonstram a detecção de serviços do ECS.

Exemplo 1

```
"ecs_service_discovery": {
  "sd_frequency": "1m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
  }
}
```

Este exemplo habilita a detecção de serviços baseada em rótulos do docker. O atendente do CloudWatch consultará os metadados das tarefas do ECS uma vez por minuto e gravará os destinos detectados no arquivo `/tmp/cwagent_ecs_auto_sd.yaml` dentro do contêiner do atendente do CloudWatch.

O valor padrão de `sd_port_label` na seção `docker_label` é `ECS_PROMETHEUS_EXPORTER_PORT`. Se qualquer contêiner em execução nas tarefas do ECS tiver um rótulo do docker `ECS_PROMETHEUS_EXPORTER_PORT`, o atendente do CloudWatch usará seu valor como `container port` para verificar todas as portas expostas do contêiner. Se houver uma correspondência, a porta do host mapeada mais o IP privado do contêiner serão usados para construir o destino do exportador do Prometheus neste formato: `private_ip:host_port`.

O valor padrão de `sd_metrics_path_label` na seção `docker_label` é `ECS_PROMETHEUS_METRICS_PATH`. Se o contêiner tiver esse rótulo do docker, seu valor será

usado como `__metrics_path__`. Se o contêiner não tiver esse rótulo, será usado o valor padrão `/metrics`.

O valor padrão de `sd_job_name_label` na seção `docker_label` é `job`. Se o contêiner tiver esse rótulo do docker, seu valor será anexado como um dos rótulos do destino para substituir o nome do trabalho padrão especificado na configuração do Prometheus. O valor desse rótulo do docker será usado como o nome do fluxo de logs no grupo de logs do CloudWatch Logs.

Exemplo 2

```
"ecs_service_discovery": {
  "sd_frequency": "15s",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
    "sd_port_label": "ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_A",
    "sd_job_name_label": "ECS_PROMETHEUS_JOB_NAME"
  }
}
```

Este exemplo habilita a detecção de serviços baseada em rótulos do docker. O atendente do CloudWatch consultará os metadados das tarefas do ECS a cada 15 segundos e gravará os destinos detectados no arquivo `/tmp/cwagent_ecs_auto_sd.yaml` dentro do contêiner do atendente do CloudWatch. Os contêineres com o rótulo do docker `ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_A` serão verificados. O valor do rótulo do docker `ECS_PROMETHEUS_JOB_NAME` é usado como nome do trabalho.

Exemplo 3

```
"ecs_service_discovery": {
  "sd_frequency": "5m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "task_definition_list": [
    {
      "sd_job_name": "java-prometheus",
      "sd_metrics_path": "/metrics",
      "sd_metrics_ports": "9404; 9406",
      "sd_task_definition_arn_pattern": ".*:task-definition/.*javajmx.*:[0-9]+"
    },
    {
      "sd_job_name": "envoy-prometheus",
      "sd_metrics_path": "/stats/prometheus",
      "sd_container_name_pattern": "^envoy$",
```

```
      "sd_metrics_ports": "9901",
      "sd_task_definition_arn_pattern": ".*:task-definition/.*appmesh.*:23"
    }
  ]
}
```

Este exemplo habilita a detecção de serviço baseado em expressão regular do ARN da definição de tarefa do ECS. O atendente do CloudWatch consultará os metadados das tarefas do ECS a cada cinco minutos e gravará os destinos detectados no arquivo `/tmp/cwagent_ecs_auto_sd.yaml` dentro do contêiner do atendente do CloudWatch.

São definidas duas seções de expressão regular do ARN da definição de tarefa:

- Na primeira seção, as tarefas do ECS com `javajmx` no ARN da definição de tarefa do ECS são filtradas para a verificação da porta do contêiner. Se os contêineres dentro dessas tarefas do ECS expuserem a porta do contêiner em 9404 ou 9406, a porta do host mapeada juntamente com o IP privado do contêiner serão usados para criar os destinos do exportador do Prometheus. O valor de `sd_metrics_path` define `__metrics_path__` como `/metrics`. Assim, o atendente do CloudWatch extrairá as métricas do Prometheus de `private_ip:host_port/metrics`, e as métricas extraídas serão enviadas ao fluxo de logs `java-prometheus` no CloudWatch Logs no grupo de logs `/aws/ecs/containerinsights/cluster_name/prometheus`.
- Na segunda seção, as tarefas do ECS com `appmesh` no ARN da definição de tarefa do ECS e com `version` de `:23` são filtradas para a verificação da porta do contêiner. Para contêineres com um nome de `envoy` que expõem a porta do contêiner em `9901`, a porta de host mapeada e o IP privado do contêiner são usados para criar os destinos do exportador do Prometheus. Se o valor dentro dessas tarefas do ECS expuser a porta do contêiner em 9404 ou 9406, a porta do host mapeada e o IP privado do contêiner serão usados para criar os destinos do exportador do Prometheus. O valor de `sd_metrics_path` define `__metrics_path__` como `/stats/prometheus`. Assim, o atendente do CloudWatch extrairá as métricas do Prometheus de `private_ip:host_port/stats/prometheus` e enviará as métricas extraídas ao fluxo de logs `envoy-prometheus` no CloudWatch Logs no grupo de logs `/aws/ecs/containerinsights/cluster_name/prometheus`.

Exemplo 4

```
"ecs_service_discovery": {
  "sd_frequency": "5m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
```

```
  "service_name_list_for_tasks": [
    {
      "sd_job_name": "nginx-prometheus",
      "sd_metrics_path": "/metrics",
      "sd_metrics_ports": "9113",
      "sd_service_name_pattern": "^nginx-.*"
    },
    {
      "sd_job_name": "haproxy-prometheus",
      "sd_metrics_path": "/stats/metrics",
      "sd_container_name_pattern": "^haproxy$",
      "sd_metrics_ports": "8404",
      "sd_service_name_pattern": ".*haproxy-service.*"
    }
  ]
}
```

Este exemplo habilita a detecção de serviço baseado em expressão regular do nome do serviço do ECS. O atendente do CloudWatch consultará os metadados dos serviços do ECS a cada cinco minutos e gravará os destinos detectados no arquivo `/tmp/cwagent_ecs_auto_sd.yaml` dentro do contêiner do atendente do CloudWatch.

São definidas duas seções de expressão regular de nome de serviço:

- Na primeira seção, as tarefas do ECS associadas aos serviços do ECS que têm nomes correspondentes à expressão regular `^nginx-.*`são filtrados para a verificação da porta do contêiner. Se os contêineres dentro dessas tarefas do ECS expuserem a porta do contêiner em 9113, a porta do host mapeada e o IP privado do contêiner serão usados para criar os destinos do exportador do Prometheus. O valor de `sd_metrics_path` define `__metrics_path__` como `/metrics`. Assim, o atendente do CloudWatch extrairá as métricas do Prometheus de `private_ip:host_port/metrics`, e as métricas extraídas serão enviadas ao fluxo de logs `nginx-prometheus` no CloudWatch Logs no grupo de logs `/aws/ecs/containerinsights/cluster_name/prometheus`.
- Na segunda seção, as tarefas do ECS associadas aos serviços do ECS que têm nomes correspondentes à expressão regular `.*haproxy-service.*`são filtrados para a verificação da porta do contêiner. Para contêineres com um nome de `haproxy` que expõem a porta do contêiner em 8404, a porta de host mapeada e o IP privado do contêiner são usados para criar os destinos do exportador do Prometheus. O valor de `sd_metrics_path` define `__metrics_path__` como `/stats/metrics`. Assim, o atendente do CloudWatch extrairá as métricas do Prometheus de `private_ip:host_port/stats/metrics`, e as métricas extraídas serão enviadas

ao fluxo de logs `haproxy-prometheus` no CloudWatch Logs no grupo de logs `/aws/ecs/containerinsights/cluster_name/prometheus`.

Exemplo 5

```
"ecs_service_discovery": {
  "sd_frequency": "1m30s",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
    "sd_port_label": "MY_PROMETHEUS_EXPORTER_PORT_LABEL",
    "sd_metrics_path_label": "MY_PROMETHEUS_METRICS_PATH_LABEL",
    "sd_job_name_label": "MY_PROMETHEUS_METRICS_NAME_LABEL"
  }
  "task_definition_list": [
    {
      "sd_metrics_ports": "9150",
      "sd_task_definition_arn_pattern": "*memcached.*"
    }
  ]
}
```

Este exemplo habilita os dois modos de detecção de serviços do ECS. O atendente do CloudWatch consultará os metadados das tarefas do ECS a cada 90 segundos e gravará os destinos detectados no arquivo `/tmp/cwagent_ecs_auto_sd.yaml` dentro do contêiner do atendente do CloudWatch.

Para a configuração de detecção de serviços baseada em docker:

- As tarefas do ECS com rótulo do docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` serão filtradas para a verificação de porta do Prometheus. A porta do contêiner do Prometheus de destino é especificada pelo valor do rótulo `MY_PROMETHEUS_EXPORTER_PORT_LABEL`.
- Utiliza-se o valor do rótulo do docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` para `__metrics_path__`. Se o contêiner não tiver esse rótulo do docker, será usado o valor padrão `/metrics`.
- O valor do rótulo do docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` é usado como rótulo do trabalho. Se o contêiner não tiver esse rótulo do docker, será usado o nome do trabalho definido na configuração do Prometheus.

Para configuração da detecção de serviço baseado em expressão regular do ARN da definição de tarefa do ECS:

- As tarefas do ECS com `memcached` no ARN da definição de tarefa do ECS são filtradas para verificação da porta do contêiner. A porta de contêiner do Prometheus de destino é 9150, conforme definido por `sd_metrics_ports`. É usado o caminho padrão das métricas `/metrics`. É usado o nome do trabalho definido na configuração do Prometheus.

(Opcional) Configurar amostra de workloads do Amazon ECS em contêineres para teste de métrica do Prometheus

Para testar o suporte para métricas do Prometheus no CloudWatch Container Insights, você pode configurar uma ou mais das seguintes workloads em contêineres. O atendente do CloudWatch com suporte ao Prometheus coleta automaticamente métricas de cada uma dessas workloads. Para visualizar as métricas coletadas por padrão, consulte Métricas do Prometheus coletadas pelo atendente do CloudWatch.

Tópicos

- Exemplo de workload do App Mesh para clusters do Amazon ECS
- Exemplo de workload do Java/JMX para clusters do Amazon ECS
- Exemplo de workload do NGINX para clusters do Amazon ECS
- Exemplo de workload do NGINX Plus para clusters do Amazon ECS
- Tutorial para adicionar um novo destino de extração do Prometheus: Memcached no Amazon ECS
- Tutorial para extração de métricas do Redis Prometheus no Amazon ECS Fargate

Exemplo de workload do App Mesh para clusters do Amazon ECS

Para coletar métricas de uma amostra de workload do Prometheus para o Amazon ECS, é necessário estar executando o Container Insights no cluster. Para obter informações sobre como instalar o Container Insights, consulte Configurar o Container Insights no Amazon ECS.

Primeiro, siga esta demonstração para implantar a amostra de aplicação de cores em seu cluster do Amazon ECS. Ao terminar, você terá métricas do App Mesh Prometheus expostas na porta 9901.

Em seguida, siga estas etapas para instalar o atendente do CloudWatch com o monitoramento Prometheus no mesmo cluster do Amazon ECS em que você instalou a aplicação de cores. As etapas desta seção instalam o atendente do CloudWatch no modo de rede de ponte.

As variáveis de ambiente `ENVIRONMENT_NAME`, `AWS_PROFILE` e `AWS_DEFAULT_REGIONque` você definir na demonstração também serão usadas nas etapas a seguir.

Para instalar o atendente do CloudWatch com monitoramento do Prometheus para teste

1. Baixe o modelo do AWS CloudFormation inserindo o comando a seguir.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
bridge-host.yaml
```

2. Defina o modo de rede inserindo os comandos a seguir.

```
export ECS_CLUSTER_NAME=${ENVIRONMENT_NAME}
export ECS_NETWORK_MODE=bridge
```

3. Crie a pilha do AWS CloudFormation inserindo os comandos abaixo.

```
aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=CWAgent-Prometheus-
TaskRole-${ECS_CLUSTER_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=CWAgent-Prometheus-
ExecutionRole-${ECS_CLUSTER_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

4. (Opcional) Quando a pilha do AWS CloudFormation for criada, você verá a mensagem `CREATE_COMPLETE`. Se conferir o status antes de visualizar essa mensagem, insira o comando a seguir.

```
aws cloudformation describe-stacks \
--stack-name CWAgent-Prometheus-ECS-${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
--query 'Stacks[0].StackStatus' \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Solução de problemas

As etapas da demonstração usam jq para analisar o resultado de saída da AWS CLI. Para obter mais informações sobre como instalar o jq, consulte jq. Use o comando a seguir para definir o formato de saída padrão da AWS CLI para JSON, de modo que o jq possa analisá-lo corretamente.

```
$ aws configure
```

Quando a resposta chegar a `Default output format`, insira **`json`**.

Instalar o atendente do CloudWatch com monitoramento do Prometheus

Quando terminar de testar, insira o comando a seguir para desinstalar o atendente do CloudWatch excluindo a pilha do AWS CloudFormation.

```
aws cloudformation delete-stack \
--stack-name CWAgent-Prometheus-ECS-${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Exemplo de workload do Java/JMX para clusters do Amazon ECS

O JMX Exporter é um exportador oficial do Prometheus que pode extrair conteúdo e expor mBeans da JMX como métricas do Prometheus. Para obter mais informações, consulte prometheus/jmx_exporter.

O atendente do CloudWatch com suporte ao Prometheus extrai as métricas do Java/JMX Prometheus com base na configuração de detecção de serviço no cluster do Amazon ECS. Você pode configurar o JMX Exporter de modo a expor as métricas em uma porta ou metrics_path diferente. Se alterar a porta ou o caminho, atualize a seção `ecs_service_discovery` padrão na configuração do atendente do CloudWatch.

Para coletar métricas de uma amostra de workload do Prometheus para o Amazon ECS, é necessário estar executando o Container Insights no cluster. Para obter informações sobre como instalar o Container Insights, consulte Configurar o Container Insights no Amazon ECS.

Para instalar o exemplo de workload do Java/JMX para clusters do Amazon ECS

1. Siga as etapas destas seções para criar suas imagens do Docker.

   - Exemplo: imagem do Docker da aplicação Java Jar com métricas do Prometheus

- Exemplo: imagem do Docker do Apache Tomcat com métricas do Prometheus

2. Especifique os dois rótulos do docker a seguir no arquivo de definição de tarefa do Amazon ECS. Em seguida, é possível executar a definição da tarefa como um serviço do Amazon ECS ou uma tarefa do Amazon ECS no cluster.

   - Defina `ECS_PROMETHEUS_EXPORTER_PORT` para apontar para a containerPort onde as métricas do Prometheus estão expostas.
   - Defina `Java_EMF_Metrics` como `true`. O atendente do CloudWatch usa esse sinalizador para gerar o formato de métrica incorporado no evento de log.

   Veja um exemplo a seguir:

```
{
  "family": "workload-java-ec2-bridge",
  "taskRoleArn": "{{task-role-arn}}",
  "executionRoleArn": "{{execution-role-arn}}",
  "networkMode": "bridge",
  "containerDefinitions": [
    {
      "name": "tomcat-prometheus-workload-java-ec2-bridge-dynamic-port",
      "image": "your_docker_image_tag_for_tomcat_with_prometheus_metrics",
      "portMappings": [
        {
          "hostPort": 0,
          "protocol": "tcp",
          "containerPort": 9404
        }
      ],
      "dockerLabels": {
        "ECS_PROMETHEUS_EXPORTER_PORT": "9404",
        "Java_EMF_Metrics": "true"
      }
    }
  ],
  "requiresCompatibilities": [
    "EC2"  ],
  "cpu": "256",
  "memory": "512"
  }
```

A configuração padrão do atendente do CloudWatch no modelo AWS CloudFormation permite a detecção de serviços baseada em rótulos do docker e a detecção de serviços baseada no ARN da definição de tarefa. Para exibir essas configurações padrão, consulte a linha 65 do arquivo de configurações YAML do atendente do CloudWatch. Os contêineres com o rótulo `ECS_PROMETHEUS_EXPORTER_PORT` serão descobertos automaticamente com base na porta de contêiner especificada para extração do Prometheus.

A configuração padrão do atendente do CloudWatch também tem a configuração `metric_declaration` para Java/JMX na linha 112 do mesmo arquivo. Todos os rótulos do docker dos contêineres de destino serão acrescentados como rótulos adicionais nas métricas do Prometheus e enviados ao CloudWatch Logs. Para os contêineres Java/JMX com rótulo do docker `Java_EMF_Metrics="true"`, será gerado o formato de métrica incorporado.

Exemplo de workload do NGINX para clusters do Amazon ECS

O exportador do NGINX Prometheus pode extrair e expor dados do NGINX como métricas do Prometheus. Este exemplo usa o exportador com o serviço de proxy reverso do NGINX para o Amazon ECS.

Para obter mais informações sobre o exportador do NGINX Prometheus, consulte nginx-prometheus-exporter no Github. Para obter mais informações sobre o proxy reverso do NGINX, consulte ecs-nginx-reverse-proxy no Github.

O atendente do CloudWatch com suporte ao Prometheus extrai as métricas do NGINX Prometheus com base na configuração de detecção de serviço no cluster do Amazon ECS. Você pode configurar o NGINX Prometheus Exporter de modo a expor as métricas em uma porta ou um caminho diferente. Se alterar a porta ou o caminho, atualize a seção `ecs_service_discovery` no arquivo de configuração do atendente do CloudWatch.

Instalar o exemplo de workload de proxy reverso do NGINX para clusters do Amazon ECS

Siga estas etapas para instalar o exemplo de workload de proxy reverso do NGINX.

Criar as imagens do Docker

Para criar as imagens do Docker para o exemplo de workload de proxy reverso do NGINX

1. Baixe esta pasta do repositório de proxy reverso do NGINX: https://github.com/awslabs/ecs-nginx-reverse-proxy/tree/master/reverse-proxy.
2. Encontre o diretório `app` e crie uma imagem a partir desse diretório:

```
docker build -t web-server-app ./path-to-app-directory
```

3. Crie uma imagem personalizada para o NGINX. Primeiro, crie um diretório com estes dois arquivos:

   - Um exemplo de Dockerfile:

     ```
     FROM nginx
     COPY nginx.conf /etc/nginx/nginx.conf
     ```

   - Um arquivo `nginx.conf`, modificado a partir de https://github.com/awslabs/ecs-nginx-reverse-proxy/tree/master/reverse-proxy:

     ```
     events {
       worker_connections 768;
     }

     http {
       # Nginx will handle gzip compression of responses from the app server
       gzip on;
       gzip_proxied any;
       gzip_types text/plain application/json;
       gzip_min_length 1000;

       server{
         listen 8080;
         location /stub_status {
             stub_status   on;
         }
       }

       server {
         listen 80;

         # Nginx will reject anything not matching /api
         location /api {
           # Reject requests with unsupported HTTP method
           if ($request_method !~ ^(GET|POST|HEAD|OPTIONS|PUT|DELETE)$) {
             return 405;
           }

           # Only requests matching the whitelist expectations will
     ```

```
      # get sent to the application server
      proxy_pass http://app:3000;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection 'upgrade';
      proxy_set_header Host $host;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_cache_bypass $http_upgrade;
    }
  }
}
```

> **Note**
>
> `stub_status` deve estar habilitado na mesma porta de onde `nginx-prometheus-exporter` está configurado para extrair métricas. Em nosso exemplo de definição de tarefa, `nginx-prometheus-exporter` está configurado para extrar métricas da porta 8080.

4. Crie uma imagem a partir de arquivos em seu novo diretório:

```
docker build -t nginx-reverse-proxy ./path-to-your-directory
```

5. Carregue as novas imagens para um repositório de imagens para uso posterior.

Crie a definição de tarefa para executar o NGINX e a aplicação de servidor da Web no Amazon ECS

Em seguida, configure a definição de tarefa.

Essa definição de tarefa permite a coleta e exportação de métricas do NGINX Prometheus. O contêiner do NGINX rastreia a entrada da aplicação e expõe esses dados à porta 8080, conforme definido em `nginx.conf`. O contêiner do exportador do NGINX Prometheus extrai essas métricas e as publica na porta 9113, para serem usadas no CloudWatch.

Para configurar a definição de tarefa para o exemplo de workload do NGINX no Amazon ECS

1. Crie um arquivo JSON de definição de tarefa com o seguinte conteúdo. Substitua *`your-customized-nginx-iamge`* pelo URI de imagem para sua imagem do NGINX personalizada e substitua *`your-web-server-app-image`* pelo URI de imagem de sua imagem de aplicação do servidor da Web.

```
{
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "your-customized-nginx-image",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "links": [
        "app"
      ]
    },
    {
      "name": "app",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 256,
      "essential": true
    },
    {
      "name": "nginx-prometheus-exporter",
      "image": "docker.io/nginx/nginx-prometheus-exporter:0.8.0",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "command": [
        "-nginx.scrape-uri",
        "http://nginx:8080/stub_status"
    ],
    "links":[
      "nginx"
    ],
      "portMappings":[
        {
          "containerPort": 9113,
          "protocol": "tcp"
        }
```

```
      ]
    }
  ],
  "networkMode": "bridge",
  "placementConstraints": [],
  "family": "nginx-sample-stack"
}
```

2. Insira o comando a seguir para registrar a definição de tarefa.

```
aws ecs register-task-definition --cli-input-json file://path-to-your-task-definition-json
```

3. Crie um serviço para executar a tarefa inserindo o comando a seguir:

   Não altere o nome do serviço. Executaremos um serviço de atendente do CloudWatch usando uma configuração que procura tarefas usando os padrões de nome dos serviços que os iniciaram. Por exemplo, para que o atendente do CloudWatch localize a tarefa iniciada por este comando, é possível especificar o valor de `sd_service_name_pattern` como `^nginx-service$`. A próxima seção oferece mais detalhes.

```
aws ecs create-service \
 --cluster your-cluster-name \
 --service-name nginx-service \
 --task-definition nginx-sample-stack:1 \
 --desired-count 1
```

Configurar o atendente do CloudWatch para extrair métricas do NGINX Prometheus

A etapa final é configurar o atendente do CloudWatch para extrair as métricas do NGINX. Neste exemplo, o atendente do CloudWatch detecta a tarefa por meio do padrão de nome de serviço e da porta 9113, onde o exportador expõe as métricas prometheus para o NGINX. Com a tarefa detectada e as métricas disponíveis, o atendente do CloudWatch começa a publicar as métricas coletadas no fluxo de logs nginx-prometheus-exporter.

Para configurar o atendente do CloudWatch para extrair métricas do NGINX

1. Baixe a versão mais recente do arquivo YAML necessário inserindo um dos comandos a seguir.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
bridge-host.yaml
```

2. Abra o arquivo com um editor de textos e encontre a confguração completa do atendente do CloudWatch na chave `value` da seção `resource:CWAgentConfigSSMParameter`. Depois, na seção `ecs_service_discovery`, adicione a seção `service_name_list_for_tasks` a seguir.

```
"service_name_list_for_tasks": [
  {
    "sd_job_name": "nginx-prometheus-exporter",
    "sd_metrics_path": "/metrics",
    "sd_metrics_ports": "9113",
    "sd_service_name_pattern": "^nginx-service$"
   }
],
```

3. No mesmo arquivo, insira a seguinte seção na seção `metric_declaration` para permitir métricas do NGINX. Siga o padrão de recuo existente.

```
{
  "source_labels": ["job"],
  "label_matcher": ".*nginx.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName"]],
  "metric_selectors": [
    "^nginx_.*$"
  ]
},
```

4. Se você ainda não tiver o atendente do CloudWatch implantado nesse cluster, pule para a etapa 8.

   Se já tiver o atendente do CloudWatch implantado no cluster do Amazon ECS usando o AWS CloudFormation, é possível criar um conjunto de alterações inserindo os seguintes comandos:

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
```

```
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name nginx-scraping-support
```

5. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
6. Revisar o changeset recém-criado nginx-scraping-support. Você deverá ver uma alteração aplicada ao recurso CWAgentConfigSSMParameter. Execute o changeset e reinicie a tarefa do atendente do CloudWatch inserindo este comando:

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Aguarde cerca de 10 segundos e insira este comando.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

8. Se você estiver instalando o atendente do CloudWatch com coleta de métricas do Prometheus no cluster pela primeira vez, insira estes comandos.

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
```

```
aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Visualizar métricas e logs do NGINX

Agora é possível visualizar as métricas do NGINX que estão sendo coletadas.

Para visualizar as métricas do exemplo de workload do NGINX

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Na região da em que o cluster está em execução, escolha Metrics (Métricas) no painel de navegação à esquerda. Encontre o namespace ContainerInsights/Prometheus para ver as métricas.
3. Para visualizar os eventos do CloudWatch Logs, escolha Log Groups (Grupos de logs) no painel de navegação. Os eventos estão no grupo de logs /aws/containerinsights/***your_cluster_name***/prometheus, no fluxo de logs *nginx-prometheus-exporter*.

Exemplo de workload do NGINX Plus para clusters do Amazon ECS

O NGINX Plus é a versão comercial do NGINX. É necessário ter uma licença para usá-lo. Para obter mais informações, consulte NGINX Plus.

O exportador do NGINX Prometheus pode extrair e expor dados do NGINX como métricas do Prometheus. Este exemplo usa o exportador com o serviço de proxy reverso do NGINX Plus para o Amazon ECS.

Para obter mais informações sobre o exportador do NGINX Prometheus, consulte nginx-prometheus-exporter no Github. Para obter mais informações sobre o proxy reverso do NGINX, consulte ecs-nginx-reverse-proxy no Github.

O atendente do CloudWatch com suporte ao Prometheus extrai as métricas do NGINX Plus Prometheus com base na configuração de detecção de serviço no cluster do Amazon ECS. Você pode configurar o NGINX Prometheus Exporter de modo a expor as métricas em uma porta ou um caminho diferente. Se alterar a porta ou o caminho, atualize a seção `ecs_service_discovery` no arquivo de configuração do atendente do CloudWatch.

Instalar o exemplo de workload de proxy reverso do NGINX Plus para clusters do Amazon ECS

Siga estas etapas para instalar o exemplo de workload de proxy reverso do NGINX.

Criar as imagens do Docker

Para criar as imagens do Docker para o exemplo de workload de proxy reverso do NGINX Plus

1. Baixe esta pasta do repositório de proxy reverso do NGINX: https://github.com/awslabs/ecs-nginx-reverse-proxy/tree/master/reverse-proxy.
2. Encontre o diretório `app` e crie uma imagem a partir desse diretório:

   ```
   docker build -t web-server-app ./path-to-app-directory
   ```

3. Crie uma imagem personalizada para o NGINX Plus. Para poder criar a imagem para o NGINX Plus, é necessário obter a chave chamada `nginx-repo.key` e o certificado SSL `nginx-repo.crt` para seu NGINX Plus licenciado. Crie um diretório e armazene em seus arquivos `nginx-repo.key` e `nginx-repo.crt`.

   No diretório que você acabou de criar, crie estes dois arquivos:

   - Um exemplo de Dockerfile com o conteúdo a seguir. Este arquivo do docker é adotado a partir de um arquivo de exemplo disponível em https://docs.nginx.com/nginx/admin-guide/installing-nginx/installing-nginx-docker/#docker_plus_image. A mudança importante que fazemos é carregar um arquivo separado, chamado `nginx.conf`, que será criado na próxima etapa.

     ```
     FROM debian:buster-slim

     LABEL maintainer="NGINX Docker Maintainers <docker-maint@nginx.com>“

     # Define NGINX versions for NGINX Plus and NGINX Plus modules
     ```

```
# Uncomment this block and the versioned nginxPackages block in the main RUN
# instruction to install a specific release
# ENV NGINX_VERSION 21
# ENV NJS_VERSION 0.3.9
# ENV PKG_RELEASE 1~buster

# Download certificate and key from the customer portal (https://cs.nginx.com
 (https://cs.nginx.com/))
# and copy to the build context
COPY nginx-repo.crt /etc/ssl/nginx/
COPY nginx-repo.key /etc/ssl/nginx/
# COPY nginx.conf /etc/ssl/nginx/nginx.conf

RUN set -x \
# Create nginx user/group first, to be consistent throughout Docker variants
&& addgroup --system --gid 101 nginx \
&& adduser --system --disabled-login --ingroup nginx --no-create-home --home /
nonexistent --gecos "nginx user" --shell /bin/false --uid 101 nginx \
&& apt-get update \
&& apt-get install --no-install-recommends --no-install-suggests -y ca-
certificates gnupg1 \
&& \
NGINX_GPGKEY=573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62; \
found=''; \
for server in \
ha.pool.sks-keyservers.net (http://ha.pool.sks-keyservers.net/) \
hkp://keyserver.ubuntu.com:80 \
hkp://p80.pool.sks-keyservers.net:80 \
pgp.mit.edu (http://pgp.mit.edu/) \
; do \
echo "Fetching GPG key $NGINX_GPGKEY from $server"; \
apt-key adv --keyserver "$server" --keyserver-options timeout=10 --recv-keys
 "$NGINX_GPGKEY" && found=yes && break; \
done; \
test -z "$found" && echo >&2 "error: failed to fetch GPG key $NGINX_GPGKEY" &&
 exit 1; \
apt-get remove --purge --auto-remove -y gnupg1 && rm -rf /var/lib/apt/lists/* \
# Install the latest release of NGINX Plus and/or NGINX Plus modules
# Uncomment individual modules if necessary
# Use versioned packages over defaults to specify a release
&& nginxPackages=" \
nginx-plus \
# nginx-plus=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-xslt \
```

```
# nginx-plus-module-xslt=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-geoip \
# nginx-plus-module-geoip=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-image-filter \
# nginx-plus-module-image-filter=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-perl \
# nginx-plus-module-perl=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-njs \
# nginx-plus-module-njs=${NGINX_VERSION}+${NJS_VERSION}-${PKG_RELEASE} \
" \
&& echo "Acquire::https::plus-pkgs.nginx.com::Verify-Peer \"true\";" >> /etc/apt/
apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::Verify-Host \"true\";" >> /etc/apt/
apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::SslCert \"/etc/ssl/nginx/nginx-
repo.crt\";" >> /etc/apt/apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::SslKey \"/etc/ssl/nginx/nginx-
repo.key\";" >> /etc/apt/apt.conf.d/90nginx \
&& printf "deb https://plus-pkgs.nginx.com/debian buster nginx-plus\n" > /etc/
apt/sources.list.d/nginx-plus.list \
&& apt-get update \
&& apt-get install --no-install-recommends --no-install-suggests -y \
$nginxPackages \
gettext-base \
curl \
&& apt-get remove --purge --auto-remove -y && rm -rf /var/lib/apt/lists/* /etc/
apt/sources.list.d/nginx-plus.list \
&& rm -rf /etc/apt/apt.conf.d/90nginx /etc/ssl/nginx

# Forward request logs to Docker log collector
RUN ln -sf /dev/stdout /var/log/nginx/access.log \
&& ln -sf /dev/stderr /var/log/nginx/error.log

COPY nginx.conf /etc/nginx/nginx.conf

EXPOSE 80

STOPSIGNAL SIGTERM

CMD ["nginx", "-g", "daemon off;"]
```

- Um arquivo `nginx.conf`, modificado a partir de https://github.com/awslabs/ecs-nginx-reverse-proxy/tree/master/reverse-proxy/nginx.

```
events {
  worker_connections 768;
}

http {
  # Nginx will handle gzip compression of responses from the app server
  gzip on;
  gzip_proxied any;
  gzip_types text/plain application/json;
  gzip_min_length 1000;

  upstream backend {
    zone name 10m;
    server app:3000    weight=2;
    server app2:3000    weight=1;
  }

  server{
    listen 8080;
    location /api {
      api write=on;
    }
  }

  match server_ok {
    status 100-599;
  }

  server {
    listen 80;
    status_zone zone;
    # Nginx will reject anything not matching /api
    location /api {
      # Reject requests with unsupported HTTP method
      if ($request_method !~ ^(GET|POST|HEAD|OPTIONS|PUT|DELETE)$) {
        return 405;
      }

      # Only requests matching the whitelist expectations will
      # get sent to the application server
      proxy_pass http://backend;
      health_check uri=/lorem-ipsum match=server_ok;
      proxy_http_version 1.1;
```

```
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_cache_bypass $http_upgrade;
    }
  }
}
```

4. Crie uma imagem a partir de arquivos em seu novo diretório:

```
docker build -t nginx-plus-reverse-proxy ./path-to-your-directory
```

5. Carregue as novas imagens para um repositório de imagens para uso posterior.

Crie a definição de tarefa para executar o NGINX Plus e a aplicação de servidor da Web no Amazon ECS

Em seguida, configure a definição de tarefa.

Essa definição de tarefa permite a coleta e exportação de métricas do NGINX Plus Prometheus. O contêiner do NGINX rastreia a entrada da aplicação e expõe esses dados à porta 8080, conforme definido em `nginx.conf`. O contêiner do exportador do NGINX Prometheus extrai essas métricas e as publica na porta 9113, para serem usadas no CloudWatch.

Para configurar a definição de tarefa para o exemplo de workload do NGINX no Amazon ECS

1. Crie um arquivo JSON de definição de tarefa com o seguinte conteúdo. Substitua *your-customized-nginx-plus-image* pelo URI de imagem para sua imagem do NGINX Plus personalizada e substitua *your-web-server-app-image* pelo URI de imagem de sua imagem de aplicação do servidor da Web.

```
{
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "your-customized-nginx-plus-image",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "portMappings": [
        {
```

```
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "links": [
        "app",
        "app2"
      ]
    },
    {
      "name": "app",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 128,
      "essential": true
    },
    {
      "name": "app2",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 128,
      "essential": true
    },
    {
      "name": "nginx-prometheus-exporter",
      "image": "docker.io/nginx/nginx-prometheus-exporter:0.8.0",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "command": [
        "-nginx.plus",
        "-nginx.scrape-uri",
         "http://nginx:8080/api"
    ],
    "links":[
      "nginx"
    ],
      "portMappings":[
        {
          "containerPort": 9113,
          "protocol": "tcp"
        }
      ]
    }
```

```
  ],
  "networkMode": "bridge",
  "placementConstraints": [],
  "family": "nginx-plus-sample-stack"
}
```

2. Registre a definição de tarefa:

```
aws ecs register-task-definition --cli-input-json file://path-to-your-task-definition-json
```

3. Crie um serviço para executar a tarefa inserindo o comando a seguir:

```
aws ecs create-service \
 --cluster your-cluster-name \
 --service-name nginx-plus-service \
 --task-definition nginx-plus-sample-stack:1 \
 --desired-count 1
```

Não altere o nome do serviço. Executaremos um serviço de atendente do CloudWatch usando uma configuração que procura tarefas usando os padrões de nome dos serviços que os iniciaram. Por exemplo, para que o atendente do CloudWatch localize a tarefa iniciada por este comando, é possível especificar o valor de `sd_service_name_pattern` como `^nginx-plus-service$`. A próxima seção oferece mais detalhes.

Configurar o atendente do CloudWatch para extrair métricas do NGINX Plus Prometheus

A etapa final é configurar o atendente do CloudWatch para extrair as métricas do NGINX. Neste exemplo, o atendente do CloudWatch detecta a tarefa por meio do padrão de nome de serviço e da porta 9113, onde o exportador expõe as métricas prometheus para o NGINX. Com a tarefa detectada e as métricas disponíveis, o atendente do CloudWatch começa a publicar as métricas coletadas no fluxo de logs nginx-prometheus-exporter.

Para configurar o atendente do CloudWatch para extrair métricas do NGINX

1. Baixe a versão mais recente do arquivo YAML necessário inserindo um dos comandos a seguir.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
```

```
cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
bridge-host.yaml
```

2. Abra o arquivo com um editor de textos e encontre a confguração completa do atendente do CloudWatch na chave `value` da seção `resource:CWAgentConfigSSMParameter`. Depois, na seção `ecs_service_discovery`, adicione a seção `service_name_list_for_tasks` a seguir.

```
"service_name_list_for_tasks": [
  {
    "sd_job_name": "nginx-plus-prometheus-exporter",
    "sd_metrics_path": "/metrics",
    "sd_metrics_ports": "9113",
    "sd_service_name_pattern": "^nginx-plus.*"
   }
],
```

3. No mesmo arquivo, insira a seguinte seção na seção `metric_declaration` para permitir métricas do NGINX Plus. Siga o padrão de recuo existente.

```
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName"]],
  "metric_selectors": [
    "^nginxplus_connections_accepted$",
    "^nginxplus_connections_active$",
    "^nginxplus_connections_dropped$",
    "^nginxplus_connections_idle$",
    "^nginxplus_http_requests_total$",
    "^nginxplus_ssl_handshakes$",
    "^nginxplus_ssl_handshakes_failed$",
    "^nginxplus_up$",
    "^nginxplus_upstream_server_health_checks_fails$"
  ]
},
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName",
 "upstream"]],
  "metric_selectors": [
    "^nginxplus_upstream_server_response_time$"
```

```
  ]
},
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName", "code"]],
  "metric_selectors": [
    "^nginxplus_upstream_server_responses$",
    "^nginxplus_server_zone_responses$"
  ]
},
```

4. Se você ainda não tiver o atendente do CloudWatch implantado nesse cluster, pule para a etapa 8.

   Se já tiver o atendente do CloudWatch implantado no cluster do Amazon ECS usando o AWS CloudFormation, é possível criar um conjunto de alterações inserindo os seguintes comandos:

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name nginx-plus-scraping-support
```

5. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
6. Revisar o changeset recém-criado nginx-plus-scraping-support. Você deverá ver uma alteração aplicada ao recurso CWAgentConfigSSMParameter. Execute o changeset e reinicie a tarefa do atendente do CloudWatch inserindo este comando:

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Aguarde cerca de 10 segundos e insira este comando.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

8. Se você estiver instalando o atendente do CloudWatch com coleta de métricas do Prometheus no cluster pela primeira vez, insira estes comandos.

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Visualizar métricas e logs do NGINX Plus

Agora é possível visualizar as métricas do NGINX Plus que estão sendo coletadas.

Para visualizar as métricas do exemplo de workload do NGINX

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

2. Na região da em que o cluster está em execução, escolha Metrics (Métricas) no painel de navegação à esquerda. Encontre o namespace ContainerInsights/Prometheus para ver as métricas.
3. Para visualizar os eventos do CloudWatch Logs, escolha Log Groups (Grupos de logs) no painel de navegação. Os eventos estão no grupo de logs /aws/containerinsights/***your_cluster_name***/prometheus, no fluxo de logs *nginx-plus-prometheus-exporter*.

Tutorial para adicionar um novo destino de extração do Prometheus: Memcached no Amazon ECS

Este tutorial fornece uma introdução prática para extrair as métricas do Prometheus de uma amostra de aplicação do Memcached em um cluster do Amazon ECS com o tipo de inicialização do EC2. O destino do exportador do Memcached Prometheus será detectado automaticamente pelo atendente do CloudWatch pela detecção de serviços baseada na definição de tarefa do ECS.

O Memcached é um sistema de cache de memória distribuída de uso geral. Geralmente é usado para acelerar sites dinâmicos orientados por banco de dados, armazenando em cache dados e objetos na RAM, a fim de reduzir o número de vezes que uma origem dos dados externa (como um banco de dados ou uma API) precisa ser lida. Para obter mais informações, consulte O que é Memcached?

O memchached_exporter (Licença Apache 2.0) é um dos exportadores oficiais do Prometheus. Por padrão, o memcache_exporter serve na porta 0.0.0.0:9150 em `/metrics`.

As imagens do Docker dos dois repositórios do Docker Hub a seguir são usadas neste tutorial:

- Memcached
- prom/memcached-exporter

Pré-requisito

Para coletar métricas de uma amostra de workload do Prometheus para o Amazon ECS, é necessário estar executando o Container Insights no cluster. Para obter informações sobre como instalar o Container Insights, consulte Configurar o Container Insights no Amazon ECS.

Tópicos

- Definir as variáveis de ambiente de cluster do Amazon ECS EC2

- Instale a amostra de workload do Memcached
- Configurar o atendente do CloudWatch para extrair métricas do Memcached Prometheus
- VIsualizar suas métricas do Memcached

Definir as variáveis de ambiente de cluster do Amazon ECS EC2

Para definir as variáveis de ambiente de cluster do Amazon ECS EC2

1. Instale a CLI do Amazon ECS, caso ainda não tenha instalado. Para obter mais informações, consulte: Instalar a CLI do Amazon ECS.
2. Defina o novo nome do cluster do Amazon ECS e a região. Por exemplo:

```
ECS_CLUSTER_NAME=ecs-ec2-memcached-tutorial
AWS_DEFAULT_REGION=ca-central-1
```

3. (Opcional) Se ainda não tiver um cluster do Amazon ECS com o tipo de inicialização do EC2 no qual deseja instalar a amostra de workload do Memcached e o atendente do CloudWatch, você pode criar um inserindo o comando a seguir.

```
ecs-cli up --capability-iam --size 1 \
--instance-type t3.medium \
--cluster $ECS_CLUSTER_NAME \
--region $AWS_REGION
```

O resultado esperado desse comando é o seguinte:

```
WARN[0000] You will not be able to SSH into your EC2 instances without a key pair.
INFO[0000] Using recommended Amazon Linux 2 AMI with ECS Agent 1.44.4 and Docker
 version 19.03.6-ce
INFO[0001] Created cluster                               cluster=ecs-ec2-memcached-
tutorial region=ca-central-1
INFO[0002] Waiting for your cluster resources to be created...
INFO[0002] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
INFO[0063] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
INFO[0124] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-xxxxxxxxxxxxxxxxx
Security Group created: sg-xxxxxxxxxxxxxxxxx
```

```
Subnet created: subnet-xxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxx
Cluster creation succeeded.
```

Instale a amostra de workload do Memcached

Para instalar a amostra de workload do Memcached que expõe as métricas do Prometheus

1. Baixe o modelo do AWS CloudFormation do Memcached inserindo o comando a seguir.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/sample_traffic/memcached/memcached-traffic-sample.yaml
```

2. Defina os nomes da função do IAM a ser criada para o Memcached inserindo os seguintes comandos.

```
MEMCACHED_ECS_TASK_ROLE_NAME=memcached-prometheus-demo-ecs-task-role-name
MEMCACHED_ECS_EXECUTION_ROLE_NAME=memcached-prometheus-demo-ecs-execution-role-name
```

3. Instale a amostra de workload do Memcached inserindo o comando a seguir. Este exemplo instala a workload no modo de rede `host`.

```
MEMCACHED_ECS_NETWORK_MODE=host

aws cloudformation create-stack --stack-name Memcached-Prometheus-Demo-ECS-
$ECS_CLUSTER_NAME-EC2-$MEMCACHED_ECS_NETWORK_MODE \
    --template-body file://memcached-traffic-sample.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=ECSNetworkMode,ParameterValue=
$MEMCACHED_ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=
$MEMCACHED_ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$MEMCACHED_ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

A pilha do AWS CloudFormation cria quatro recursos:

- Uma função de tarefa do ECS
- Uma função de execução de tarefa do ECS
- Uma definição de tarefa do Memcached
- Um serviço do Memcached

Na definição de tarefa do Memcached, são definidos dois contêineres:

- O contêiner primário executa uma aplicação do Memcached simples e abre a porta 11211 para acesso.
- O outro contêiner executa o processo do exportador do Redis para expor as métricas do Prometheus na porta 9150. É o contêiner a ser detectado e extraído pelo atendente do CloudWatch.

Configurar o atendente do CloudWatch para extrair métricas do Memcached Prometheus

Para configurar o atendente do CloudWatch para extrair métricas do Memcached Prometheus

1. Baixe a versão mais recente do arquivo `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` inserindo o comando a seguir.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml
   ```

2. Abra o arquivo com um editor de textos e encontre a confguração completa do atendente do CloudWatch atrás da chave `value` da seção `resource:CWAgentConfigSSMParameter`.

   Em seguida, na seção `ecs_service_discovery`, adicione a seguinte configuração na seção `task_definition_list`.

   ```
   {
       "sd_job_name": "ecs-memcached",
       "sd_metrics_ports": "9150",
       "sd_task_definition_arn_pattern": ".*:task-definition/memcached-prometheus-demo.*:[0-9]+"
   },
   ```

Para a seção `metric_declaration`, a configuração padrão não permite nenhuma métrica do Memcached. Adicione a seção a seguir para permitir métricas do Memcached. Siga o padrão de recuo existente.

```
{
  "source_labels": ["container_name"],
  "label_matcher": "memcached-exporter-.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily"]],
  "metric_selectors": [
    "^memcached_current_(bytes|items|connections)$",
    "^memcached_items_(reclaimed|evicted)_total$",
    "^memcached_(written|read)_bytes_total$",
    "^memcached_limit_bytes$",
    "^memcached_commands_total$"
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "memcached-exporter-.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily","status","command"],
 ["ClusterName", "TaskDefinitionFamily","command"]],
  "metric_selectors": [
    "^memcached_commands_total$"
  ]
},
```

3. Se já tiver o atendente do CloudWatch implantado no cluster do Amazon ECS com o AWS CloudFormation, é possível criar um conjunto de alterações inserindo os comandos a seguir.

```
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
```

```
                  ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name memcached-scraping-support
```

4. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
5. Revise o changeset recém-criado `memcached-scraping-support`. Você deverá ver uma alteração aplicada ao recurso `CWAgentConfigSSMParameter`. Execute o changeset e reinicie a tarefa do atendente do CloudWatch inserindo os comandos a seguir.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

6. Aguarde cerca de 10 segundos e insira este comando.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Se estiver instalando o atendente do CloudWatch com coleta de métricas do Prometheus no cluster pela primeira vez, insira estes comandos:

```
ECS_NETWORK_MODEE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

VIsualizar suas métricas do Memcached

Este tutorial envia as seguintes métricas ao namespace ECS/ContainerInsights/Prometheus no CloudWatch. É possível usar o console do CloudWatch para ver as métricas nesse namespace.

Nome da métrica	Dimensões	
memcached_current_items	ClusterName , TaskDefinitionFamily	
memcached_current_connections	ClusterName , TaskDefinitionFamily	
memcached_limit_bytes	ClusterName , TaskDefinitionFamily	
memcached_current_bytes	ClusterName , TaskDefinitionFamily	
memcached_written_bytes_total	ClusterName , TaskDefinitionFamily	
memcached_read_bytes_total	ClusterName , TaskDefinitionFamily	
memcached_items_evicted_total	ClusterName , TaskDefinitionFamily	
memcached_items_reclaimed_total	ClusterName , TaskDefinitionFamily	
memcached_commands_total	ClusterName , TaskDefinitionFamily ClusterName , TaskDefinitionFamily, comando	

Nome da métrica	Dimensões	
	`ClusterName` , TaskDefinitionFamily, status, comando	

> **Note**
>
> O valor da dimensão command pode ser: `delete`, `get`, `cas`, `set`, `decr`, `touch`, `incr` ou `flush`.
> O valor da dimensão status pode ser `hit`, `miss` ou `badval`.

Também é possível criar um painel do CloudWatch para suas métricas do Memcached Prometheus.

Para criar um painel para métricas do Memcached Prometheus

1. Crie variáveis de ambiente, substituindo os valores abaixo para corresponder a sua implantação.

```
DASHBOARD_NAME=your_memcached_cw_dashboard_name
ECS_TASK_DEF_FAMILY=memcached-prometheus-demo-$ECS_CLUSTER_NAME-EC2-
$MEMCACHED_ECS_NETWORK_MOD
```

2. Use o comando a seguir para criar o painel.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/ecs-task-definition-templates/deployment-mode/
replica-service/cwagent-prometheus/sample_cloudwatch_dashboards/memcached/
cw_dashboard_memcached.json \
| sed "s/{{YOUR_AWS_REGION}}/$AWS_REGION/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/$ECS_CLUSTER_NAME/g" \
| sed "s/{{YOUR_TASK_DEF_FAMILY}}/$ECS_TASK_DEF_FAMILY/g" \
| xargs -0 aws cloudwatch put-dashboard --dashboard-name ${DASHBOARD_NAME} --region
 $AWS_REGION --dashboard-body
```

Tutorial para extração de métricas do Redis Prometheus no Amazon ECS Fargate

Este tutorial fornece uma introdução prática para extrair as métricas do Prometheus de uma amostra de aplicação do Redis em um cluster do Amazon ECS Fargate. O destino do exportador do Redis

Prometheus será detectado automaticamente pelo atendente do CloudWatch com suporte à métrica do Prometheus com base nos rótulos do docker do contêiner.

O Redis (https://redis.io/) é um armazenamento de estrutura de dados de código aberto (licença BSD) na memória, usado como banco de dados, cache e atendente de mensagens. Para obter mais informações, consulte redis.

redis_exporter (Licença MIT) é usado para expor as métricas do Redis prometheus na porta especificada (padrão: 0.0.0.0:9121). Para obter mais informações, consulte redis_exporter.

As imagens do Docker dos dois repositórios do Docker Hub a seguir são usadas neste tutorial:

- redis
- redis_exporter

Pré-requisito

Para coletar métricas de uma amostra de workload do Prometheus para o Amazon ECS, é necessário estar executando o Container Insights no cluster. Para obter informações sobre como instalar o Container Insights, consulte Configurar o Container Insights no Amazon ECS.

Tópicos

- Definir as variáveis de ambiente de cluster do Amazon ECS Fargate
- Definir as variáveis de ambiente de rede para o cluster do Amazon ECS Fargate
- Instale a workload do Redis
- Configurar o atendente do CloudWatch para extrair métricas do Redis Prometheus
- Exibir métricas do Redis

Definir as variáveis de ambiente de cluster do Amazon ECS Fargate

Para definir as variáveis de ambiente de cluster do Amazon ECS Fargate

1. Instale a CLI do Amazon ECS, caso ainda não tenha instalado. Para obter mais informações, consulte: Instalar a CLI do Amazon ECS.
2. Defina o novo nome do cluster do Amazon ECS e a região. Por exemplo:

```
ECS_CLUSTER_NAME=ecs-fargate-redis-tutorial
```

```
AWS_DEFAULT_REGION=ca-central-1
```

3. (Opcional) Se ainda não tiver um cluster do Amazon ECS Fargate onde deseja instalar a amostra de workload do Redis e o atendente do CloudWatch, você pode criar um inserindo o comando a seguir.

```
ecs-cli up --capability-iam \
--cluster $ECS_CLUSTER_NAME \
--launch-type FARGATE \
--region $AWS_DEFAULT_REGION
```

O resultado esperado desse comando é o seguinte:

```
INFO[0000] Created cluster   cluster=ecs-fargate-redis-tutorial region=ca-central-1
INFO[0001] Waiting for your cluster resources to be created...
INFO[0001] Cloudformation stack status   stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Cluster creation succeeded.
```

Definir as variáveis de ambiente de rede para o cluster do Amazon ECS Fargate

Para definir as variáveis de ambiente de rede para o cluster do Amazon ECS Fargate

1. Defina a VPC e o ID de sub-rede do cluster do Amazon ECS. Se criou um novo cluster no procedimento anterior, você verá esses valores no resultado do comando final. Senão, use os IDs do cluster existente que você usará com o Redis.

```
ECS_CLUSTER_VPC=vpc-xxxxxxxxxxxxxxxxx
ECS_CLUSTER_SUBNET_1=subnet-xxxxxxxxxxxxxxxxx
ECS_CLUSTER_SUBNET_2=subnet-xxxxxxxxxxxxxxxxx
```

2. Neste tutorial, instalaremos a aplicação Redis e o atendente do CloudWatch no grupo de segurança padrão da VPC do cluster do Amazon ECS. O grupo de segurança padrão permite toda a conexão de rede dentro do mesmo grupo de segurança para que o atendente do CloudWatch possa exrair as métricas do Prometheus expostas nos contêineres do Redis. Em um ambiente de produção real, convém criar grupos de segurança dedicados para à aplicação do Redis e ao atendente do CloudWatch e definir permissões personalizadas para eles.

Para obter o ID do grupo de segurança padrão, insira o comando a seguir.

```
aws ec2 describe-security-groups \
--filters Name=vpc-id,Values=$ECS_CLUSTER_VPC  \
--region $AWS_DEFAULT_REGION
```

Em seguida, defina a variável do grupo de segurança padrão do cluster Fargate inserindo o seguinte comando, substituindo *my-default-security-group* pelo valor que você encontrou no comando anterior.

```
ECS_CLUSTER_SECURITY_GROUP=my-default-security-group
```

Instale a workload do Redis

Para instalar a amostra de workload do Redis que expõe as métricas do Prometheus

1. Baixe o modelo do AWS CloudFormation do Redis inserindo o comando a seguir.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/sample_traffic/redis/redis-traffic-sample.yaml
```

2. Defina os nomes da função do IAM a ser criada para o Redis inserindo os seguintes comandos.

```
REDIS_ECS_TASK_ROLE_NAME=redis-prometheus-demo-ecs-task-role-name
REDIS_ECS_EXECUTION_ROLE_NAME=redis-prometheus-demo-ecs-execution-role-name
```

3. Instale a amostra de workload do Redis inserindo o comando a seguir.

```
aws cloudformation create-stack --stack-name Redis-Prometheus-Demo-ECS-
$ECS_CLUSTER_NAME-fargate-awsvpc \
    --template-body file://redis-traffic-sample.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET_1 \
                 ParameterKey=TaskRoleName,ParameterValue=$REDIS_ECS_TASK_ROLE_NAME
 \
```

```
                  ParameterKey=ExecutionRoleName,ParameterValue=
$REDIS_ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_DEFAULT_REGION
```

A pilha do AWS CloudFormation cria quatro recursos:

- Uma função de tarefa do ECS
- Uma função de execução de tarefa do ECS
- Uma definição de tarefa do Redis
- Um serviço do Redis

Na definição de tarefa do Redis, são definidos dois contêineres:

- O contêiner primário executa uma aplicação do Redis simples e abre a porta 6379 para acesso.
- O outro contêiner executa o processo do exportador do Redis para expor as métricas do Prometheus na porta 9121. É o contêiner a ser detectado e extraído pelo atendente do CloudWatch. O rótulo do docker a seguir é definido para que o atendente do CloudWatch possa detectar esse contêiner com base nele.

```
ECS_PROMETHEUS_EXPORTER_PORT: 9121
```

Configurar o atendente do CloudWatch para extrair métricas do Redis Prometheus

Para configurar o atendente do CloudWatch para extrair métricas do Redis Prometheus

1. Baixe a versão mais recente do arquivo `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` inserindo o comando a seguir.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
awsvpc.yaml
```

2. Abra o arquivo com um editor de textos e encontre a confguração completa do atendente do CloudWatch atrás da chave `value` da seção `resource:CWAgentConfigSSMParameter`.

Em seguida, na seção `ecs_service_discovery` mostrada aqui, a detecção de serviços baseado em `docker_label` está habilitada com as configurações padrão que são baseadas em `ECS_PROMETHEUS_EXPORTER_PORT`, que corresponde ao rótulo do docker definido na definição de tarefa do Redis ECS. Portanto, não precisamos fazer alterações nesta seção:

```
ecs_service_discovery": {
  "sd_frequency": "1m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
*  "docker_label": {
  },*
  ...
```

Para a seção `metric_declaration`, a configuração padrão não permite nenhuma métrica do Redis. Adicione a seção a seguir para permitir métricas do Redis. Siga o padrão de recuo existente.

```
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily"]],
  "metric_selectors": [
    "^redis_net_(in|out)put_bytes_total$",
    "^redis_(expired|evicted)_keys_total$",
    "^redis_keyspace_(hits|misses)_total$",
    "^redis_memory_used_bytes$",
    "^redis_connected_clients$"
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily","cmd"]],
  "metric_selectors": [
    "^redis_commands_total$"
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily","db"]],
  "metric_selectors": [
```

```
    "^redis_db_keys$"
  ]
},
```

3. Se já tiver o atendente do CloudWatch implantado no cluster do Amazon ECS com o AWS CloudFormation, é possível criar um conjunto de alterações inserindo os comandos a seguir.

```
ECS_LAUNCH_TYPE=FARGATE
CREATE_IAM_ROLES=True
ECS_CLUSTER_SUBNET=$ECS_CLUSTER_SUBNET_1
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
$ECS_CLUSTER_NAME-$ECS_LAUNCH_TYPE-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSLaunchType,ParameterValue=$ECS_LAUNCH_TYPE \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --change-set-name redis-scraping-support
```

4. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.

5. Revise o changeset recém-criado `redis-scraping-support`. Você deverá ver uma alteração aplicada ao recurso `CWAgentConfigSSMParameter`. Execute o changeset e reinicie a tarefa do atendente do CloudWatch inserindo os comandos a seguir.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE-awsvpc \
--region ${AWS_DEFAULT_REGION}
```

6. Aguarde cerca de 10 segundos e insira este comando.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
```

```
--desired-count 1 \
--service cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE-awsvpc \
--region ${AWS_DEFAULT_REGION}
```

7. Se estiver instalando o atendente do CloudWatch com coleta de métricas do Prometheus no cluster pela primeira vez, insira estes comandos:

```
ECS_LAUNCH_TYPE=FARGATE
CREATE_IAM_ROLES=True
ECS_CLUSTER_SUBNET=$ECS_CLUSTER_SUBNET_1
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
$ECS_CLUSTER_NAME-$ECS_LAUNCH_TYPE-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSLaunchType,ParameterValue=$ECS_LAUNCH_TYPE \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION}
```

Exibir métricas do Redis

Este tutorial envia as seguintes métricas ao namespace ECS/ContainerInsights/Prometheus no CloudWatch. É possível usar o console do CloudWatch para ver as métricas nesse namespace.

Nome da métrica	Dimensões	
`redis_net_input_bytes_total`	ClusterName, `TaskDefinitionFamily`	

Nome da métrica	Dimensões	
redis_net_output_bytes_total	ClusterName, TaskDefinitionFamily	
redis_expired_keys_total	ClusterName, TaskDefinitionFamily	
redis_evicted_keys_total	ClusterName, TaskDefinitionFamily	
redis_keyspace_hits_total	ClusterName, TaskDefinitionFamily	
redis_keyspace_misses_total	ClusterName, TaskDefinitionFamily	
redis_memory_used_bytes	ClusterName, TaskDefinitionFamily	
redis_connected_clients	ClusterName, TaskDefinitionFamily	
redis_commands_total	ClusterName , TaskDefinitionFamily , cmd	
redis_db_keys	ClusterName , TaskDefinitionFamily , db	

> **Note**
>
> O valor da dimensão cmd pode ser: append, client, command, config, dbsize, flushall, get, incr, info, latency ou slowlog.

Os valores da dimensão db podem ser `db0` ou `db15`.

Também é possível criar um painel do CloudWatch para suas métricas do Redis Prometheus.

Para criar um painel para métricas do Redis Prometheus

1. Crie variáveis de ambiente, substituindo os valores abaixo para corresponder a sua implantação.

```
DASHBOARD_NAME=your_cw_dashboard_name
ECS_TASK_DEF_FAMILY=redis-prometheus-demo-$ECS_CLUSTER_NAME-fargate-awsvpc
```

2. Use o comando a seguir para criar o painel.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/redis/cw_dashboard_redis.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| sed "s/{{YOUR_NAMESPACE}}/${NAMESPACE}/g" \
```

Instalar e configurar a coleta de métricas do Prometheus em clusters do Amazon EKS e do Kubernetes

Para coletar métricas do Prometheus de clusters que executam o Amazon EKS ou Kubernetes, é possível usar o atendente do CloudWatch como coletor ou usar o coletor do AWS Distro for OpenTelemetry. Para obter informações sobre como usar o coletor do AWS Distro for OpenTelemetry, consulte https://aws-otel.github.io/docs/getting-started/container-insights/eks-prometheus.

As seções a seguir explicam como coletar métricas do Prometheus usando o atendente do CloudWatch. Elas explicam como instalar o atendente do CloudWatch com o monitoramento do Prometheus em clusters que executam o Amazon EKS e o Kubernetes e como configurar o atendente para extrair outros destinos. Estas seções também fornecem tutoriais opcionais para configurar amostras de workloads para usar testes com monitoramento do Prometheus.

Tópicos

- Instalar o atendente do CloudWatch com a coleção de métricas do Prometheus em clusters do Amazon EKS e do Kubernetes

Instalar o atendente do CloudWatch com a coleção de métricas do Prometheus em clusters do Amazon EKS e do Kubernetes

Esta seção explica como configurar o atendente do CloudWatch com monitoramento do Prometheus em um cluster que está executando o Amazon EKS ou o Kubernetes. Depois que você fizer isso, o atendente automaticamente extrairá e importará métricas para as seguintes workloads em execução nesse cluster.

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy
- Fluent Bit

Também é possível configurar o atendente para extrair e importar outras workloads e origens do Prometheus.

Antes de seguir estas etapas de instalação do atendente do CloudWatch para coleta de métricas do Prometheus, você deve ter um cluster em execução no Amazon EKS ou um cluster do Kubernetes em execução em uma instância do Amazon EC2.

Requisitos para grupo de segurança de VPC

As regras de entrada dos grupos de segurança para as workloads do Prometheus devem abrir as portas do Prometheus para o atendente do CloudWatch para extrair as métricas Prometheus pelo IP privado.

As regras de saída do grupo de segurança do atendente do CloudWatch devem permitir que o atendente do CloudWatch se conecte à porta das workloads do Prometheus por IP privado.

Tópicos

Instalar o atendente do CloudWatch com a coleção de métricas do Prometheus em clusters do Amazon EKS e do Kubernetes

Esta seção explica como configurar o atendente do CloudWatch com monitoramento do Prometheus em um cluster que está executando o Amazon EKS ou o Kubernetes. Depois que você fizer isso, o atendente automaticamente extrairá e importará métricas para as seguintes workloads em execução nesse cluster.

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy
- Fluent Bit

Também é possível configurar o atendente para extrair e importar outras workloads e origens do Prometheus.

Antes de seguir estas etapas de instalação do atendente do CloudWatch para coleta de métricas do Prometheus, você deve ter um cluster em execução no Amazon EKS ou um cluster do Kubernetes em execução em uma instância do Amazon EC2.

Requisitos para grupo de segurança de VPC

As regras de entrada dos grupos de segurança para as workloads do Prometheus devem abrir as portas do Prometheus para o atendente do CloudWatch para extrair as métricas Prometheus pelo IP privado.

As regras de saída do grupo de segurança do atendente do CloudWatch devem permitir que o atendente do CloudWatch se conecte à porta das workloads do Prometheus por IP privado.

Tópicos

Configurar funções do IAM

A primeira etapa é configurar a função do IAM necessária no cluster. Há dois métodos:

- Configurar uma função do IAM para uma conta de serviço, também conhecida como função de serviço. Esse método funciona tanto para o tipo de inicialização do EC2 como para o tipo de inicialização do Fargate.
- Adicione uma política do IAM à função do IAM usada para o cluster. Isso funciona apenas para o tipo de inicialização do EC2.

Configurar uma função de serviço (tipo de inicialização do EC2 e tipo de inicialização do Fargate)

Para configurar uma função de serviço, insira o comando a seguir. Substitua *MyCluster* pelo nome do cluster.

```
eksctl create iamserviceaccount \
 --name cwagent-prometheus \
--namespace amazon-cloudwatch \
 --cluster MyCluster \
--attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
--approve \
--override-existing-serviceaccounts
```

Adicione uma política à função do IAM do cluster (somente tipo de inicialização do EC2)

Como configurar a política do IAM em um cluster para suporte ao Prometheus

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, escolha Instances (Instâncias).
3. Você precisa encontrar o prefixo do nome da função do IAM para o cluster. Para isso, marque a caixa de seleção ao lado do nome de uma instância que está no cluster e escolha Actions (Ações), Instance Settings (Configurações da instância), Attach/Replace IAM Role (Anexar/substituir função do IAM). Copie o prefixo da função do IAM, como `eksctl-dev303-workshop-nodegroup`.
4. Abra o console do IAM em https://console.aws.amazon.com/iam/.
5. No painel de navegação, escolha Perfis.
6. Use a caixa de pesquisa para localizar o prefixo copiado anteriormente neste procedimento e escolha a função.
7. Escolha Anexar políticas.
8. Use a caixa de pesquisa para localizar o CloudWatchAgentServerPolicy. Marque a caixa de seleção ao lado de CloudWatchAgentServerPolicy e escolha Attach policy (Anexar política).

Instalar o atendente do CloudWatch para coletar métricas do Prometheus

Você deve instalar o atendente do CloudWatch no cluster para coletar as métricas. O método de instalação do atendente é diferente para clusters do Amazon EKS e clusters do Kubernetes.

Excluir versões anteriores do atendente do CloudWatch compatíveis com Prometheus

Se você já instalou em seu cluster uma versão do atendente do CloudWatch com suporte ao Prometheus, exclua essa versão inserindo o comando a seguir. Isso é necessário apenas para versões anteriores do atendente com suporte ao Prometheus. Não é necessário excluir o atendente do CloudWatch que habilita o Container Insights sem suporte ao Prometheus.

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

Instalar o atendente CloudWatch em clusters do Amazon EKS com o tipo de inicialização do EC2

Para instalar o atendente do CloudWatch com suporte ao Prometheus em um cluster do Amazon EKS, siga estas etapas.

Para instalar o atendente do CloudWatch com suporte ao Prometheus em um cluster do Amazon EKS

1. Insira o comando a seguir para verificar se o namespace `amazon-cloudwatch` já foi criado:

   ```
   kubectl get namespace
   ```

2. Se `amazon-cloudwatch` não for exibido nos resultados, crie-o inserindo o comando a seguir:

   ```
   kubectl create namespace amazon-cloudwatch
   ```

3. Para implantar o atendente com a configuração padrão e fazer com que ele envie dados para a região da AWS em que está instalado, insira o comando a seguir:

   ```
   kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
   container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
   service/cwagent-prometheus/prometheus-eks.yaml
   ```

   Em vez disso, para que o atendente envie dados para uma região diferente, siga estas etapas:

   a. Faça download do arquivo YAML para o atendente inserindo o comando a seguir:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
service/cwagent-prometheus/prometheus-eks.yaml
```

b. Abra o arquivo com um editor de textos e procure o bloco `cwagentconfig.json` do arquivo.

c. Adicione as linhas realçadas, especificando a região desejada:

```
cwagentconfig.json: |
    {
      "agent": {
        "region": "us-east-2"
      },
      "logs": { ...
```

d. Salve o arquivo e implante o atendente usando o arquivo atualizado.

```
kubectl apply -f prometheus-eks.yaml
```

Instalar o atendente CloudWatch em clusters do Amazon EKS com o tipo de inicialização do Fargate

Para instalar o atendente do CloudWatch com suporte ao Prometheus em um cluster do Amazon EKS com tipo de inicialização do Fargate, siga estas etapas.

Para instalar o atendente do CloudWatch com suporte ao Prometheus em um cluster do Amazon EKS com tipo de inicialização do Fargate

1. Insira o comando a seguir para criar um perfil Fargate para o atendente do CloudWatch de modo que ele possa ser executado dentro do cluster. Substitua *MyCluster* pelo nome do cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--name amazon-cloudwatch \
--namespace amazon-cloudwatch
```

2. Para instalar o atendente do CloudWatch, insira o comando a seguir. Substitua *MyCluster* pelo nome do cluster. Esse nome é usado no nome do grupo de logs que armazena os eventos de log coletados pelo atendente, além de ser usado como uma dimensão para as métricas coletadas pelo atendente.

Substitua a *region* pelo nome da região para onde você deseja que as métricas sejam enviadas. Por exemplo, `us-west-1`.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-eks-fargate.yaml |
sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" |
kubectl apply -f -
```

Instalar o atendente do CloudWatch em um cluster do Kubernetes

Para instalar o atendente do CloudWatch com suporte ao Prometheus em um cluster do Kubernetes em execução, insira o comando a seguir:

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-k8s.yaml |
sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" |
kubectl apply -f -
```

Substitua *MyCluster* pelo nome do cluster. Esse nome é usado no nome do grupo de logs que armazena os eventos de log coletados pelo atendente, além de ser usado como uma dimensão para as métricas coletadas pelo atendente.

Substitua a *região* pelo nome da região da AWS para onde você deseja que as métricas sejam enviadas. Por exemplo, **us-west-1**.

Verificar se o atendente está em execução

Nos clusters do Amazon EKS e do Kubernetes, você pode inserir o seguinte comando para confirmar se o atendente está em execução.

```
kubectl get pod -l "app=cwagent-prometheus" -n amazon-cloudwatch
```

Se os resultados incluírem um único pod de atendente do CloudWatch no estado `Running`, o atendente está em execução e coletando métricas do Prometheus. Por padrão, o atendente do CloudWatch coleta métricas para App Mesh, NGINX, Memcached, Java/JMX e HAProxy a cada minuto. Para obter mais informações sobre essas métricas, consulte Métricas do Prometheus

coletadas pelo atendente do CloudWatch. Para obter instruções sobre como ver as métricas do Prometheus no CloudWatch, consulte Visualizar as métricas do Prometheus

Você também pode configurar o atendente do CloudWatch para coletar métricas de outros exportadores do Prometheus. Para ter mais informações, consulte Extrair outras fontes do Prometheus e importar essas métricas.

Extrair outras fontes do Prometheus e importar essas métricas

O atendente do CloudWatch com monitoramento Prometheus precisa de duas configurações para extrair as métricas do Prometheus. Uma serve para as configurações padrão do Prometheus, conforme documentado em <scrape_config> na documentação do Prometheus. A outra é para a configuração do atendente do CloudWatch.

Para clusters do Amazon EKS, as configurações são definidas em`prometheus-eks.yaml` (para o tipo de inicialização EC2) ou `prometheus-eks-fargate.yaml` (para o tipo de inicialização do Fargate) como dois mapas de configuração:

- A seção `name: prometheus-config` contém as configurações para extração de conteúdo do Prometheus.
- A seção `name: prometheus-cwagentconfig` contém a configuração para o atendente do CloudWatch. Você pode usar esta seção para configurar como as métricas do Prometheus são coletadas pelo CloudWatch. Por exemplo, você pode especificar quais métricas devem ser importadas ao CloudWatch e definir suas dimensões.

Para clusters do Kubernetes em execução em instâncias do Amazon EC2, as configurações são definidas no arquivo YAML `prometheus-k8s.yaml` como dois mapas de configuração:

- A seção `name: prometheus-config` contém as configurações para extração de conteúdo do Prometheus.
- A seção `name: prometheus-cwagentconfig` contém a configuração para o atendente do CloudWatch.

Para extrair outras origens de métricas do Prometheus e importar essas métricas para o CloudWatch, modifique a configuração de extração do Prometheus e a configuração do atendente do CloudWatch e implante novamente o atendente com a configuração atualizada.

Requisitos para grupo de segurança de VPC

As regras de entrada dos grupos de segurança para as workloads do Prometheus devem abrir as portas do Prometheus para o atendente do CloudWatch para extrair as métricas Prometheus pelo IP privado.

As regras de saída do grupo de segurança do atendente do CloudWatch devem permitir que o atendente do CloudWatch se conecte à porta das workloads do Prometheus por IP privado.

Configuração de extração do Prometheus

O atendente do CloudWatch oferece suporte às configurações de extração padrão do Prometheus, conforme documentado em <scrape_config> na documentação do Prometheus. É possível editar essa seção para atualizar as configurações que já estão nesse arquivo e adicionar outros destinos de extração do Prometheus. Por padrão, um exemplo de arquivo de configuração contém as seguintes linhas de configuração global:

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
```

- scrape_interval: define a frequência da adição de destinos de extração de conteúdo.
- scrape_timeout: define quanto tempo aguardar até a expiração de uma solicitação de extração de conteúdo.

Também é possível definir valores diferentes para essas configurações no nível do trabalho, a fim de substituir as configurações globais.

Trabalhos de extração do Prometheus

Os arquivos YAML do atendente do CloudWatch já têm alguns trabalhos padrão de extração configurados. Por exemplo, em `prometheus-eks.yaml`, os trabalhos de extração padrão são configurados nas linhas `job_name` da seção `scrape_configs`. Nesse arquivo, a seguinte seção padrão `kubernetes-pod-jmx` extrai métricas do JMX Exporter.

```
   - job_name: 'kubernetes-pod-jmx'
      sample_limit: 10000
      metrics_path: /metrics
      kubernetes_sd_configs:
      - role: pod
      relabel_configs:
      - source_labels: [__address__]
```

```
      action: keep
      regex: '.*:9404$'
    - action: labelmap
      regex: __meta_kubernetes_pod_label_(.+)
    - action: replace
      source_labels:
      - __meta_kubernetes_namespace
      target_label: Namespace
    - source_labels: [__meta_kubernetes_pod_name]
      action: replace
      target_label: pod_name
    - action: replace
      source_labels:
      - __meta_kubernetes_pod_container_name
      target_label: container_name
    - action: replace
      source_labels:
      - __meta_kubernetes_pod_controller_name
      target_label: pod_controller_name
    - action: replace
      source_labels:
      - __meta_kubernetes_pod_controller_kind
      target_label: pod_controller_kind
    - action: replace
      source_labels:
      - __meta_kubernetes_pod_phase
      target_label: pod_phase
```

Cada um desses destinos padrão é extraído e as métricas são enviadas ao CloudWatch em eventos de log usando o formato de métricas incorporado. Para ter mais informações, consulte Incorporação de métricas em logs.

Os eventos de log dos clusters do Amazon EKS e do Kubernetes são armazenados no grupo de logs /aws/containerinsights/***cluster_name***/prometheus no CloudWatch Logs. Os eventos de log dos clusters do Amazon ECS são armazenados no grupo de logs /aws/ecs/containerinsights/***cluster_name***/prometheus.

Cada trabalho de extração está contido em um stream de log diferente nesse grupo de logs. Por exemplo, o trabalho de extração do Prometheus `kubernetes-pod-appmesh-envoy` é definido para o App Mesh. Todas as métricas do App Mesh Prometheus de clusters do Amazon EKS e do Kubernetes são enviadas ao fluxo de logs chamado /aws/containerinsights/***cluster_name***>prometheus/kubernetes-pod-appmesh-envoy/.

Para adicionar um novo destino de extração, adicione uma nova seção `job_name` à seção `scrape_configs` do arquivo YAML e reinicie o atendente. Para obter um exemplo desse processo, consulte Tutorial para adicionar um novo destino de extração do Prometheus: métricas do servidor de API do Prometheus.

Configuração do atendente do CloudWatch para o Prometheus

O arquivo de configuração do atendente do CloudWatch tem uma seção `prometheus` na seção `metrics_collected` para a configuração de extração do Prometheus. Contém as seguintes opções de configuração:

- cluster_name: especifica o nome do cluster a ser adicionado como um rótulo no evento de log. Esse campo é opcional. Se você omitir, o atendente poderá detectar o nome do cluster do Amazon EKS ou do Kubernetes.
- log_group_name: especifica o nome do grupo de log para as métricas do Prometheus extraídas. Esse campo é opcional. Se você omitir, o CloudWatch usará /aws/containerinsights/***`cluster_name`***/prometheus para logs de clusters do Amazon EKS e do Kubernetes.
- prometheus_config_path: especifica o caminho do arquivo de configuração de extração do Prometheus. Se o valor desse campo começar com `env:`, o conteúdo do arquivo de configuração de extração do Prometheus será recuperado da variável de ambiente do contêiner. Não altere esse campo.
- ecs_service_discovery: é a seção para especificar a configuração da detecção de serviço do Amazon ECS Prometheus. Para ter mais informações, consulte Guia detalhado para detecção automática em clusters do Amazon ECS.

  A seção `ecs_service_discovery` pode conter os seguintes campos:

  - `sd_frequency` é a frequência para detectar os exportadores Prometheus. Especifique um número e um sufixo de unidade. Por exemplo, `1m` uma vez por minuto ou `30s` uma vez a cada 30 segundos. Os sufixos de unidade válidos são: `ns`, `us`, `ms`, `s`, `m` e `h`.

    Esse campo é opcional. O padrão é 60 segundos (1 minuto).
  - `sd_target_cluster` é o nome do cluster do Amazon ECS de destino para detecção automática. Esse campo é opcional. O padrão é o nome do cluster do Amazon ECS em que o atendente do CloudWatch está instalado.

- `sd_cluster_region` é a região do cluster do Amazon ECS de destino. Esse campo é opcional. O padrão é a região do cluster do Amazon ECS em que o atendente do CloudWatch está instalado.
- `sd_result_file` é o caminho do arquivo YAML para os resultados de destino do Prometheus. A configuração de extração do Prometheus referenciará esse arquivo.
- `docker_label` é uma seção opcional que você pode usar para especificar a configuração para detecção de serviço baseada em rótulos do docker. Se você omitir essa seção, a detecção baseada em rótulos do docker não será usada. A seção pode conter os seguintes campos:
  - `sd_port_label` é o nome do rótulo do docker do contêiner que especifica a porta do contêiner para métricas do Prometheus. O valor padrão é `ECS_PROMETHEUS_EXPORTER_PORT`. Se o contêiner não tiver esse rótulo do docker, o atendente do CloudWatch o ignorará.
  - `sd_metrics_path_label` é o nome do rótulo do docker do contêiner que especifica o caminho das métricas do Prometheus. O valor padrão é `ECS_PROMETHEUS_METRICS_PATH`. Se o contêiner não tiver esse rótulo do docker, o agente assumirá o caminho padrão `/metrics`.
  - `sd_job_name_label` é o nome do rótulo do docker do contêiner que especifica o nome do trabalho de extração do Prometheus. O valor padrão é `job`. Se o contêiner não tiver esse rótulo do docker, o atendente do CloudWatch usará o nome do trabalho na configuração de extração do Prometheus.
- `task_definition_list` é uma seção opcional que você pode usar para especificar a configuração para detectção de serviço baseada em definição de tarefa. Se você omitir essa seção, a detecção baseada em definição de tarefa não será usada. A seção pode conter os seguintes campos:
  - `sd_task_definition_arn_pattern` é o padrão a ser usado para especificar as definições de tarefa do Amazon ECS a serem detectadas. Essa é uma expressão regular.
  - `sd_metrics_ports` lista a containerPort para as métricas do Prometheus. Separe as containerPorts com ponto e vírgula.
  - `sd_container_name_pattern` especifica os nomes de contêiner de tarefas do Amazon ECS. Essa é uma expressão regular.
  - `sd_metrics_path` especifica o caminho da métrica do Prometheus. Se você omitir isso, o atendente assumirá o caminho padrão `/metrics`

    - `sd_job_name` especifica o nome do trabalho de extração do Prometheus. Se você omitir esse campo, o atendente do CloudWatch usará o nome do trabalho na configuração de extração do Prometheus.

- metric_declaration: são seções que especificam a matriz de logs com formato de métrica incorporado a ser gerada. Há seções `metric_declaration` para cada destino do Prometheus do qual o atendente do CloudWatch importa por padrão. Essas seções incluem os seguintes campos:
    - `label_matcher` é uma expressão regular que confere o valor dos rótulos listados em `source_labels`. As métricas correspondentes são disponibilizadas para inclusão no formato de métrica incorporado enviado ao CloudWatch.

      Se você tiver vários rótulos especificados em `source_labels`, recomendamos não utilizar os caracteres ^ ou $ na expressão regular para `label_matcher`.
    - `source_labels` especifica o valor dos rótulos verificados pela linha `label_matcher`.
    - `label_separator` especifica o separador a ser usado na linha `label_matcher` se vários `source_labels` forem especificados. O padrão é `;`. É possível ver esse padrão usado na linha `label_matcher` no exemplo a seguir.
    - `metric_selectors` é uma expressão regular que especifica as métricas a serem coletadas e enviadas ao CloudWatch.
    - `dimensions` é a lista de rótulos a serem usados como dimensões do CloudWatch para cada métrica selecionada.

Veja o exemplo de `metric_declaration` a seguir.

```
"metric_declaration": [
  {
     "source_labels":[ "Service", "Namespace"],
     "label_matcher":"(.*node-exporter.*|.*kube-dns.*);kube-system",
     "dimensions":[
        ["Service", "Namespace"]
     ],
     "metric_selectors":[
        "^coredns_dns_request_type_count_total$"
     ]
  }
]
```

Esse exemplo configura uma seção de formato de métrica incorporada a ser enviada como um evento de log se as seguintes condições forem atendidas:

- O valor de `Service` contém `node-exporter` ou `kube-dns`.
- O valor de `Namespace` é `kube-system`.
- A métrica do Prometheus `coredns_dns_request_type_count_total` contém rótulos `Namespace` e `Service`.

O evento de log enviado inclui a seguinte seção destacada:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"coredns_dns_request_type_count_total"
            }
         ],
         "Dimensions":[
            [
               "Namespace",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "Namespace":"kube-system",
   "Service":"kube-dns",
   "coredns_dns_request_type_count_total":2562,
   "eks_amazonaws_com_component":"kube-dns",
   "instance":"192.168.61.254:9153",
   "job":"kubernetes-service-endpoints",
   ...
}
```

Tutorial para adicionar um novo destino de extração do Prometheus: métricas do servidor de API do Prometheus

O servidor de API do Kubernetes expõe métricas do Prometheus em endpoints por padrão. O exemplo oficial da configuração de extração do servidor de API do Kubernetes está disponível no Github.

O tutorial a seguir mostra como executar as seguintes etapas para começar a importar métricas do servidor de API do Kubernetes para o CloudWatch:

- Como adicionar a configuração de extração do Prometheus para o servidor de API do Kubernetes ao arquivo YAML do atendente do CloudWatch.
- Como configurar as definições de métricas em formato de métrica incorporada no arquivo YAML do atendente do CloudWatch.
- (Opcional) Como criar um painel do CloudWatch para as métricas do servidor de API do Kubernetes.

> **Note**
>
> O servidor de API do Kubernetes expõe métricas de indicador, contador, histograma e resumo. Nesta versão do suporte a métricas do Prometheus, o CloudWatch importa apenas as métricas com tipos de indicador e contador.

Como começar a coletar métricas do servidor de API do Kubernetes do Prometheus no CloudWatch

1. Baixe a versão mais recente do arquivo `prometheus-eks.yaml`, `prometheus-eks-fargate.yaml` ou `prometheus-k8s.yaml` inserindo um dos comandos a seguir.

   Para um cluster do Amazon EKS com o tipo de inicialização do EC2, insira o comando a seguir:

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
   ```

   Para um cluster do Amazon EKS com o tipo de inicialização do Fargate, insira o comando a seguir:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-eks-fargate.yaml
```

Para um cluster do Kubernetes em execução em uma instância do Amazon EC2, insira o comando a seguir:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-k8s.yaml
```

2. Abra o arquivo com um editor de texto, localize a seção `prometheus-config` e adicione a seção a seguir nela. Salve as alterações:

```
    # Scrape config for API servers
    - job_name: 'kubernetes-apiservers'
      kubernetes_sd_configs:
        - role: endpoints
          namespaces:
            names:
              - default
      scheme: https
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        insecure_skip_verify: true
      bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
      relabel_configs:
      - source_labels: [__meta_kubernetes_service_name,
 __meta_kubernetes_endpoint_port_name]
        action: keep
        regex: kubernetes;https
      - action: replace
        source_labels:
        - __meta_kubernetes_namespace
        target_label: Namespace
      - action: replace
        source_labels:
        - __meta_kubernetes_service_name
        target_label: Service
```

3. Enquanto o arquivo YAML ainda está aberto no editor de texto, encontre a seção `cwagentconfig.json`. Adicione a seguinte subseção e salve as alterações. Esta seção coloca as métricas do servidor de API na lista de permissões do atendente do CloudWatch. Três tipos de métricas do servidor de API são adicionados à lista de permissões:

   - contagens de objetos etcd
   - Métricas do controlador de registro do servidor de API
   - Métricas de solicitação do servidor de API

```
{"source_labels": ["job", "resource"],
  "label_matcher": "^kubernetes-apiservers;(services|daemonsets.apps|
deployments.apps|configmaps|endpoints|secrets|serviceaccounts|replicasets.apps)",
  "dimensions": [["ClusterName","Service","resource"]],
  "metric_selectors": [
  "^etcd_object_counts$"
  ]
},
{"source_labels": ["job", "name"],
   "label_matcher": "^kubernetes-apiservers;APIServiceRegistrationController$",
   "dimensions": [["ClusterName","Service","name"]],
   "metric_selectors": [
   "^workqueue_depth$",
   "^workqueue_adds_total$",
   "^workqueue_retries_total$"
  ]
},
{"source_labels": ["job","code"],
  "label_matcher": "^kubernetes-apiservers;2[0-9]{2}$",
  "dimensions": [["ClusterName","Service","code"]],
  "metric_selectors": [
   "^apiserver_request_total$"
  ]
},
{"source_labels": ["job"],
  "label_matcher": "^kubernetes-apiservers",
  "dimensions": [["ClusterName","Service"]],
  "metric_selectors": [
  "^apiserver_request_total$"
  ]
},
```

4. Se você já tem o atendente do CloudWatch com suporte ao Prometheus implantado no cluster, exclua-o inserindo o comando a seguir:

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

5. Implante o atendente do CloudWatch com a configuração atualizada inserindo um dos comandos a seguir. Para um cluster do Amazon EKS com o tipo de inicialização do EC2, insira:

```
kubectl apply -f prometheus-eks.yaml
```

Para um cluster do Amazon EKS com o tipo de inicialização do Fargate, insira o comando a seguir. Substitua *MyCluster* e *region* com valores para corresponder a sua implantação.

```
cat prometheus-eks-fargate.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Para um cluster do Kubernetes, insira o comando a seguir. Substitua *MyCluster* e *region* com valores para corresponder a sua implantação.

```
cat prometheus-k8s.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Uma vez feito isso, será necessário ver um novo stream de log chamado kubernetes-apiservers no grupo de log /aws/containerinsights/***cluster_name***/prometheus. Esse fluxo de logs deve incluir eventos de log com uma definição de formato de métrica incorporada como a seguinte:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"apiserver_request_total"
            }
         ],
         "Dimensions":[
            [
               "ClusterName",
```

```
                "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "ClusterName":"my-cluster-name",
   "Namespace":"default",
   "Service":"kubernetes",
   "Timestamp":"1592267020339",
   "Version":"0",
   "apiserver_request_count":0,
   "apiserver_request_total":0,
   "code":"0",
   "component":"apiserver",
   "contentType":"application/json",
   "instance":"192.0.2.0:443",
   "job":"kubernetes-apiservers",
   "prom_metric_type":"counter",
   "resource":"pods",
   "scope":"namespace",
   "verb":"WATCH",
   "version":"v1"
}
```

Você pode visualizar suas métricas no console do CloudWatch no namespace ContainerInsights/Prometheus. Também é possível criar um painel do CloudWatch para as métricas do servidor de API do Kubernetes do Prometheus.

(Opcional) Criar um painel para as métricas do servidor de API do Kubernetes

Para ver as métricas do servidor de API do Kubernetes em seu painel, é necessário ter concluído primeiro as etapas nas seções anteriores para começar a coletar essas métricas no CloudWatch.

Como criar um painel para métricas do servidor de API do Kubernetes

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Verifique se você selecionou a região da AWS correta.
3. No painel de navegação, escolha Painéis.
4. Escolha Create dashboard (Criar painel). Insira um nome para o novo painel e escolha Create dashboard (Criar painel).

5. Em Add to this dashboard (Adicionar a este painel), escolha Cancel (Cancelar).
6. Escolha Actions (Ações), View/edit source (Exibir/editar origem).
7. Faça o download do seguinte arquivo JSON: Origem do painel da API do Kubernetes.
8. Abra o arquivo JSON obtido por download com um editor de textos e faça as seguintes alterações:
   - Substitua todas as strings `{{YOUR_CLUSTER_NAME}}` pelo nome exato do cluster. Não adicione espaços em branco antes ou depois do texto.
   - Substitua todas as strings `{{YOUR_AWS_REGION}}` pelo nome da região onde as métricas são coletadas. Por exemplo, `us-west-2`. Não adicione espaços em branco antes ou depois do texto.
9. Copie todo o blob JSON e cole-o na caixa de texto no console do CloudWatch, substituindo o que já está na caixa.
10. Escolha Update (Atualizar), Save dashboard (Salvar painel).

(Opcional) Configurar workloads de exemplo do Amazon EKS em contêineres para teste de métrica do Prometheus

Para testar o suporte para métricas do Prometheus no CloudWatch Container Insights, você pode configurar uma ou mais das seguintes workloads em contêineres. O atendente do CloudWatch com suporte ao Prometheus coleta automaticamente métricas de cada uma dessas workloads. Para visualizar as métricas coletadas por padrão, consulte Métricas do Prometheus coletadas pelo atendente do CloudWatch.

Antes de instalar qualquer uma dessas cargas de trabalho, instale o Helm 3.x inserindo os comandos a seguir:

```
brew install helm
```

Para obter mais informações, consulte Helm.

Tópicos

- Configurar amostra de workload AWS App Mesh para o Amazon EKS e o Kubernetes
- Configurar o NGINX com tráfego de amostra no Amazon EKS e no Kubernetes
- Configurar memcached com um exportador de métricas no Amazon EKS e no Kubernetes

- Configurar amostra de workload do Java/JMX para o Amazon EKS e o Kubernetes
- Configurar HAProxycom um exportador de métricas no Amazon EKS e no Kubernetes
- Tutorial para adicionar um novo destino de extração do Prometheus: Redis nos clusters do Amazon EKS e do Kubernetes

Configurar amostra de workload AWS App Mesh para o Amazon EKS e o Kubernetes

O suporte do Prometheus no CloudWatch Container Insights oferece suporte a AWS App Mesh. As seções a seguir explicam como configurar o App Mesh.

O CloudWatch Container Insights também pode coletar logs de acesso do App Mesh Envoy. Para ter mais informações, consulte (Opcional) Habilitar logs de acesso do App Mesh Envoy.

Tópicos

- Configurar a amostra de workload AWS App Mesh em um cluster do Amazon EKS com o tipo de inicialização do EC2 ou um cluster do Kubernetes
- Configurar a amostra de workload AWS App Mesh em um cluster do Amazon EKS com o tipo de inicialização do Fargate

Configurar a amostra de workload AWS App Mesh em um cluster do Amazon EKS com o tipo de inicialização do EC2 ou um cluster do Kubernetes

Use estas instruções ao configurar o App Mesh em um cluster que executa o Amazon EKS com o tipo de inicialização do EC2 ou em um cluster do Kubernetes.

Configurar permissões do IAM

É necessário adicionar a política AWSAppMeshFullAccess à função do IAM para seu grupo de nós do Amazon EKS ou do Kubernetes. No Amazon EKS, esse nome de grupo de nós é semelhante a `eksctl-integ-test-eks-prometheus-NodeInstanceRole-ABCDEFHIJKL`. No Kubernetes, ele pode ser semelhante a `nodes.integ-test-kops-prometheus.k8s.local`.

Instalar o App Mesh

Para instalar o controlador do App Mesh Kubernetes, siga as instruções em Controlador do App Mesh.

Instalar uma amostra de aplicação

aws-app-mesh-examples contém várias demonstrações do Kubernetes App Mesh. Com este tutorial, você instala uma amostra de aplicação de cor que mostra como as rotas http podem usar cabeçalhos para correspondência de solicitações recebidas.

Para usar uma amostra de aplicação do App Mesh amostra para testar Insights de contêiner

1. Instale a aplicação usando estas instruções: https://github.com/aws/aws-app-mesh-examples/tree/main/walkthroughs/howto-k8s-http-headers.
2. Inicie um pod de curler para gerar tráfego:

   ```
   kubectl -n default run -it curler --image=tutum/curl /bin/bash
   ```

3. Execute curl em diferentes endpoints alterando cabeçalhos HTTP. Execute o comando curl várias vezes, conforme mostrado:

   ```
   curl -H "color_header: blue" front.howto-k8s-http-headers.svc.cluster.local:8080/;
    echo;

   curl -H "color_header: red" front.howto-k8s-http-headers.svc.cluster.local:8080/;
    echo;

   curl -H "color_header: yellow" front.howto-k8s-http-
   headers.svc.cluster.local:8080/; echo;
   ```

4. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
5. Na região da AWS em que o cluster está em execução, escolha Metrics (Métricas) no painel de navegação. A métrica está no namespace ContainerInsights/Prometheus.
6. Para visualizar os eventos do CloudWatch Logs, escolha Log Groups (Grupos de logs) no painel de navegação. Os eventos estão no grupo de logs `/aws/containerinsights/your_cluster_name/prometheus` no fluxo de logs `kubernetes-pod-appmesh-envoy`.

Excluir o ambiente de teste do App Mesh

Ao terminar de usar o App Mesh e a aplicação de exemplo, use os comandos a seguir para excluir os recursos desnecessários. Excluir a aplicação de exemplo inserindo o comando a seguir:

```
cd aws-app-mesh-examples/walkthroughs/howto-k8s-http-headers/
```

```
kubectl delete -f _output/manifest.yaml
```

Excluir o controlador do App Mesh inserindo o comando a seguir:

```
helm delete appmesh-controller -n appmesh-system
```

Configurar a amostra de workload AWS App Mesh em um cluster do Amazon EKS com o tipo de inicialização do Fargate

Use estas instruções ao configurar o App Mesh em um cluster que executa o Amazon EKS com o tipo de inicialização do Fargate.

Configurar permissões do IAM

Insira o seguinte comando para definir as permissões do IAM. Substitua *MyCluster* pelo nome do cluster.

```
eksctl create iamserviceaccount --cluster MyCluster \
 --namespace howto-k8s-fargate \
 --name appmesh-pod \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshEnvoyAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapDiscoverInstanceAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSXRayDaemonWriteAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshFullAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapFullAccess \
 --override-existing-serviceaccounts \
 --approve
```

Instalar o App Mesh

Para instalar o controlador do App Mesh Kubernetes, siga as instruções em Controlador do App Mesh. Siga as instruções para o Amazon EKS com o tipo de inicialização do Fargate.

Instalar uma amostra de aplicação

aws-app-mesh-examples contém várias demonstrações do Kubernetes App Mesh. Neste tutorial, você instala uma amostra de aplicaçao de cor que funciona para clusters do Amazon EKS com o tipo de inicialização do Fargate.

Para usar uma amostra de aplicação do App Mesh amostra para testar Insights de contêiner

1. Instale a aplicação usando estas instruções: https://github.com/aws/aws-app-mesh-examples/tree/main/walkthroughs/howto-k8s-fargate.

   Essas instruções pressupõem que você esteja criando um novo cluster com o perfil correto do Fargate. Caso queira usar um cluster do Amazon EKS que você já configurou, use os seguintes comandos para configurar o cluster para esta demonstração. Substitua *MyCluster* pelo nome do cluster.

   ```
   eksctl create iamserviceaccount --cluster MyCluster \
    --namespace howto-k8s-fargate \
    --name appmesh-pod \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshEnvoyAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapDiscoverInstanceAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSXRayDaemonWriteAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshFullAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapFullAccess \
    --override-existing-serviceaccounts \
    --approve
   ```

   ```
   eksctl create fargateprofile --cluster MyCluster \
   --namespace howto-k8s-fargate --name howto-k8s-fargate
   ```

2. Encaminhe pela porta a implantação frontal da aplicação:

   ```
   kubectl -n howto-k8s-fargate port-forward deployment/front 8080:8080
   ```

3. Execute curl na aplicação frontal:

   ```
   while true; do  curl -s http://localhost:8080/color; sleep 0.1; echo ; done
   ```

4. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
5. Na região da AWS em que o cluster está em execução, escolha Metrics (Métricas) no painel de navegação. A métrica está no namespace ContainerInsights/Prometheus.
6. Para visualizar os eventos do CloudWatch Logs, escolha Log Groups (Grupos de logs) no painel de navegação. Os eventos estão no grupo de logs `/aws/containerinsights/`*`your_cluster_name`*`/prometheus` no fluxo de logs `kubernetes-pod-appmesh-envoy`.

Excluir o ambiente de teste do App Mesh

Ao terminar de usar o App Mesh e a aplicação de exemplo, use os comandos a seguir para excluir os recursos desnecessários. Excluir a aplicação de exemplo inserindo o comando a seguir:

```
cd aws-app-mesh-examples/walkthroughs/howto-k8s-fargate/
kubectl delete -f _output/manifest.yaml
```

Excluir o controlador do App Mesh inserindo o comando a seguir:

```
helm delete appmesh-controller -n appmesh-system
```

Configurar o NGINX com tráfego de amostra no Amazon EKS e no Kubernetes

O NGINX é um servidor web que também pode ser usado como load balancer e proxy reverso. Para obter mais informações sobre como o Kubernetes usa o NGINX para entrada, consulte kubernetes/ingress-nginx.

Como instalar o Ingress-NGINX com uma amostra de serviço de tráfego para testar o suporte ao Container Insights Prometheus

1. Insira comando a seguir para adicionar o repositório ingress-nginx do Helm:

   ```
   helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
   ```

2. Insira os comandos a seguir:

   ```
   kubectl create namespace nginx-ingress-sample

   helm install my-nginx ingress-nginx/ingress-nginx \
   --namespace nginx-ingress-sample \
   --set controller.metrics.enabled=true \
   --set-string controller.metrics.service.annotations."prometheus\.io/port"="10254" \
   --set-string controller.metrics.service.annotations."prometheus\.io/scrape"="true"
   ```

3. Verifique se os serviços foram iniciados corretamente inserindo o comando a seguir:

   ```
   kubectl get service -n nginx-ingress-sample
   ```

   A saída deste comando deve exibir várias colunas, incluindo uma coluna `EXTERNAL-IP`.

4. Defina uma variável `EXTERNAL-IP` para o valor da coluna `EXTERNAL-IP` na linha do controlador de entrada do NGINX.

```
EXTERNAL_IP=your-nginx-controller-external-ip
```

5. Inicie alguns exemplos de tráfego do NGINX inserindo o comando a seguir.

```
SAMPLE_TRAFFIC_NAMESPACE=nginx-sample-traffic
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_traffic/nginx-traffic/nginx-traffic-sample.yaml |
sed "s/{{external_ip}}/$EXTERNAL_IP/g" |
sed "s/{{namespace}}/$SAMPLE_TRAFFIC_NAMESPACE/g" |
kubectl apply -f -
```

6. Insira o comando a seguir para confirmar se todos os três pods estão no status `Running`.

```
kubectl get pod -n $SAMPLE_TRAFFIC_NAMESPACE
```

Se eles estiverem em execução, logo você verá as métricas no namespace ContainerInsights/Prometheus .

Como desinstalar o NGINX e o aplicativo de tráfego de exemplo

1. Exclua o serviço de tráfego de exemplo inserindo o comando a seguir:

```
kubectl delete namespace $SAMPLE_TRAFFIC_NAMESPACE
```

2. Exclua a saída do NGINX pelo nome da versão do Helm.

```
helm uninstall my-nginx --namespace nginx-ingress-sample
kubectl delete namespace nginx-ingress-sample
```

Configurar memcached com um exportador de métricas no Amazon EKS e no Kubernetes

memcached é um sistema de armazenamento em cache de objetos na memória de código aberto. Para obter mais informações, consulte O que é Memcached?

Se você estiver executando o memcached em um cluster com o tipo de inicialização do Fargate, precisará configurar um perfil do Fargate antes de executar as etapas deste procedimento. Para configurar o perfil, insira o comando a seguir. Substitua *MyCluster* pelo nome do cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace memcached-sample --name memcached-sample
```

Como instalar memcached com um exportador de métricas para testar o suporte do Container Insights Prometheus

1. Insira comando a seguir para adicionar o repositório:

   ```
   helm repo add bitnami https://charts.bitnami.com/bitnami
   ```

2. Insira o comando a seguir para criar um novo namespace:

   ```
   kubectl create namespace memcached-sample
   ```

3. Insira o comando a seguir para instalar o Memcached

   ```
   helm install my-memcached bitnami/memcached --namespace memcached-sample \
   --set metrics.enabled=true \
   --set-string serviceAnnotations.prometheus\\.io/port="9150" \
   --set-string serviceAnnotations.prometheus\\.io/scrape="true"
   ```

4. Insira o comando a seguir para confirmar a anotação do serviço em execução:

   ```
   kubectl describe service my-memcached-metrics -n memcached-sample
   ```

   Você verá as duas anotações a seguir:

   ```
   Annotations:   prometheus.io/port: 9150
                  prometheus.io/scrape: true
   ```

Como desinstalar o memcached

- Insira os comandos a seguir:

```
helm uninstall my-memcached --namespace memcached-sample
kubectl delete namespace memcached-sample
```

Configurar amostra de workload do Java/JMX para o Amazon EKS e o Kubernetes

O JMX Exporter é um exportador oficial do Prometheus que pode extrair conteúdo e expor mBeans da JMX como métricas do Prometheus. Para obter mais informações, consulte prometheus/jmx_exporter.

O Container Insights pode coletar métricas predefinidas do Prometheus a partir da Java Virtual Machine (JVM), Java e Tomcat (Catalina) usando o JMX Exporter.

Configuração padrão de extração do Prometheus

Por padrão, o atendente do CloudWatch com suporte ao Prometheus extrai as métricas do Java/JMX do Prometheus de `http://CLUSTER_IP:9404/metrics` em cada pod em um cluster do Amazon EKS ou do Kubernetes. Isso é feito pela detecção `role: pod` do Prometheus `kubernetes_sd_config`. 9404 é a porta padrão alocada para o JMX Exporter pelo Prometheus. Para obter mais informações sobre a detecção `role: pod`, consulte pod. Você pode configurar o JMX Exporter de modo a expor as métricas em uma porta ou metrics_path diferente. Se você alterar a porta ou o caminho, atualize o jmx scrape_config padrão no mapa de configuração do atendente do CloudWatch. Execute o comando a seguir para obter a configuração do Prometheus atual do atendente do CloudWatch:

```
kubectl describe cm prometheus-config -n amazon-cloudwatch
```

Os campos a serem alterados são os campos `/metrics` e `regex: '.*:9404$'`, conforme destacado no exemplo a seguir.

```
job_name: 'kubernetes-jmx-pod'
sample_limit: 10000
metrics_path: /metrics
kubernetes_sd_configs:
- role: pod
relabel_configs:
- source_labels: [__address__]
  action: keep
  regex: '.*:9404$'
```

```
- action: replace
  regex: (.+)
  source_labels:
```

Outra configuração de extração do Prometheus

Se você expuser sua aplicação em execução em um conjunto de pods com exportadores Java/JMX Prometheus por um Kubernetes Service, também será possível alternar para usar detecção `role: service` ou detecção `role: endpoint` do Prometheus `kubernetes_sd_config`. Para obter mais informações sobre esses métodos de detecção, consulte serviço, endpoints e<kubernetes_sd_config>..

Mais meta rótulos são fornecidos por esses dois modos de detecção de serviço que podem ser úteis para você criar as dimensões de métricas do CloudWatch. Por exemplo, é possível rotular `__meta_kubernetes_service_name` como `Service` e incluí-lo na dimensão de suas métricas. Para obter mais informações sobre como personalizar suas métricas do CloudWatch e suas dimensões, consulte Configuração do atendente do CloudWatch para o Prometheus.

Imagem do docker com o JMX Exporter

Crie uma imagem do Docker. As seções a seguir fornecem dois exemplos de Dockerfiles.

Quando você tiver compilado a imagem, carregue-a no Amazon EKS ou no Kubernetes e execute o comando a seguir para verificar se as métricas do Prometheus são expostas por `JMX_EXPORTER` na porta 9404. Substitua *`$JAR_SAMPLE_TRAFFIC_POD`* pelo nome do pod em execução e substitua *`$JAR_SAMPLE_TRAFFIC_NAMESPACE`* pelo namespace do aplicativo.

Se você estiver executando o JMX Exporter em um cluster com o tipo de inicialização do Fargate, precisará configurar um perfil do Fargate antes de executar as etapas deste procedimento. Para configurar o perfil, insira o comando a seguir. Substitua *`MyCluster`* pelo nome do cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace $JAR_SAMPLE_TRAFFIC_NAMESPACE\
 --name $JAR_SAMPLE_TRAFFIC_NAMESPACE
```

```
kubectl exec $JAR_SAMPLE_TRAFFIC_POD -n $JARCAT_SAMPLE_TRAFFIC_NAMESPACE -- curl
 http://localhost:9404
```

Exemplo: imagem do Docker do Apache Tomcat com métricas do Prometheus

O servidor Apache Tomcat expõe mBeans da JMX por padrão. Você pode integrar o JMX Exporter ao Tomcat para expor o mBeans da JMX como métricas do Prometheus. O exemplo de dockerfile a seguir mostra as etapas para criação de uma imagem de teste:

```
# From Tomcat 9.0 JDK8 OpenJDK
FROM tomcat:9.0-jdk8-openjdk

RUN mkdir -p /opt/jmx_exporter

COPY ./jmx_prometheus_javaagent-0.12.0.jar /opt/jmx_exporter
COPY ./config.yaml /opt/jmx_exporter
COPY ./setenv.sh /usr/local/tomcat/bin
COPY your web application.war /usr/local/tomcat/webapps/

RUN chmod  o+x /usr/local/tomcat/bin/setenv.sh

ENTRYPOINT ["catalina.sh", "run"]
```

A lista a seguir explica as quatro linhas `COPY` deste dockerfile.

- Faça download do arquivo jar mais recente do JMX Exporter em https://github.com/prometheus/jmx_exporter.
- `config.yaml` é o arquivo de configuração do JMX Exporter. Para obter mais informações, consulte https://github.com/prometheus/jmx_exporter#Configuration.

  Veja a seguir um arquivo de configuração de exemplo para Java e Tomcat:

  ```
  lowercaseOutputName: true
  lowercaseOutputLabelNames: true

  rules:
  - pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
  TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
  ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
    name: java_lang_OperatingSystem_$1
    type: GAUGE

  - pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
    name: java_lang_threading_$1
    type: GAUGE
  ```

```
- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

- `setenv.sh` é um script de inicialização do Tomcat para iniciar o JMX exporter com o Tomcat e expor métricas do Prometheus na porta 9404 do localhost. Ele também fornece ao JMX Exporter o caminho do arquivo `config.yaml`.

```
$ cat setenv.sh
export JAVA_OPTS="-javaagent:/opt/jmx_exporter/
jmx_prometheus_javaagent-0.12.0.jar=9404:/opt/jmx_exporter/config.yaml $JAVA_OPTS"
```

- o aplicativo web .war é o arquivo de aplicativo web `war` a ser carregado pelo Tomcat.

Crie uma imagem do Docker com essa configuração e carregue para um repositório de imagens.

Exemplo: imagem do Docker da aplicação Java Jar com métricas do Prometheus

O exemplo de dockerfile a seguir mostra as etapas para criação de uma imagem de teste:

```
# Alpine Linux with OpenJDK JRE
FROM openjdk:8-jre-alpine

RUN mkdir -p /opt/jmx_exporter

COPY ./jmx_prometheus_javaagent-0.12.0.jar /opt/jmx_exporter
COPY ./SampleJavaApplication-1.0-SNAPSHOT.jar /opt/jmx_exporter
COPY ./start_exporter_example.sh /opt/jmx_exporter
COPY ./config.yaml /opt/jmx_exporter

RUN chmod -R o+x /opt/jmx_exporter
RUN apk add curl

ENTRYPOINT exec /opt/jmx_exporter/start_exporter_example.sh
```

A lista a seguir explica as quatro linhas `COPY` deste dockerfile.

- Faça download do arquivo jar mais recente do JMX Exporter em https://github.com/prometheus/jmx_exporter.
- `config.yaml` é o arquivo de configuração do JMX Exporter. Para obter mais informações, consulte https://github.com/prometheus/jmx_exporter#Configuration.

  Veja a seguir um arquivo de configuração de exemplo para Java e Tomcat:

```
lowercaseOutputName: true
lowercaseOutputLabelNames: true

rules:
```

```
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
```

```
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

- `start_exporter_example.sh` é o script para iniciar o aplicativo JAR com as métricas exportadas do Prometheus. Ele também fornece ao JMX Exporter o caminho do arquivo `config.yaml`.

```
$ cat start_exporter_example.sh
java -javaagent:/opt/jmx_exporter/jmx_prometheus_javaagent-0.12.0.jar=9404:/
opt/jmx_exporter/config.yaml -cp  /opt/jmx_exporter/SampleJavaApplication-1.0-
SNAPSHOT.jar com.gubupt.sample.app.App
```

- SampleJavaApplication-1.0-SNAPSHOT.jar é o arquivo jar do aplicativo Java de amostra. Substitua-o pelo aplicativo Java que você deseja monitorar.

Crie uma imagem do Docker com essa configuração e carregue para um repositório de imagens.

Configurar HAProxycom um exportador de métricas no Amazon EKS e no Kubernetes

HAProxy é um aplicativo proxy de código aberto. Para obter mais informações, consulte HAProxy.

Se você estiver executando o HAProxyem um cluster com o tipo de inicialização do Fargate, precisará configurar um perfil do Fargate antes de executar as etapas deste procedimento. Para configurar o perfil, insira o comando a seguir. Substitua *`MyCluster`* pelo nome do cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace haproxy-ingress-sample --name haproxy-ingress-sample
```

Como instalar o HAProxy com um exportador de métricas para testar o suporte do Container Insights Prometheus

1. Insira o comando a seguir para adicionar o repositório da incubadora do Helm:

   ```
   helm repo add haproxy-ingress https://haproxy-ingress.github.io/charts
   ```

2. Insira o comando a seguir para criar um novo namespace:

   ```
   kubectl create namespace haproxy-ingress-sample
   ```

3. Insira os comandos a seguir para instalar o HAProxy:

```
helm install haproxy haproxy-ingress/haproxy-ingress \
--namespace haproxy-ingress-sample \
--set defaultBackend.enabled=true \
--set controller.stats.enabled=true \
--set controller.metrics.enabled=true \
--set-string controller.metrics.service.annotations."prometheus\.io/port"="9101" \
--set-string controller.metrics.service.annotations."prometheus\.io/scrape"="true"
```

4. Insira o comando a seguir para confirmar a anotação do serviço:

```
kubectl describe service haproxy-haproxy-ingress-metrics -n haproxy-ingress-sample
```

   Você verá as anotações a seguir.

```
Annotations:   prometheus.io/port: 9101
               prometheus.io/scrape: true
```

Como desinstalar o HAProxy

- Insira os comandos a seguir:

```
helm uninstall haproxy --namespace haproxy-ingress-sample
kubectl delete namespace haproxy-ingress-sample
```

Tutorial para adicionar um novo destino de extração do Prometheus: Redis nos clusters do Amazon EKS e do Kubernetes

Este tutorial fornece uma introdução prática para extrair as métricas do Prometheus de uma aplicação de exemplo do Redis em um cluster do Amazon EKS e do Kubernetes. O Redis (https://redis.io/) é um armazenamento de estrutura de dados de código aberto (licença BSD) na memória, usado como banco de dados, cache e atendente de mensagens. Para obter mais informações, consulte redis.

redis_exporter (Licença MIT) é usado para expor as métricas do Redis prometheus na porta especificada (padrão: 0.0.0.0:9121). Para obter mais informações, consulte redis_exporter.

As imagens do Docker dos dois repositórios do Docker Hub a seguir são usadas neste tutorial:

- redis
- redis_exporter

Para instalar uma workload de exemplo do Redis que expõe as métricas do Prometheus

1. Defina o namespace para a workload de exemplo do Redis.

```
REDIS_NAMESPACE=redis-sample
```

2. Se você estiver executando o Redis em um cluster com o tipo de inicialização do Fargate, precisará configurar um perfil do Fargate. Para configurar o perfil, insira o comando a seguir. Substitua *MyCluster* pelo nome do cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace $REDIS_NAMESPACE --name $REDIS_NAMESPACE
```

3. Instale a workload de exemplo do Redis inserindo o comando a seguir.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_traffic/redis/redis-traffic-sample.yaml \
| sed "s/{{namespace}}/$REDIS_NAMESPACE/g" \
| kubectl apply -f -
```

4. A instalação inclui um serviço chamado `my-redis-metrics` que expõe a métrica do Redis Prometheus na porta 9121. Insira o seguinte comando para obter os detalhes do serviço:

```
kubectl describe service/my-redis-metrics  -n $REDIS_NAMESPACE
```

   Na seção `Annotations` dos resultados, você verá duas anotações que correspondem à configuração de extração do Prometheus do atendente do CloudWatch, para que ele possa detectar as workloads automaticamente:

```
prometheus.io/port: 9121
prometheus.io/scrape: true
```

   A configuração de extração do Prometheus relacionada pode ser encontrada na seção `- job_name: kubernetes-service-endpoints` de`kubernetes-eks.yaml` ou `kubernetes-k8s.yaml`.

Como começar a coletar métricas do Redis Prometheus no CloudWatch

1. Baixe a versão mais recente do arquivo `kubernetes-eks.yaml` ou `kubernetes-k8s.yaml` inserindo um dos comandos a seguir. Para um cluster do Amazon EKS com o tipo de inicialização do EC2, insira este comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
   ```

   Para um cluster do Amazon EKS com o tipo de inicialização do Fargate, insira este comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks-fargate.yaml
   ```

   Para um cluster do Kubernetes em execução em uma instância do Amazon EC2, insira este comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-k8s.yaml
   ```

2. Abra o arquivo com um editor de textos e localize a seção `cwagentconfig.json`. Adicione a seguinte subseção e salve as alterações. Verifique se o recuo segue o padrão existente.

   ```
   {
     "source_labels": ["pod_name"],
     "label_matcher": "^redis-instance$",
     "dimensions": [["Namespace","ClusterName"]],
     "metric_selectors": [
       "^redis_net_(in|out)put_bytes_total$",
       "^redis_(expired|evicted)_keys_total$",
       "^redis_keyspace_(hits|misses)_total$",
       "^redis_memory_used_bytes$",
       "^redis_connected_clients$"
     ]
   },
   {
     "source_labels": ["pod_name"],
     "label_matcher": "^redis-instance$",
   ```

```
  "dimensions": [["Namespace","ClusterName","cmd"]],
  "metric_selectors": [
    "^redis_commands_total$"
  ]
},
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName","db"]],
  "metric_selectors": [
    "^redis_db_keys$"
  ]
},
```

A seção que você adicionou coloca as métricas do Redis na lista de permissões do atendente do CloudWatch. Para obter a lista dessas métricas, consulte a seção a seguir.

3. Se você já tem o atendente do CloudWatch com suporte ao Prometheus implantado nesse cluster, exclua-o inserindo o comando a seguir.

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

4. Implante o atendente do CloudWatch com a configuração atualizada inserindo um dos comandos a seguir. Substitua *`MyCluster`* e *`region`* para corresponder a suas configurações.

   Para um cluster do Amazon EKS com o tipo de inicialização do EC2, insira este comando.

```
kubectl apply -f prometheus-eks.yaml
```

   Para um cluster do Amazon EKS com o tipo de inicialização do Fargate, insira este comando.

```
cat prometheus-eks-fargate.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

   Para um cluster do Kubernetes, insira este comando.

```
cat prometheus-k8s.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Visualizar as métricas do Redis Prometheus

Este tutorial envia as seguintes métricas ao namespace ContainerInsights/Prometheus no CloudWatch. É possível usar o console do CloudWatch para ver as métricas nesse namespace.

Nome da métrica	Dimensões	
`redis_net_input_bytes_total`	ClusterName, `Namespace`	
`redis_net_output_bytes_total`	ClusterName, `Namespace`	
`redis_expired_keys_total`	ClusterName, `Namespace`	
`redis_evicted_keys_total`	ClusterName, `Namespace`	
`redis_keyspace_hits_total`	ClusterName, `Namespace`	
`redis_keyspace_misses_total`	ClusterName, `Namespace`	
`redis_memory_used_bytes`	ClusterName, `Namespace`	
`redis_connected_clients`	ClusterName, `Namespace`	
`redis_commands_total`	ClusterName, `Namespace` , cmd	

Nome da métrica	Dimensões	
redis_db_keys	ClusterName, Namespace , db	

> **Note**
>
> O valor da dimensão cmd pode ser: append, client, command, config, dbsize, flushall, get, incr, info, latency ou slowlog.
> Os valores da dimensão db podem ser db0 ou db15.

Também é possível criar um painel do CloudWatch para suas métricas do Redis Prometheus.

Para criar um painel para métricas do Redis Prometheus

1. Crie variáveis de ambiente, substituindo os valores abaixo para corresponder a sua implantação.

```
DASHBOARD_NAME=your_cw_dashboard_name
REGION_NAME=your_metric_region_such_as_us-east-1
CLUSTER_NAME=your_k8s_cluster_name_here
NAMESPACE=your_redis_service_namespace_here
```

2. Use o comando a seguir para criar o painel.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/redis/cw_dashboard_redis.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| sed "s/{{YOUR_NAMESPACE}}/${NAMESPACE}/g" \
```

Conversão de tipo de métrica do Prometheus pelo CloudWatch Agent

As bibliotecas cliente Prometheus oferecem quatro tipos de métricas principais:

- Contador
- Medidor
- Resumo

- Histograma

O atendente do CloudWatch oferece suporte aos tipos de métricas de contador, medidor e resumo. O suporte para métricas de histograma está planejado para um lançamento futuro.

As métricas do Prometheus com o tipo de métrica de histograma não compatível são descartadas pelo atendente do CloudWatch. Para ter mais informações, consulte Registrar as métricas descartadas do Prometheus.

Métricas de medidor

Uma métrica de medidor Prometheus é uma métrica que representa um único valor numérico que pode aumentar e diminuir arbitrariamente. O atendente do CloudWatch extrai métricas de medidor e envia esses valores diretamente.

Métricas de contador

Uma métrica de contador Prometheus é uma métrica cumulativa que representa um único contador que aumenta de forma monotônica cujo valor só pode aumentar ou ser redefinido para zero. O atendente do CloudWatch calcula um delta da extração anterior e envia o valor delta como o valor da métrica no evento de log. Assim, o atendente do CloudWatch começará a produzir um evento de log a partir da segunda extração e continuará com as extrações subsequentes, se houver.

Métricas de resumo

Uma métrica de resumo do Prometheus é um tipo de métrica complexa que é representada por vários pontos de dados. Ela fornece uma contagem total de observações e uma soma de todos os valores observados. Calcula quantis configuráveis sobre uma janela de tempo deslizante.

A soma e a contagem de uma métrica resumida são cumulativas, mas os quantis não são. O exemplo a seguir mostra a variância dos quantis.

```
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 7.123e-06
go_gc_duration_seconds{quantile="0.25"} 9.204e-06
go_gc_duration_seconds{quantile="0.5"} 1.1065e-05
go_gc_duration_seconds{quantile="0.75"} 2.8731e-05
go_gc_duration_seconds{quantile="1"} 0.003841496
go_gc_duration_seconds_sum 0.37630427
go_gc_duration_seconds_count 9774
```

O atendente do CloudWatch lida com a soma e a contagem de uma métrica resumida da mesma forma que lida com métricas de contador, conforme descrito na seção anterior. O atendente do CloudWatch preserva os valores dos quantis conforme eles são originalmente informados.

Métricas do Prometheus coletadas pelo atendente do CloudWatch

O atendente do CloudWatch com suporte ao Prometheus coleta automaticamente métricas de vários serviços e workloads. As métricas que são coletadas por padrão estão listadas nas seções a seguir. Você também pode configurar o atendente para coletar mais métricas desses serviços e coletar métricas do Prometheus de outras aplicações e serviços. Para obter mais informações sobre coletar outras métricas, consulte Configuração do atendente do CloudWatch para o Prometheus.

As métricas da Prometheus coletadas de clusters do Amazon EKS e do Kubernetes estão no namespace ContainerInsights/Prometheus. As métricas da Prometheus coletadas de clusters do Amazon ECS estão no namespace ECS/ContainerInsights/Prometheus.

Tópicos

- Métricas do Prometheus para o App Mesh
- Métricas do Prometheus para NGINX
- Métricas do Prometheus para Memcached
- Métricas do Prometheus para Java/JMX
- Métricas do Prometheus para HAProxy

Métricas do Prometheus para o App Mesh

As métricas a seguir são coletadas automaticamente do App Mesh.

O CloudWatch Container Insights também pode coletar logs de acesso do App Mesh Envoy. Para ter mais informações, consulte (Opcional) Habilitar logs de acesso do App Mesh Envoy.

Métricas do Prometheus para App Mesh em clusters do Amazon EKS e do Kubernetes

Nome da métrica	Dimensões	
`envoy_http_downstream_rq_total`	ClusterName, `Namespace`	

Nome da métrica	Dimensões	
envoy_http_downstream_rq_xx	ClusterName, Namespace ClusterName, Namespace , envoy_http_conn_manager_prefix, envoy_response_code_class	
envoy_cluster_upstream_cx_rx_bytes_total	ClusterName, Namespace	
envoy_cluster_upstream_cx_tx_bytes_total	ClusterName, Namespace	
envoy_cluster_membership_healthy	ClusterName, Namespace	
envoy_cluster_membership_total	ClusterName, Namespace	
envoy_server_memory_heap_size	ClusterName, Namespace	
envoy_server_memory_allocated	ClusterName, Namespace	
envoy_cluster_upstream_cx_connect_timeout	ClusterName, Namespace	

Nome da métrica	Dimensões	
envoy_cluster_upstream_rq_pending_failure_eject	ClusterName, Namespace	
envoy_cluster_upstream_rq_pending_overflow	ClusterName, Namespace	
envoy_cluster_upstream_rq_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_try_per_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_rx_reset	ClusterName, Namespace	
envoy_cluster_upstream_cx_destroy_local_with_active_rq	ClusterName, Namespace	

Nome da métrica	Dimensões	
envoy_cluster_upstream_cx_destroy_remote_active_rq	ClusterName, Namespace	
envoy_cluster_upstream_rq_maintenance_mode	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_paused_reading_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_resumed_reading_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_backed_up_total	ClusterName, Namespace	

Nome da métrica	Dimensões	
envoy_cluster_upstream_flow_control_drained_total	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry_success	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry_overflow	ClusterName, Namespace	
envoy_server_live	ClusterName, Namespace	
envoy_server_uptime	ClusterName, Namespace	

Métricas do Prometheus para App Mesh em clusters do Amazon ECS

Nome da métrica	Dimensões	
envoy_http_downstream_rq_total	ClusterName, TaskDefinitionFamily	

Nome da métrica	Dimensões	
envoy_http_downstream_rq_xx	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_rx_bytes_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_tx_bytes_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_membership_healthy	ClusterName, TaskDefinitionFamily	
envoy_cluster_membership_total	ClusterName, TaskDefinitionFamily	
envoy_server_memory_heap_size	ClusterName, TaskDefinitionFamily	
envoy_server_memory_allocated	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_connect_timeout	ClusterName, TaskDefinitionFamily	

Nome da métrica	Dimensões	
envoy_cluster_upstream_rq_pending_failure_eject	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_pending_overflow	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_timeout	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_try_per_timeout	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_rx_reset	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_destroy_local_with_active_rq	ClusterName, TaskDefinitionFamily	

Nome da métrica	Dimensões	
envoy_cluster_upstream_cx_destroy_remote_active_rq	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_maintenance_mode	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_paused_reading_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_resumed_reading_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_backed_up_total	ClusterName, TaskDefinitionFamily	

Nome da métrica	Dimensões
envoy_cluster_upstream_flow_control_drained_total	ClusterName, TaskDefinitionFamily
envoy_cluster_upstream_rq_retry	ClusterName, TaskDefinitionFamily
envoy_cluster_upstream_rq_retry_success	ClusterName, TaskDefinitionFamily
envoy_cluster_upstream_rq_retry_overflow	ClusterName, TaskDefinitionFamily
envoy_server_live	ClusterName, TaskDefinitionFamily
envoy_server_uptime	ClusterName, TaskDefinitionFamily
envoy_http_downstream_rq_xx	ClusterName, TaskDefinitionFamily, envoy_http_conn_manager_prefix, envoy_response_code_class ClusterName, TaskDefinitionFamily, envoy_response_code_class

> **Note**
>
> `TaskDefinitionFamily` é o namespace do Kubernetes da malha.

O valor de `envoy_http_conn_manager_prefix` pode ser `ingress`, `egress` ou `admin`. O valor de `envoy_response_code_class` pode ser 1 (significa `1xx`) , 2 (significa `2xx`) ,3significa `3xx`), 4 (significa `4xx`) ou 5 (significa`5xx`).

Métricas do Prometheus para NGINX

As métricas a seguir são coletadas automaticamente do NGINX em clusters do Amazon EKS e do Kubernetes.

Nome da métrica	Dimensões	
`nginx_ingress_controller_nginx_process_cpu_seconds_total`	ClusterName, `Namespace` , serviço	
`nginx_ingress_controller_success`	ClusterName, `Namespace` , serviço	
`nginx_ingress_controller_requests`	ClusterName, `Namespace` , serviço	
`nginx_ingress_controller_nginx_process_connections`	ClusterName, `Namespace` , serviço	
`nginx_ingress_controller_nginx_proce`	ClusterName, `Namespace` , serviço	

Nome da métrica	Dimensões	
ss_connections_total		
nginx_ingress_controller_nginx_process_resident_memory_bytes	ClusterName, Namespace , serviço	
nginx_ingress_controller_config_last_reload_successful	ClusterName, Namespace , serviço	
nginx_ingress_controller_requests	ClusterName, Namespace , serviço, status	

Métricas do Prometheus para Memcached

As métricas a seguir são coletadas automaticamente do Memcached em clusters do Amazon EKS e do Kubernetes.

Nome da métrica	Dimensões	
memcached_current_items	ClusterName, Namespace , serviço	

Nome da métrica	Dimensões
memcached_current_connections	ClusterName, Namespace , serviço
memcached_limit_bytes	ClusterName, Namespace , serviço
memcached_current_bytes	ClusterName, Namespace , serviço
memcached_written_bytes_total	ClusterName, Namespace , serviço
memcached_read_bytes_total	ClusterName, Namespace , serviço
memcached_items_evicted_total	ClusterName, Namespace , serviço
memcached_items_reclaimed_total	ClusterName, Namespace , serviço
memcached_commands_total	ClusterName, Namespace , serviço ClusterName, Namespace , serviço, comando ClusterName, Namespace , serviço, status, comando

Métricas do Prometheus para Java/JMX

Métricas coletadas em clusters do Amazon EKS e do Kubernetes

Em clusters do Amazon EKS e do Kubernetes, o Container Insights pode coletar as seguintes métricas predefinidas do Prometheus do Java Virtual Machine (JVM), Java e Tomcat (Catalina) usando o JMX Exporter. Para obter mais informações, consulte prometheus/jmx_exporter no Github.

Java/JMX em clusters do Amazon EKS e do Kubernetes

Nome da métrica	Dimensões
jvm_classes_loaded	ClusterName , Namespace
jvm_threads_current	ClusterName , Namespace
jvm_threads_daemon	ClusterName , Namespace
java_lang_operatingsystem_totalswapspacesize	ClusterName , Namespace
java_lang_operatingsystem_systemcpuload	ClusterName , Namespace
java_lang_operatingsystem_processcpuload	ClusterName , Namespace
java_lang_operatingsystem_freeswapspacesize	ClusterName , Namespace

Nome da métrica	Dimensões	
java_lang_operatingsystem_totalphysicalmemorysize	ClusterName , Namespace	
java_lang_operatingsystem_freephysicalmemorysize	ClusterName , Namespace	
java_lang_operatingsystem_openfiledescriptorcount	ClusterName , Namespace	
java_lang_operatingsystem_availableprocessors	ClusterName , Namespace	
jvm_memory_bytes_used	ClusterName , Namespace , área	
jvm_memory_pool_bytes_used	ClusterName , Namespace , grupo	

> **Note**
>
> Os valores da dimensão `area` podem ser `heap` ou `nonheap`.

Os valores da dimensão `pool` podem ser `Tenured Gen`, `Compress Class Space`, `Survivor Space`, `Eden Space`, `Code Cache` ou `Metaspace`.

Tomcat/JMX em clusters do Amazon EKS e do Kubernetes

Além das métricas Java/JMX na tabela anterior, as métricas a seguir também são coletadas para a workload do Tomcat.

Nome da métrica	Dimensões	
`catalina_manager_activesessions`	`ClusterName` , `Namespace`	
`catalina_manager_rejectedsessions`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_bytesreceived`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_bytessent`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_requestcount`	`ClusterName` , `Namespace`	

Nome da métrica	Dimensões	
catalina_globalrequestprocessor_errorcount	ClusterName , Namespace	
catalina_globalrequestprocessor_processingtime	ClusterName , Namespace	

Java/JMX em clusters do Amazon ECS

Nome da métrica	Dimensões	
jvm_classes_loaded	ClusterName , TaskDefinitionFamily	
jvm_threads_current	ClusterName , TaskDefinitionFamily	
jvm_threads_daemon	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_totalswapspacesize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_systemcpuload	ClusterName , TaskDefinitionFamily	

Nome da métrica	Dimensões	
java_lang_operatingsystem_processcpuload	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_freeswapspacesize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_totalphysicalmemorysize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_freephysicalmemorysize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_openfiledescriptorcount	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_availableprocessors	ClusterName , TaskDefinitionFamily	

Nome da métrica	Dimensões	
jvm_memory_bytes_used	ClusterName , TaskDefinitionFamily, área	
jvm_memory_pool_bytes_used	ClusterName , TaskDefinitionFamily, grupo	

> **Note**
>
> Os valores da dimensão area podem ser heap ou nonheap.
> Os valores da dimensão pool podem ser Tenured Gen, Compress Class Space, Survivor Space, Eden Space, Code Cache ou Metaspace.

Tomcat/JMX em clusters do Amazon ECS

Além das métricas Java/JMX na tabela anterior, as métricas a seguir também são coletadas para a workload do Tomcat em clusters do Amazon ECS.

Nome da métrica	Dimensões	
catalina_manager_activesessions	ClusterName , TaskDefinitionFamily	
catalina_manager_rejectedsessions	ClusterName , TaskDefinitionFamily	
catalina_globalrequestprocessor_bytesreceived	ClusterName , TaskDefinitionFamily	

Nome da métrica	Dimensões	
catalina_globalrequestprocessor_bytessent	ClusterName , TaskDefinitionFamily	
catalina_globalrequestprocessor_requestcount	ClusterName , TaskDefinitionFamily	
catalina_globalrequestprocessor_errorcount	ClusterName , TaskDefinitionFamily	
catalina_globalrequestprocessor_processingtime	ClusterName , TaskDefinitionFamily	

Métricas do Prometheus para HAProxy

As métricas a seguir são coletadas automaticamente do HAProxy em clusters do Amazon EKS e do Kubernetes.

As métricas coletadas dependem da versão do HAProxy Ingress que você está usando. Para obter mais informações sobre o HAProxy Ingress e suas versões, consulte haproxy-ingress.

Nome da métrica	Dimensões	Disponibilidade
haproxy_backend_bytes_in_total	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress
haproxy_backend_bytes_out_total	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress
haproxy_backend_connection_errors_total	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress
haproxy_backend_connections_total	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress
haproxy_backend_current_sessions	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress
haproxy_backend_http_responses_total	ClusterName , Namespace , Serviço, código, back-end	Todas as versões do HAProxy Ingress
haproxy_backend_status	ClusterName , Namespace , serviço	Somente nas versões 0.10 ou posteriores do HAProxy Ingress
haproxy_backend_up	ClusterName , Namespace , serviço	Somente nas versões do HAProxy Ingress anteriores à 0.10

Nome da métrica	Dimensões	Disponibilidade
haproxy_frontend_bytes_in_total	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress
haproxy_frontend_bytes_out_total	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress
haproxy_frontend_connections_total	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress
haproxy_frontend_current_sessions	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress
haproxy_frontend_http_requests_total	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress
haproxy_frontend_http_responses_total	ClusterName , Namespace , Serviço, código, front-end	Todas as versões do HAProxy Ingress
haproxy_frontend_request_errors_total	ClusterName , Namespace , serviço	Todas as versões do HAProxy Ingress

Nome da métrica	Dimensões	Disponibilidade
`haproxy_frontend_requests_denied_total`	`ClusterName` , `Namespace` , serviço	Todas as versões do HAProxy Ingress

> **Note**
>
> Os valores da dimensão `code` podem ser `1xx`, `2xx`, `3xx`, `4xx`, `5xx` ou `other`.
> Os valores da dimensão `backend` podem ser:
>
> - `http-default-backend`, `http-shared-backend` ou `httpsback-shared-backend` para HAProxy Ingress versão 0.0.27 ou anteriores.
> - `_default_backend` para versões HAProxy Ingress posteriores a 0.0.27.
>
> Os valores da dimensão `frontend` podem ser:
>
> - `httpfront-default-backend`, `httpfront-shared-frontend` ou `httpfronts` para HAProxy Ingress versão 0.0.27 ou anteriores.
> - `_front_http` ou `_front_https` para versões HAProxy Ingress posteriores a 0.0.27.

Visualizar as métricas do Prometheus

Você pode monitorar e utilizar alarmes com todas as métricas do Prometheus, incluindo as métricas selecionadas e pré-agregadas do App Mesh, NGINX, Java/JMX, Memcached e HAProxy, e qualquer outro exportador do Prometheus, configurado manualmente, que você possa ter adicionado. Para obter mais informações sobre como coletar métricas de outros exportadores do Prometheus, consulte Tutorial para adicionar um novo destino de extração do Prometheus: métricas do servidor de API do Prometheus.

No console do CloudWatch, o Container Insights fornece os seguintes relatórios pré-criados:

- Para clusters do Amazon EKS e do Kubernetes, há relatórios pré-criados para App Mesh, NGINX, HAPROXY, Memcached e Java/JMX.
- Para clusters do Amazon ECS, há relatórios pré-compilados para App Mesh e Java/JMX.

O Container Insights também fornece painéis personalizados para cada uma das workloads das quais o Container Insights coleta métricas selecionadas. É possível baixar esses painéis no GitHub

Como visualizar todas as métricas do Prometheus

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Na lista de namespaces, escolha ContainerInsights/Prometheus or ECS/ContainerInsights/Prometheus.
4. Escolha um dos conjuntos de dimensões na lista a seguir. Marque a caixa de seleção ao lado das métricas que você deseja visualizar.

Como visualizar relatórios pré-compilados sobre as métricas do Prometheus

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Performance Monitoring (Monitoramento de performance).
3. Na caixa suspensa perto da parte superior da página, escolha qualquer uma das opções do Prometheus.

   Na outra caixa suspensa, escolha um cluster a ser visualizado

Também fornecemos painéis personalizados para NGINX, App Mesh, Memcached, HAProxy e Java/JMX.

Como usar um painel personalizado fornecido pela Amazon

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha Create dashboard (Criar painel). Insira um nome para o novo painel e escolha Create dashboard (Criar painel).
4. Em Add to this dashboard (Adicionar a este painel), escolha Cancel (Cancelar).
5. Escolha Actions (Ações), View/edit source (Exibir/editar origem).
6. Faça download de um dos arquivos JSON a seguir:

   - Origem do painel personalizado NGINX no Github.
   - Origem do painel personalizado App Mesh no Github.

- Origem do painel personalizado Memcached no Github
- Origem do painel personalizado HAProxy-Ingress no Github
- Origem do painel personalizado Java/JMX no Github.

7. Abra o arquivo JSON obtido por download com um editor de textos e faça as seguintes alterações:

   - Substitua todas as strings `{{YOUR_CLUSTER_NAME}}` pelo nome exato do cluster. Não adicione espaços em branco antes ou depois do texto.
   - Substitua todos as strings `{{YOUR_REGION}}` pela região da AWS em que o cluster está em execução. Por exemplo, **`us-west-1`** Não adicione espaços em branco antes ou depois do texto.
   - Substitua todas as strings `{{YOUR_NAMESPACE}}` pelo namespace exato da workload.
   - Substitua todas as strings `{{YOUR_SERVICE_NAME}}` pelo nome de serviço exato da workload. Por exemplo, **`haproxy-haproxy-ingress-controller-metrics`**

8. Copie todo o blob JSON e cole-o na caixa de texto no console do CloudWatch, substituindo o que já está na caixa.
9. Escolha Update (Atualizar), Save dashboard (Salvar painel).

Solucionar problemas de métricas do Prometheus

Esta seção fornece ajuda para solucionar problemas de configuração de métricas do Prometheus.

Tópicos

- Solucionar problemas de métricas do Prometheus no Amazon ECS
- Solucionar problemas de métricas do Prometheus em clusters do Amazon EKS e do Kubernetes

Solucionar problemas de métricas do Prometheus no Amazon ECS

Esta seção fornece ajuda para solucionar problemas de configuração de métricas do Prometheus em clusters do Amazon ECS.

Não visualizo as métricas do Prometheus enviadas ao CloudWatch Logs

As métricas do Prometheus devem ser ingeridas como eventos de log no grupo de logs /aws/ecs/containerinsights/cluster-name/Prometheus. Se o grupo de logs não estiver criado ou se as métricas do Prometheus não forem enviadas ao grupo de logs, primeiro você precisará conferir se os destinos

do Prometheus foram detectados corretamente pelo atendente do CloudWatch. Em seguida, confira o grupo de segurança e as configurações de permissão do atendente do CloudWatch. As etapas a seguir orientam a fazer a depuração.

Etapa 1: habilitar o modo de depuração do atendente do CloudWatch

Primeiro, altere o atendente do CloudWatch para o modo de depuração adicionando as seguintes linhas em negrito ao modelo de arquivo do AWS CloudFormation: `cwagent-ecs-prometheus-metric-for-bridge-host.yaml` ou `cwagent-ecs-prometheus-metric-for-awsvpc.yaml`. Salve o arquivo.

```
cwagentconfig.json: |
    {
      "agent": {
        "debug": true
      },
      "logs": {
        "metrics_collected": {
```

Criar um novo changeset do AWS CloudFormation em relação à pilha existente. Defina outros parâmetros do changeset para os mesmos valores de sua pilha do AWS CloudFormation existente. O exemplo a seguir é de um atendente do CloudWatch instalado em um cluster do Amazon ECS usando o tipo de inicialização do EC2 e o modo de rede de ponte.

```
ECS_NETWORK_MODE=bridge
 CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
NEW_CHANGESET_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=$ECS_EXECUTION_ROLE_NAME
 \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
```

```
    --change-set-name $NEW_CHANGESET_NAME
```

Acesse o console do AWS CloudFormation para revisar o novo changeset, `$NEW_CHANGESET_NAME`. Deve haver uma alteração aplicada ao recurso CWAgentConfigSSMParameter. Execute o changeset e reinicie a tarefa do atendente do CloudWatch inserindo os comandos a seguir.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service your_service_name_here \
--region $AWS_REGION
```

Aguarde cerca de 10 segundos e insira o comando a seguir.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service your_service_name_here \
--region $AWS_REGION
```

Etapa 2: verificar os logs de detecção de serviço do ECS

Por padrão, a definição de tarefa do ECS do atendente do CloudWatch habilita os logs na seção abaixo. Os logs são enviados ao CloudWatch Logs no grupo de logs /ecs/ecs-cwagent-prometheus.

```
LogConfiguration:
  LogDriver: awslogs
    Options:
      awslogs-create-group: 'True'
      awslogs-group: "/ecs/ecs-cwagent-prometheus"
      awslogs-region: !Ref AWS::Region
      awslogs-stream-prefix: !Sub 'ecs-${ECSLaunchType}-awsvpc'
```

Filtrar os logs pela string `ECS_SD_Stats` para obter as métricas relacionadas à detecção de serviços do ECS, conforme mostrado no exemplo a seguir.

```
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeContainerInstances: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeInstancesRequest: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeTaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeTasks: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_ListTasks: 1
```

```
2020-09-1T01:53:14Z D! ECS_SD_Stats: Exporter_DiscoveredTargetCount: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Get_EC2MetaData: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Get_TaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Size_ContainerInstance: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Size_TaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: Latency: 43.399783ms
```

O significado de cada métrica para determinado ciclo de detecção de serviços do ECS é o seguinte:

- AWSCLI_DescribeContainerInstances: o número de chamadas de API `ECS::DescribeContainerInstances` realizadas.
- AWSCLI_DescribeInstancesRequest: o número de chamadas de API `ECS::DescribeInstancesRequest` realizadas.
- AWSCLI_DescribeTaskDefinition: o número de chamadas de API `ECS::DescribeTaskDefinition` realizadas.
- AWSCLI_DescribeTasks: o número de chamadas de API `ECS::DescribeTasks` realizadas.
- AWSCLI_ListTasks: o número de chamadas de API `ECS::ListTasks` realizadas.
- ExporterDiscoveredTargetCount: o número de destinos do Prometheus que foram detectados e exportados corretamente para o arquivo de resultados de destino dentro do contêiner.
- LRUCache_Get_EC2MetaData: o número de vezes em que os metadados de instâncias de contêiner foram recuperados do cache.
- LRUCache_Get_TaskDefinition: o número de vezes que os metadados de definição de tarefa do ECS foram recuperados do cache.
- LRUCache_Size_ContainerInstance: o número de metadados da instância de contêiner exclusiva armazenados em cache na memória.
- LRUCache_Size_TaskDefinition: o número de definições de tarefa de ECS exclusivas armazenadas em cache na memória.
- Latency: quanto tempo demora o ciclo de detecção de serviços.

Confira o valor de `ExporterDiscoveredTargetCount` para ver se os destinos detectados do Prometheus correspondem a suas expectativas. Caso contrário, os possíveis motivos são:

- A configuração da detecção de serviços do ECS pode não corresponder à configuração de sua aplicação. Para a detecção de serviços baseada em rótulos do docker, seus contêineres de destino talvez não tenham o rótulo do docker necessário configurado no atendente do CloudWatch para detectá-los automaticamente. Para a detecção de serviços baseada em expressão regular do

ARN da definição de tarefa do ECS, a configuração regex no atendente do CloudWatch pode não corresponder à definição de tarefa da aplicação.

- A função da tarefa do ECS do atendente do CloudWatch pode não ter permissão para recuperar os metadados das tarefas do ECS. Verifique se o atendente do CloudWatch recebeu as seguintes permissões somente para leitura:
  - `ec2:DescribeInstances`
  - `ecs:ListTasks`
  - `ecs:DescribeContainerInstances`
  - `ecs:DescribeTasks`
  - `ecs:DescribeTaskDefinition`

Etapa 3: verificar a conexão de rede e a política de função de tarefa do ECS

Se ainda não houver eventos de log enviados ao grupo de logs do CloudWatch Logs de destino, mesmo que o valor de `Exporter_DiscoveredTargetCount` indique que há destinos do Prometheus detectados, isso pode ser causado por uma destas situações:

- O atendente do CloudWatch talvez não consiga se conectar às portas de destino do Prometheus. Verifique a configuração do grupo de segurança por trás do atendente do CloudWatch. O IP privado deve permitir que o atendente do CloudWatch se conecte às portas do exportador do Prometheus.
- A função da tarefa do ECS do atendente do CloudWatch talvez não tenha a política gerenciada CloudWatchAgentServerPolicy. A função da tarefa do ECS do atendente do CloudWatch precisa ter essa política para poder enviar as métricas do Prometheus como eventos de log. Se você usou o modelo do AWS CloudFormation para criar as funções do IAM automaticamente, tanto a função da tarefa do ECS como a função de execução do ECS são concedidas com o menor privilégio para executar o monitoramento do Prometheus.

Solucionar problemas de métricas do Prometheus em clusters do Amazon EKS e do Kubernetes

Esta seção fornece ajuda para solucionar problemas de configuração de métricas do Prometheus em clusters do Amazon EKS e do Kubernetes.

Etapas gerais de solução de problemas no Amazon EKS

Insira o comando a seguir para confirmar se o atendente do CloudWatch está em execução.

```
kubectl get pod -n amazon-cloudwatch
```

A saída deve incluir uma linha com `cwagent-prometheus-id` na coluna `NAME`, e `Running` no campo `STATUS column.`

Para exibir detalhes sobre o pod em execução, insira o comando a seguir. Substitua o *pod-name* pelo nome completo do pod que tem o nome que começa com `cw-agent-prometheus`.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

Se você tiver o Container Insights Container Insights instalado, poderá usar o CloudWatch Logs Insights para consultar os logs do atendente do CloudWatch que coleta as métricas do Prometheus.

Como consultar os logs do aplicativo

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, em Logs, escolha CloudWatch Logs Insights.
3. Selecione o grupo de logs para os logs do aplicativo, /aws/containerinsights/***cluster-name***/application
4. Substitua a expressão de consulta de pesquisa pela seguinte consulta e escolha Run query (Executar consulta)

   ```
   fields ispresent(kubernetes.pod_name) as haskubernetes_pod_name, stream,
    kubernetes.pod_name, log |
   filter haskubernetes_pod_name and kubernetes.pod_name like /cwagent-prometheus
   ```

Você também pode confirmar se as métricas e os metadados do Prometheus estão sendo ingeridos como eventos do CloudWatch Logs.

Como confirmar se os dados do Prometheus estão sendo ingeridos

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, em Logs, escolha CloudWatch Logs Insights.
3. Selecione /aws/containerinsights/***cluster-name***/prometheus
4. Substitua a expressão de consulta de pesquisa pela seguinte consulta e escolha Run query (Executar consulta)

```
fields @timestamp, @message | sort @timestamp desc | limit 20
```

Registrar as métricas descartadas do Prometheus

Essa versão não coleta métricas do Prometheus do tipo histograma. Você pode usar o atendente do CloudWatch para verificar se alguma métrica do Prometheus está sendo descartada por ser de uma métrica de histograma. Também é possível registrar uma lista das primeiras 500 métricas do Prometheus que forem descartadas e não enviadas ao CloudWatch por serem métricas de histograma.

Para ver se alguma métrica está sendo descartada, insira o comando a seguir:

```
kubectl logs -l "app=cwagent-prometheus" -n amazon-cloudwatch --tail=-1
```

Se alguma métrica estiver sendo descartada, você verá as seguintes linhas no arquivo `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`.

```
I! Drop Prometheus metrics with unsupported types. Only Gauge, Counter and Summary are
 supported.
I! Please enable CWAgent debug mode to view the first 500 dropped metrics
```

Se você vir essas linhas e quiser saber quais métricas estão sendo descartadas, siga as etapas a seguir.

Como registrar uma lista de métricas descartadas do Prometheus

1. Altere o atendente do CloudWatch para o modo de depuração adicionando as seguintes linhas em negrito ao arquivo `prometheus-eks.yaml` ou `prometheus-k8s.yaml` e salve o arquivo.

   ```
   {
         "agent": {
           "debug": true
         },
   ```

   Esta seção do arquivo deve ser semelhante ao seguinte:

   ```
   cwagentconfig.json: |
       {
   ```

```
      "agent": {
        "debug": true
      },
      "logs": {
        "metrics_collected": {
```

2. Reinstale o atendente do CloudWatch para habilitar o modo de depuração inserindo os comandos a seguir:

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
kubectl apply -f prometheus.yaml
```

   As métricas descartadas são registradas no pod do atendente do CloudWatch.

3. Para recuperar os logs do pod do atendente do CloudWatch, insira o comando a seguir:

```
kubectl logs -l "app=cwagent-prometheus" -n amazon-cloudwatch --tail=-1
```

   Ou, se você tiver a geração de logs do Fluentd do Container Insights instalada, os logs também serão salvos no grupo de logs do CloudWatch Logs /aws/containerinsights/***cluster_name***/application.

   Para consultar esses logs, você pode seguir as etapas para consultar os logs do aplicativo em Etapas gerais de solução de problemas no Amazon EKS.

Onde estão as métricas do Prometheus ingeridas como eventos de log do CloudWatch Logs?

O atendente do CloudWatch cria um fluxo de logs para cada configuração do trabalho de extração do Prometheus. Por exemplo, nos arquivos `prometheus-eks.yaml` e `prometheus-k8s.yaml`, a linha `job_name: 'kubernetes-pod-appmesh-envoy'` extrai conteúdo de métricas do App Mesh. O alvo do Prometheus é definido como `kubernetes-pod-appmesh-envoy`. Portanto, todas as métricas do App Mesh Prometheus são ingeridas como eventos do CloudWatch Logs no fluxo de logs kubernetes-pod-appmesh-envoy no grupo de logs chamado /aws/containerinsights/cluster-name/Prometheus.

Não visualizo métricas do Amazon EKS ou do Kubernetes Prometheus nas métricas do CloudWatch

Primeiro, verifique se as métricas do Prometheus são ingeridas como eventos de log no grupo de logs /aws/containerinsights/cluster-name/Prometheus. Use as informações em Onde estão as métricas do Prometheus ingeridas como eventos de log do CloudWatch Logs? para ajudar a verificar

o fluxo de logs de destino. Se o fluxo de logs não foi criado ou não houver novos eventos de log no fluxo de logs, confira o seguinte:

- Verifique se os endpoints do exportador de métricas do Prometheus estão configurados corretamente
- Verifique se as configurações de extração de conteúdo do Prometheus na seção `config map: cwagent-prometheus` do arquivo YAML do atendente do CloudWatch estão corretas. A configuração deve ser a mesma de um arquivo de configuração do Prometheus. Para obter mais informações, consulte <scrape_config> na documentação do Prometheus.

Se as métricas do Prometheus foram corretamente ingeridas como eventos de log, verifique se as configurações de formato de métrica incorporadas foram adicionadas aos eventos de log para gerar as métricas do CloudWatch.

```
"CloudWatchMetrics":[
   {
      "Metrics":[
         {
            "Name":"envoy_http_downstream_cx_destroy_remote_active_rq"
         }
      ],
      "Dimensions":[
         [
            "ClusterName",
            "Namespace"
         ]
      ],
      "Namespace":"ContainerInsights/Prometheus"
   }
],
```

Para obter mais informações sobre o formato de métrica incorporado, consulte Especificação: formato de métricas incorporadas .

Se não houver nenhum formato de métrica incorporado nos eventos de log, verifique se a seção `metric_declaration` está configurada corretamente na seção `config map: prometheus-cwagentconfig` do arquivo YAML de instalação do atendente do CloudWatch. Para ter mais informações, consulte Tutorial para adicionar um novo destino de extração do Prometheus: métricas do servidor de API do Prometheus.

Integração ao Application Insights

O Amazon CloudWatch Application Insights ajuda a monitorar suas aplicações, além de identificar e configurar as principais métricas, logs e alarmes nos recursos da aplicação e pilha de tecnologia. Para ter mais informações, consulte Amazon CloudWatch Application Insights.

Você pode habilitar o Application Insights para coletar dados adicionais de suas aplicações e microsserviços conteinerizados. Se ainda não fez isso, poderá habilitá-lo escolhendo Auto-configure Application Insights (Configurar automaticamente o Application Insights) abaixo da visualização de performance no painel Container Insights.

Se você já configurou o CloudWatch Application Insights para monitorar suas aplicações em contêiner, o painel do Application Insights aparecerá abaixo do painel do Container Insights.

Para obter mais informações sobre o Application Insights e aplicações em contêiner, consulte Habilitar o Application Insights para monitoramento de recursos do Amazon ECS e do Amazon EKS.

Ver eventos do ciclo de vida do Amazon ECS no Container Insights

É possível visualizar os eventos do ciclo de vida do Amazon ECS no console do Container Insights. Isso ajuda a correlacionar suas métricas, logs e eventos de contêiner em uma única visualização para oferecer uma visibilidade operacional mais completa.

Os eventos incluem eventos de alteração de estado da instância de contêiner, eventos de alteração de estado de tarefas e eventos de ação de serviços. São enviados automaticamente pelo Amazon ECS ao Amazon EventBridge e também são coletados no CloudWatch no formato de log de eventos. Para obter mais informações sobre esses eventos, consulte Eventos do Amazon ECS.

Os preços padrão do Container Insights se aplicam a eventos de ciclo de vida do Amazon ECS. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

Para configurar a tabela de eventos do ciclo de vida e criar regras para um cluster, é necessário ter as permissões `events:PutRule`, `events:PutTargets` e `logs:CreateLogGroup`. Você também deve se certificar de que há uma política de recursos que permite que o EventBridge crie o fluxo de logs e envie os logs para o CloudWatch Logs. Se essa política de recursos não existir, você pode digitar o seguinte comando para criá-la:

```
aws --region region logs put-resource-policy --policy-name 'EventBridgeCloudWatchLogs'
 --policy-document '{
    "Statement": [
        {
```

```
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": ["events.amazonaws.com", "delivery.logs.amazonaws.com"]
            },
            "Resource": "arn:aws:logs:region:account-id:log-group:/aws/events/ecs/
containerinsights/*:*",
            "Sid": "TrustEventBridgeToStoreECSLifecycleLogEvents"
        }
    ],
    "Version": "2012-10-17"
}'
```

Você pode usar o comando a seguir para verificar se você já possui essa política e para confirmar se a anexação funcionou corretamente.

```
aws logs describe-resource-policies --region region --output json
```

Para visualizar a tabela de eventos do ciclo de vida, é necessário ter as permissões `events:DescribeRule`, `events:ListTargetsByRule` e `logs:DescribeLogGroups`.

Como visualizar os eventos do ciclo de vida do Amazon ECS no console do CloudWatch Container Insights

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Escolha Insights, Container Insights.
3. Escolha Exibir painéis de desempenho.
4. No próximo menu suspenso, escolha entre ECS Clusters (Clusters do ECS), ECS Services (Serviços do ECS) ou ECS Tasks (Tarefas do ECS).
5. Se você escolheu ECS Services (Serviços do ECS) ou ECS Tasks (Tarefas do ECS) na etapa anterior, escolha a guia Lifecycle events (Eventos do ciclo de vida).
6. Na parte inferior da página, se você visualizar Configure lifecycle events (Configurar eventos do ciclo de vida), escolha essa opção para criar regras do EventBridge para o cluster.

   Os eventos são exibidos abaixo dos painéis do Container Insights e acima da seção Application Insights. Para executar mais análises e criar outras visualizações sobre esses eventos, escolha

View in Logs Insights (Exibir no Logs Insights) na tabela Lifecycle Events (Eventos do ciclo de vida).

Solução de problemas do Container Insights

As seções a seguir podem ajudar se você estiver tendo problemas com o Container Insights.

Falha na implantação no Amazon EKS ou no Kubernetes

Se o atendente não for implantado corretamente em um cluster do Kubernetes, tente o seguinte:

- Execute o comando a seguir para obter a lista de pods.

```
kubectl get pods -n amazon-cloudwatch
```

- Execute o comando a seguir e verifique os eventos na parte inferior da saída.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- Execute o comando a seguir para verificar os logs.

```
kubectl logs pod-name -n amazon-cloudwatch
```

Pânico não autorizado: não é possível recuperar dados cadvisor do kubelet

Se a implantação falhar com o erro `Unauthorized panic: Cannot retrieve cadvisor data from kubelet`, o kubelet talvez não tenha o modo de autorização Webhook habilitado. Esse modo é necessário para o Container Insights. Para ter mais informações, consulte Verifique os pré-requisitos do .

Implantar o Container Insights em um cluster excluído e recriado no Amazon ECS

Se você excluir um cluster existente do Amazon ECS que não tenha o Container Insights habilitado e recriá-lo com o mesmo nome, não será possível habilitar o Container Insights nesse novo cluster ao recriá-lo. Você pode habilitá-lo recriando-o e inserindo o seguinte comando:

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

Erro de endpoint inválido

Se você vir uma mensagem de erro semelhante à seguinte, verifique se você substituiu todos os espaços reservados, como *cluster-name* e *region-name* nos comandos que você está usando pelas informações corretas para sua implantação.

```
"log": "2020-04-02T08:36:16Z E! cloudwatchlogs: code: InvalidEndpointURL, message:
 invalid endpoint uri, original error: &url.Error{Op:\"parse\", URL:\"https://
logs.{{region_name}}.amazonaws.com/\", Err:\"{\"}, &awserr.baseError{code:
\"InvalidEndpointURL\", message:\"invalid endpoint uri\", errs:[]error{(*url.Error)
(0xc0008723c0)}}\n",
```

As métricas não são exibidas no console

Se você não vir nenhuma métrica do Container Insights no AWS Management Console, certifique-se de que você tenha concluído a configuração do Container Insights. As métricas não serão exibidas antes de o Container Insights ser configurado completamente. Para ter mais informações, consulte Configurar o Container Insights.

Métricas de pod ausentes no Amazon EKS ou no Kubernetes após a atualização do cluster

Esta seção pode ser útil se todas ou algumas métricas de pods estiverem ausentes depois de você implantar o agente do CloudWatch como daemonset em um cluster novo ou atualizado, ou se você vir um log de erros com a mensagem `W! No pod metric collected`.

Esses erros podem ser causados por alterações no runtime do contêiner, como containerd ou o driver cgroup systemd do docker. Normalmente, você pode resolver isso atualizando seu manifesto de implantação para que o soquete containerd do host seja montado no contêiner. Veja o exemplo a seguir:

```
# For full example see https://github.com/aws-samples/amazon-cloudwatch-container-
insights/blob/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cwagent/cwagent-daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: cloudwatch-agent
  namespace: amazon-cloudwatch
```

```
spec:
  template:
    spec:
      containers:
        - name: cloudwatch-agent
# ...
          # Don't change the mountPath
          volumeMounts:
# ...
            - name: dockersock
              mountPath: /var/run/docker.sock
              readOnly: true
            - name: varlibdocker
              mountPath: /var/lib/docker
              readOnly: true
            - name: containerdsock # NEW mount
              mountPath: /run/containerd/containerd.sock
              readOnly: true
# ...
      volumes:
# ...
        - name: dockersock
          hostPath:
            path: /var/run/docker.sock
        - name: varlibdocker
          hostPath:
            path: /var/lib/docker
        - name: containerdsock # NEW volume
          hostPath:
            path: /run/containerd/containerd.sock
```

Nenhuma métrica de pod ao usar Bottlerocket para o Amazon EKS

O Bottlerocket é um sistema operacional de código aberto baseado em Linux que foi criado especificamente pela AWS para executar contêineres.

O Bottlerocket usa um caminho de `containerd` diferente no host, então é necessário alterar os volumes para o local dele. Se não fizer isso, você verá um erro nos logs que inclui `W! No pod metric collected`. Veja o exemplo a seguir.

```
volumes:
  # ...
    - name: containerdsock
```

```
      hostPath:
        # path: /run/containerd/containerd.sock
        # bottlerocket does not mount containerd sock at normal place
        # https://github.com/bottlerocket-os/bottlerocket/
commit/91810c85b83ff4c3660b496e243ef8b55df0973b
        path: /run/dockershim.sock
```

Nenhuma métrica do filesystem de contêiner ao usar o runtime do containerd para Amazon EKS ou Kubernetes

Esse é um problema conhecido, e colaboradores da comunidade estão trabalhando nele. Para obter mais informações, consulte Métrica de uso de disco para conteinerd e métricas do sistema de arquivos de contêiner não são compatíves com o cadvisor para containerd no GitHub.

Aumento inesperado do volume de log do atendente do CloudWatch ao coletar métricas do Prometheus

Essa foi uma regressão introduzida na versão 1.247347.6b250880 do atendente do CloudWatch. Essa regressão já foi corrigida em versões mais recentes do atendente. Seu impacto foi limitado a cenários em que os clientes coletavam os logs do próprio atendente do CloudWatch e estavam usando o Prometheus. Para obter mais informações, consulte atendente [do prometheus] está imprimindo todas as métricas extraídas no log no GitHub.

A imagem do Docker mais recente mencionada nas notas de release não foi encontrada no Dockerhub

Atualizamos a nota de release e a etiqueta no Github antes de iniciarmos a versão real internamente. Normalmente, leva de 1 a 2 semanas para ver a imagem do Docker mais recente nos registros depois de bater o número da versão no Github. Não há versão noturna para a imagem do contêiner do atendente do CloudWatch. É possível criar a imagem diretamente da origem no seguinte local: https://github.com/aws/amazon-cloudwatch-agent/tree/main/amazon-cloudwatch-container-insights/cloudwatch-agent-dockerfile

Erro CrashLoopBackoff no atendente do CloudWatch

Ao ver um erro `CrashLoopBackOff` do atendente do CloudWatch, verifique se suas permissões do IAM estão definidas corretamente. Para ter mais informações, consulte Verifique os pré-requisitos do .

Agente do CloudWatch ou pod do Fluentd travado em pendente

Se você tiver um agente do CloudWatch ou pod do Fluentd travado em `Pending` ou com um erro `FailedScheduling`, determine se seus nós têm recursos de computação suficientes com base no número de núcleos e na quantidade de RAM exigida pelos agentes. Use o comando a seguir para descrever o pod:

```
kubectl describe pod cloudwatch-agent-85ppg -n amazon-cloudwatch
```

Criar sua própria imagem do Docker do atendente do CloudWatch

Você pode criar sua própria imagem do Docker do atendente do CloudWatch fazendo referência ao Dockerfile localizado em https://github.com/aws-samples/amazon-cloudwatch-container-insights/blob/latest/cloudwatch-agent-dockerfile/Dockerfile.

O Dockerfile oferece suporte para criar imagens multiarquitetura diretamente usando `docker buildx`.

Implantar outros recursos do atendente do CloudWatch nos contêineres

Você pode implantar recursos de monitoramento adicionais em seus contêineres usando o atendente do CloudWatch. Esses recursos incluem o seguinte:

- Formato de métrica incorporado: para obter mais informações, consulte Incorporação de métricas em logs.
- StatsD: para obter mais informações, consulte Recuperar métricas personalizadas com o StatsD .

Instruções e arquivos necessários estão localizados nos seguintes locais do GitHub:

- Para contêineres do Amazon ECS, consulte Exemplo de definições de tarefas do Amazon ECS com base nos modos de implantação.
- Para contêineres do Amazon EKS e do Kubernetes, consulte Exemplo de arquivos YAML do Kubernetes com base nos modos de implantação.

Lambda Insights

O Lambda Insights do CloudWatch Lambda é uma solução de monitoramento e solução de problemas para aplicações sem servidor em execução no AWS Lambda. A solução coleta, agrega e

resume métricas no nível do sistema, incluindo tempo da CPU, memória, disco e rede. Ele também coleta, agrega e resume informações de diagnóstico, como inicializações a frio e desligamentos do operador do Lambda para ajudar a isolar problemas com as funções do Lambda e resolvê-los rapidamente.

O Lambda Insights usa uma nova extensão do CloudWatch Lambda, que é fornecida como uma camada do Lambda. Quando você instala essa extensão em uma função Lambda, ela coleta métricas no nível do sistema e emite um único evento de log de performance para cada invocação dessa função Lambda. O CloudWatch usa formatação métrica incorporada para extrair métricas dos eventos de log.

Para obter mais informações sobre as extensões do Lambda, consulte Usar extensões do AWS Lambda. Para obter mais informações sobre o formato de métrica incorporado, consulte Incorporação de métricas em logs.

Você pode usar o Lambda Insights com qualquer função Lambda que utilize um runtime do Lambda compatível com extensões do Lambda. Para obter uma lista desses tempos de execução, consulte API de extensões do Lambda.

Definição de preço

Para cada função do Lambda habilitada para o Lambda Insights, você paga apenas pelo que usar com relação a métricas e logs. Para obter exemplos de preço, consulte Preço do Amazon CloudWatch.

O runtime consumido pela extensão do Lambda (em incrementos de 1 ms) será cobrado. Para obter mais informações sobre preços do Lambda, consulte Preço do AWS Lambda.

Conceitos básicos do Lambda Insights

Para habilitar o Lambda Insights em uma função Lambda, é possível usar uma alternância de um clique no console do Lambda. Se preferir, você pode usar a AWS CLI, o AWS CloudFormation, a AWS Serverless Application ModelCLI ou o AWS Cloud Development Kit (AWS CDK).

As seções a seguir fornecem instruções detalhadas para concluir essas etapas.

Tópicos

- Usar a AWS SAM CLI para habilitar o Lambda Insights em uma função existente do Lambda
- Usar o AWS CloudFormation para habilitar o Lambda Insights em uma função existente do Lambda
- Usar a AWS CDK para habilitar o Lambda Insights em uma função existente do Lambda
- Usar o Serverless Framework para habilitar o Lambda Insights em uma função existente do Lambda
- Habilitar o Lambda Insights em uma implantação de imagem de contêiner do Lambda

Versões disponíveis da extensão do Lambda Insights

Esta seção relaciona as versões da extensão do Lambda Insights e os ARNs a serem usados para essas extensões em cada região da AWS.

Tópicos

- plataformas x86-64
- Plataformas ARM64

plataformas x86-64

Esta seção relaciona as versões da extensão do Lambda Insights para plataformas x84-64, bem como os ARNs a serem usados para essas extensões em cada região da AWS.

> **Important**
>
> As extensões do Lambda Insights na versão 1.0.317.0 e em versões posteriores não são compatíveis com o Amazon Linux 1.

1.0.317.0

A versão 1.0.317.0 inclui a remoção do suporte para a plataforma Amazon Linux 1 e correções de bugs. Além disso, há a inclusão de suporte para regiões AWS GovCloud (US).

ARNs para a versão 1.0.317.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:52`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:52`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:52`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:52`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:43`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:43`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:25`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:29`
Ásia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:20`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:50`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:33`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:51`

Região	ARN
Ásia-Pacífico (Singapura)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:52
Ásia-Pacífico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:52
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:79
Canadá (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:51
Oeste do Canadá (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:12
China (Pequim)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:42
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:42
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:52
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:52
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:52
Europa (Milão)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:43
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:51
Europa (Espanha)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:27

Região	ARN
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:49`
Europa (Zurique)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:26`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:20`
Oriente Médio (Barém)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:43`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:26`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:51`
AWS GovCloud (Leste dos EUA)	`arn:aws-us-gov:lambda:us-gov-east-1:122132214140:layer:LambdaInsightsExtension:19`
AWS GovCloud (Oeste dos EUA)	`arn:aws-us-gov:lambda:us-gov-west-1:751350123760:layer:LambdaInsightsExtension:19`

1.0.295.0

A versão 1.0.295.0 inclui atualizações de dependência para todos os runtimes compatíveis.

ARNs para a versão 1.0.295.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:51`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:51`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:51`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:51`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:42`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:42`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:24`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:28`
Ásia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:19`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:49`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:32`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:50`

Região	ARN
Ásia-Pacífico (Singapura)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:51
Ásia-Pacífico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:51
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:78
Canadá (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:50
Oeste do Canadá (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:11
China (Pequim)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:41
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:41
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:51
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:51
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:51
Europa (Milão)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:42
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:50
Europa (Espanha)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:26

Região	ARN
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:48`
Europa (Zurique)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:25`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:19`
Oriente Médio (Barém)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:42`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:25`
América do Sul (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:50`

1.0.275.0

A versão 1.0.275.0 inclui atualizações importantes de dependência para todos os runtimes compatíveis.

ARNs para a versão 1.0.275.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:49`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:49`

Região	ARN
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:49`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:49`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:40`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:40`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:22`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:26`
Ásia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:17`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:47`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:30`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:48`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:49`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:49`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:76`

Região	ARN
Canadá (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:48
Oeste do Canadá (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:9
China (Pequim)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:39
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:39
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:49
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:49
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:49
Europa (Milão)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:40
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:48
Europa (Espanha)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:24
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:46
Europa (Zurique)	arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:23
Israel (Tel Aviv)	arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:17

Região	ARN
Oriente Médio (Barém)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:40`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:23`
América do Sul (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:48`

1.0.273.0

A versão 1.0.273.0 inclui correções de bugs importantes para todos os runtimes compatíveis e adiciona suporte para a região Oeste do Canadá (Calgary).

ARNs para a versão 1.0.273.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:45`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:45`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:45`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:45`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:35`

Região	ARN
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:35`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:17`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:21`
Ásia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:12`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:43`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:26`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:44`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:45`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:45`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:72`
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:44`
Oeste do Canadá (Calgary)	`arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:4`
China (Pequim)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:36`

Região	ARN
China (Ningxia);	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:36`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:45`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:45`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:45`
Europa (Milão)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:35`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:44`
Europa (Espanha)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:19`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:42`
Europa (Zurique)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:17`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:12`
Oriente Médio (Barém)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:35`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:18`

Região	ARN
América do Sul (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:44`

1.0.229.0

A versão 1.0.229.0 inclui correções de bugs importantes para todos os runtimes compatíveis e adiciona suporte para a região de Israel (Tel Aviv).

ARNs para a versão 1.0.229.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:38`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:38`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:38`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:38`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:28`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:28`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:10`

Região	ARN
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:14`
Ásia-Pacífico (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:5`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:36`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:19`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:37`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:38`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:38`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:60`
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:37`
China (Pequim)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:29`
China (Ningxia);	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:29`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:38`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:38`

Região	ARN
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:38`
Europa (Milão)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:28`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:37`
Europa (Espanha)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:12`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:35`
Europa (Zurique)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:11`
Israel (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:5`
Oriente Médio (Barém)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:28`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:11`
América do Sul (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:37`

1.0.178.0

A versão 1.0.178.0 adiciona suporte às regiões da AWS a seguir.

- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Jacarta)

- Europa (Espanha)
- Europa (Zurique)
- Oriente Médio (Emirados Árabes Unidos)

ARNs para a versão 1.0.178.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:35`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:33`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:33`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:33`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:25`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:25`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:8`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:11`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:31`

Região	ARN
Ásia-Pacífico (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:2
Ásia-Pacífico (Seul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:32
Ásia-Pacífico (Singapura)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:33
Ásia-Pacífico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:33
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:50
Canadá (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:32
China (Pequim)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:26
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:26
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:35
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:33
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:33
Europa (Milão)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:25
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:32

Região	ARN
Europa (Espanha)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:10`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:30`
Europa (Zurique)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:7`
Oriente Médio (Barém)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:25`
Oriente Médio (Emirados Árabes Unidos)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:9`
América do Sul (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:32`

1.0.143.0

A versão 1.0.143.0 inclui correções de bugs em compatibilidade com Python 3.7 e Go 1.x. O runtime do Python 3.6 Lambda está sendo substituído. Para obter mais informações, consulte Tempos de execução do Lambda.

ARNs para a versão 1.0.143.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:21`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:21`

Região	ARN
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:20`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:21`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:13`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:13`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:21`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:2`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:20`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:21`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:21`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:32`
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:20`
China (Pequim)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:14`
China (Ningxia);	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:14`

Região	ARN
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:21`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:21`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:21`
Europa (Milão)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:13`
Europa (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:20`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:20`
Oriente Médio (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:13`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:20`

1.0.135.0

A versão 1.0.135.0 inclui correções de bugs relacionados a como o Lambda Insights coleta e relata o uso do descritor de disco e arquivo. Nas versões anteriores da extensão, a métrica `tmp_free` relatava o espaço livre máximo no diretório `/tmp`enquanto uma função era executada. Essa versão altera a métrica para relatar o valor mínimo, tornando-a mais útil ao avaliar o uso do disco. Para obter mais informações sobre cotas de armazenamento de diretórios `tmp`, consulte Cotas do Lambda.

A versão 1.0.135.0 agora também relata o uso do descritor de arquivos (`fd_use` e `fd_max`) como o valor máximo entre os processos, em vez de relatar o nível do sistema operacional.

ARNs para a versão 1.0.135.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:18`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:18`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:18`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:18`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:11`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:11`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:18`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:1`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:18`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:18`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:18`

Região	ARN
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:25
Canadá (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:18
China (Pequim)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:11
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:11
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:18
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:18
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:18
Europa (Milão)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:11
Europa (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:18
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:18
Oriente Médio (Bahrein)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:11
South America (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:18

1.0.119.0

ARNs para a versão 1.0.119.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:16`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:16`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:16`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:16`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:9`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:9`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:16`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:16`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:16`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:16`

Região	ARN
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:23`
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:16`
China (Pequim)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:9`
China (Ningxia);	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:9`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:16`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:16`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:16`
Europa (Milão)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:9`
Europa (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:16`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:16`
Oriente Médio (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:9`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:16`

1.0.98.0

Essa versão elimina registros desnecessários e resolve um problema com as chamadas locais da AWS Serverless Application Model CLI. Para obter mais informações sobre esse problema, consulteAdicionar resultados do LambdaInsightsExTension em tempo limite com 'sam local invoke'.

ARNs para a versão 1.0.98.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:14`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:14`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:14`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:14`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:8`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:8`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:14`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:14`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:14`

Região	ARN
Ásia-Pacífico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:14
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:14
Canadá (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:14
China (Pequim)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:8
China (Ningxia);	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:8
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:14
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:14
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:14
Europa (Milão)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:8
Europa (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:14
Europa (Estocolmo)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:14
Oriente Médio (Bahrein)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:8
South America (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:14

1.0.89.0

Essa versão corrige o carimbo de data/hora do evento de performance para representar sempre o início da chamada da função.

ARNs para a versão 1.0.89.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:12`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:12`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:12`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:12`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:12`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:12`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:12`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:12`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:12`

Região	ARN
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:12`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:12`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:12`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:12`
Europa (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:12`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:12`
América do Sul (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:12`

1.0.86.0

Na versão 1.0.54.0 da extensão, as métricas de memória às vezes eram relatadas incorretamente e às vezes eram superiores a 100%. A versão 1.0.86.0 corrige o problema de medição de memória usando os mesmos dados de eventos como métricas da plataforma Lambda. Isso significa que é possível ver uma mudança dramática nos valores de métrica da memória registrada. Isso é possível com a nova API do Lambda Logs. Ela fornece uma medição mais precisa do uso de memória sandbox do Lambda. Porém, não se esqueça que a API do Lambda Logs não poderá entregar eventos de relatório de plataforma se uma sandbox de função expirar e for posteriormente girada para baixo. Nesse caso. O Lambda Insights não conseguirá gravar as métricas de invocação. Para obter mais informações sobre a API do Lambda Logs, consulte API do AWS Lambda Logs.

Novos recursos na versão 1.0.86.0

- Usa a API do Lambda Logs para corrigir a métrica de memória. Assim, resolve o problema anterior em que as estatísticas de memória eram superiores a 100%.

- Apresenta `Init Duration` como uma nova métrica do CloudWatch.
- Usa o ARN de invocação para adicionar uma dimensão de versão para aliases e versões invocadas. Caso esteja usando aliases ou versões do Lambda para obter implantações incrementais (como implantações azul-verde), você poderá exibir suas métricas com base no alias invocado. A versão não será aplicada se a função não usar um alias ou uma versão. Para obter mais informações, consulte Aliases de funções do Lambda.
- Adiciona `billed_mb_ms field` aos eventos de performance para exibir o custo por invocação. Não considera nenhum custo associado à simultaneidade provisionada.
- Adiciona os campo `billed_duration` e `duration` aos eventos de performance.

ARNs para a versão 1.0.86.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:11`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:11`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:11`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:11`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:11`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:11`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:11`

Região	ARN
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:11`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:11`
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:11`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:11`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:11`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:11`
Europa (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:11`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:11`
América do Sul (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:11`

1.0.54.0

A versão 1.0.54.0 foi a versão inicial da extensão do Lambda Insights.

ARNs para a versão 1.0.54.0

A tabela a seguir reaciona os ARNs a serem usados para essa versão da extensão em cada região da AWS onde está disponível.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:2`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:2`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:2`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:2`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:2`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:2`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:2`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:2`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:2`
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:2`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:2`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:2`

Região	ARN
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:2`
Europa (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:2`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:2`
América do Sul (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:2`

Plataformas ARM64

Esta seção relaciona as versões da extensão do Lambda Insights para ARM64 e os ARNs a serem usados para essas extensões em cada região da AWS.

> **Important**
>
> As extensões do Lambda Insights na versão 1.0.317.0 e em versões posteriores não são compatíveis com o Amazon Linux 1.

1.0.317.0

A versão 1.0.317.0 inclui a remoção do suporte para a plataforma Amazon Linux 1 e correções de bugs. Além disso, há a inclusão de suporte para regiões AWS GovCloud (US).

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:21`

Região	ARN
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:19`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:17`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:17`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:5`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:17`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:21`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:16`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:30`
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`

Região	ARN
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Europa (Milão)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:17`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Europa (Espanha)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:5`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Oriente Médio (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:17`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`
AWS GovCloud (Leste dos EUA)	`arn:aws-us-gov:lambda:us-gov-east-1:122132214140:layer:LambdaInsightsExtension-Arm64:1`
AWS GovCloud (Oeste dos EUA)	`arn:aws-us-gov:lambda:us-gov-west-1:751350123760:layer:LambdaInsightsExtension-Arm64:1`

1.0.295.0

A versão 1.0.295.0 inclui atualizações de dependência para todos os runtimes compatíveis.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:20`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:16`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:16`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:4`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:16`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:20`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:15`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`

Região	ARN
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:29`
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Europa (Milão)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:16`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Europa (Espanha)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:4`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Oriente Médio (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:16`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`

1.0.275.0

A versão 1.0.275.0 inclui correções de erros para todos os runtimes compatíveis e para as regiões Europa (Espanha) e Ásia-Pacífico (Hyderabad).

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:16`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:14`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:14`
Ásia-Pacífico (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:2`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:14`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:13`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:15`

Região	ARN
Ásia-Pacífico (Singapura)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Ásia-Pacífico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:16
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:27
Canadá (Central)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:14
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europa (Milão)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:14
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:14
Europa (Espanha)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:2
Europe (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:14
Oriente Médio (Bahrein)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:14
South America (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:14

1.0.273.0

A versão 1.0.273.0 inclui correções de erros para todos os runtimes compatíveis.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:9`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:12`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:9`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:9`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:9`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:9`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:11`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`

Região	ARN
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:23`
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:10`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Europa (Milão)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:9`
Europa (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:10`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:10`
Oriente Médio (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:9`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:10`

1.0.229.0

A versão 1.0.229.0 inclui correções de erros para todos os runtimes compatíveis. Ela também adiciona suporte para as regiões a seguir:

- Oeste dos EUA (N. da Califórnia)
- Africa (Cape Town)
- Ásia-Pacífico (Hong Kong)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Osaka)
- Ásia-Pacífico (Seul)
- Canadá (Central)
- Europe (Milan)
- Europa (Paris)
- Europa (Estocolmo)
- Oriente Médio (Bahrein)
- South America (São Paulo)

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:7`
Oeste dos EUA (N. da Califórnia)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
África (Cidade do Cabo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:2`
Ásia-Pacífico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:2`
Ásia-Pacífico (Jacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:2`

Região	ARN
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:7`
Ásia-Pacífico (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:2`
Ásia-Pacífico (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:4`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Ásia-Pacífico (Tóquio)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:11`
Canadá (Central)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Europa (Espanha)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:2`
Europa (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:3`
Europa (Estocolmo)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`

Região	ARN
Oriente Médio (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:2`
South America (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`

1.0.135.0

A versão 1.0.135.0 inclui correções de bugs relacionados a como o Lambda Insights coleta e relata o uso do descritor de disco e arquivo. Nas versões anteriores da extensão, a métrica `tmp_free` relatava o espaço livre máximo no diretório `/tmp`enquanto uma função era executada. Essa versão altera a métrica para relatar o valor mínimo, tornando-a mais útil ao avaliar o uso do disco. Para obter mais informações sobre cotas de armazenamento de diretórios `tmp`, consulte Cotas do Lambda.

A versão 1.0.135.0 agora também relata o uso do descritor de arquivos (`fd_use` e `fd_max`) como o valor máximo entre os processos, em vez de relatar o nível do sistema operacional.

Região	ARN
Leste dos EUA (Norte da Virgínia)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Leste dos EUA (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Oeste dos EUA (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Ásia-Pacífico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Ásia-Pacífico (Singapura)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Ásia-Pacífico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`

Região	ARN
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:2
Europa (Frankfurt)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:2
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:2
Europa (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:2

1.0.119.0

Região	ARN
Leste dos EUA (Norte da Virgínia)	arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:1
Leste dos EUA (Ohio)	arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:1
Oeste dos EUA (Oregon)	arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:1
Ásia-Pacífico (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:1
Ásia-Pacífico (Singapura)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:1
Ásia-Pacífico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:1
Ásia-Pacífico (Tóquio)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:1

Região	ARN
Europa (Frankfurt)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europa (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`

Usar o console para habilitar o Lambda Insights em uma função existente do Lambda

Siga as etapas a seguir no console do Lambda para habilitar o Lambda Insights em uma função existente do Lambda.

Para habilitar o Lambda Insights em uma função Lambda

1. Abra o console AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Escolha o nome de uma função e selecione a guia Configuration (Configuração) na tela a seguir.
3. Na guia Configuração, selecione Ferramentas de monitoramento e operações no menu de navegação à esquerda e escolha Editar.

   Você será direcionado para uma tela na qual poderá editar as ferramentas de monitoramento.
4. Em Monitoramento aprimorado do Lambda Insights, escolha Editar.
5. Em CloudWatch Lambda Insights, habilite Monitoramento aprimorado e escolha Salvar.

Usar a AWS CLI para habilitar o Lambda Insights em uma função existente do Lambda

Siga estas etapas para usar a AWS CLI para habilitar o Lambda Insights em uma função existente do Lambda.

Etapa 1: atualizar as permissões da função

Para atualizar as permissões da função

- Anexe a política do IAM gerenciada CloudWatchLambdaInsightsExecutionRolePolicy para a função de execução da função inserindo o comando a seguir.

```
aws iam attach-role-policy \
--role-name function-execution-role \
--policy-arn "arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy"
```

Etapa 2: instalar a extensão do Lambda

Instale a extensão do Lambda inserindo o comando a seguir. Substitua o valor do ARN pelo parâmetro `layers` com o ARN correspondente a sua região e à versão da extensão que você deseja usar. Para ter mais informações, consulte Versões disponíveis da extensão do Lambda Insights.

```
aws lambda update-function-configuration \
   --function-name function-name \
   --layers "arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:14"
```

Etapa 3: habilitar o endpoint da VPC do CloudWatch Logs

Essa etapa é necessária somente para funções em execução em uma sub-rede privada sem acesso à Internet e caso você ainda não tenha configurado um endpoint da Virtual Private Cloud (VPC) do CloudWatch Logs.

Caso precise realizar esta etapa, insira o seguinte comando, substituindo os espaços reservados por informações para sua VPC.

Para obter mais informações, consulte Usar o CloudWatch Logs com endpoints da VPC de interface.

```
aws ec2 create-vpc-endpoint \
--vpc-id vpcId \
--vpc-endpoint-type Interface \
--service-name com.amazonaws.region.logs \
--subnet-id subnetId
--security-group-id securitygroupId
```

Usar a AWS SAM CLI para habilitar o Lambda Insights em uma função existente do Lambda

Siga estas etapas para usar a AWS SAM AWS CLI para habilitar o Lambda Insights em uma função existente do Lambda.

Se você ainda não tem a versão mais recente da AWS SAM CLI instalada, primeiro é necessário instalá-la ou atualizá-la. Para obter mais informações, consulte Instalar a AWS SAM CLI.

Etapa 1: instalar a camada

Para disponibilizar a extensão Lambda Insights a todas as funções do Lambda, adicione uma propriedade `Layers` à seção `Globals` de seu modelo do SAM com o ARN da camada do Lambda Insights. O exemplo abaixo usa a camada para a versão inicial do Lambda Insights. Para obter a versão mais recente da camada de extensão do Lambda Insights, consulte Versões disponíveis da extensão do Lambda Insights.

```
Globals:
  Function:
    Layers:
      - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Para habilitar essa camada para apenas uma única função, adicione a propriedade `Layers` à função, conforme apresentado neste exemplo.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Layers:
        - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Etapa 2: adicionar a política gerenciada

Para cada função, adicione a política do IAM CloudWatchLambdaInsightsExecutionRolePolicy.

Como o AWS SAM não oferece suporte a políticas globais, é necessário habilitá-las em cada função individualmente, conforme apresentado neste exemplo. Para obter mais informações sobre globais, consulte Seção de globais.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
```

```
      Policies:
        - CloudWatchLambdaInsightsExecutionRolePolicy
```

Invocar localmente

A AWS SAM CLI é compatível com extensões do Lambda. Porém, cada invocação executada localmente redefine o ambiente do runtime. Os dados do Lambda Insights não estarão disponíveis a partir de invocações locais porque o runtime é reiniciado sem um evento de desligamento. Para obter mais informações, consulteRelease 1.6.0: adicionar suporte para testes locais de extensões doAWS Lambda.

Solução de problemas

Para solucionar problemas de instalação do Lambda Insights, adicione a seguinte variável de ambiente a sua função Lambda para habilitar o log de depuração.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Environment:
        Variables:
          LAMBDA_INSIGHTS_LOG_LEVEL: info
```

Usar o AWS CloudFormation para habilitar o Lambda Insights em uma função existente do Lambda

Siga estas etapas para usar o AWS CloudFormation para habilitar o Lambda Insights em uma função existente do Lambda.

Etapa 1: instalar a camada

Adicione a camada do Lambda Insights à ´propriedade `Layers` dentro do ARN da camada do Lambda Insights. O exemplo abaixo usa a camada para a versão inicial do Lambda Insights. Para obter a versão mais recente da camada de extensão do Lambda Insights, consulte Versões disponíveis da extensão do Lambda Insights.

```
Resources:
  MyFunction:
```

```
    Type: AWS::Lambda::Function
    Properties:
      Layers:
        - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Etapa 2: adicionar a política gerenciada

Adicione a política do IAM CloudWatchLambdaInsightsExecutionRolePolicy a sua função de execução.

```
Resources:
  MyFunctionExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy'
```

Etapa 3: (opcional) adicionar endpoints da VPC

Essa etapa é necessária somente para funções em execução em uma sub-rede privada sem acesso à Internet e caso você ainda não tenha configurado um endpoint da Virtual Private Cloud (VPC) do CloudWatch Logs. Para obter mais informações, consulte Usar o CloudWatch Logs com endpoints da VPC de interface.

```
Resources:
  CloudWatchLogsVpcPrivateEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      PrivateDnsEnabled: 'true'
      VpcEndpointType: Interface
      VpcId: !Ref: VPC
      ServiceName: !Sub com.amazonaws.${AWS::Region}.logs
      SecurityGroupIds:
        - !Ref InterfaceVpcEndpointSecurityGroup
      SubnetIds:
        - !Ref PublicSubnet01
        - !Ref PublicSubnet02
        - !Ref PublicSubnet03
```

Usar a AWS CDK para habilitar o Lambda Insights em uma função existente do Lambda

Siga estas etapas para usar a AWS CDK para habilitar o Lambda Insights em uma função existente do Lambda. Para utilizar essas etapas, você já deve estar usando o AWS CDK para gerenciar seus recursos.

Os comandos desta seção estão em TypeScript.

Primeiro, atualize as permissões da função.

```
executionRole.addManagedPolicy(
 ManagedPolicy.fromAwsManagedPolicyName('CloudWatchLambdaInsightsExecutionRolePolicy')
);
```

Em seguida, instale a extensão na função Lambda. Substitua o valor do ARN pelo parâmetro `layerArn` com o ARN correspondente a sua região e à versão da extensão que você deseja usar. Para ter mais informações, consulte Versões disponíveis da extensão do Lambda Insights.

```
import lambda = require('@aws-cdk/aws-lambda');
const layerArn = 'arn:aws:lambda:us-
west-1:580247275435:layer:LambdaInsightsExtension:14';
const layer = lambda.LayerVersion.fromLayerVersionArn(this, 'LayerFromArn', layerArn);
```

Se necessário, habilite o endpoint da Virtual Private Cloud (VPC) para o CloudWatch Logs. Essa etapa é necessária somente para funções em execução em uma sub-rede privada sem acesso à Internet e caso você ainda não tenha configurado um endpoint da VPC do CloudWatch Logs.

```
const cloudWatchLogsEndpoint = vpc.addInterfaceEndpoint('cwl-gateway', {
  service: InterfaceVpcEndpointAwsService.CLOUDWATCH_LOGS,
});

cloudWatchLogsEndpoint.connections.allowDefaultPortFromAnyIpv4();
```

Usar o Serverless Framework para habilitar o Lambda Insights em uma função existente do Lambda

Siga estas etapas para usar o Serverless Framework para habilitar o Lambda Insights em uma função existente do Lambda. Para obter mais informações sobre o Serverless Framework, acesse serverless.com.

Isso é feito por meio de um plugin do Lambda Insights para o Serverless. Para obter mais informações, consulte serverless-plugin-lambda-insights.

Se você ainda não tem a versão mais recente da interface de linha de comando do Serverless instalada, primeiro é necessário instalá-la ou atualizá-la. Para obter mais informações, consulte Comece a usar o Serverless Framework Open Source e a AWS.

Para usar o Serverless Framework para habilitar o Lambda Insights em uma função Lambda

1. Instale o plugin do Serverless para o Lambda Insights executando o seguinte comando no diretório do Serverless:

```
npm install --save-dev serverless-plugin-lambda-insights
```

2. Em seu arquivo `serverless.yml`, adicione o plugin na seção `plugins`, conforme o exemplo:

```
provider:
  name: aws
plugins:
  - serverless-plugin-lambda-insights
```

3. Habilite o Lambda Insights.
   - É possível habilitar o Lambda Insights individualmente por função adicionando a seguinte propriedade ao arquivo serverless.yml

```
functions:
  myLambdaFunction:
    handler: src/app/index.handler
    lambdaInsights: true #enables Lambda Insights for this function
```

   - Você pode habilitar o Lambda Insights para todas as funções dentro do arquivo `serverless.yml` adicionando a seguinte seção personalizada:

```
custom:
```

```
  lambdaInsights:
    defaultLambdaInsights: true #enables Lambda Insights for all functions
```

4. Implante novamente o serviço do Serveless inserindo este comando:

```
serverless deploy
```

Ele implantará novamente todas as funções e habilitará o Lambda Insights para as funções que você especificou. Habilitará o Lambda Insights adicionando a camada do Lambda Insights e anexando as permissões necessárias usando a política do IAM `arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy`.

Habilitar o Lambda Insights em uma implantação de imagem de contêiner do Lambda

Para habilitar o Lambda Insights em uma função do Lambda implantada como uma imagem de contêiner, adicione linhas ao Dockerfile. Essas linhas instalam o agente do Lambda Insights como uma extensão em sua imagem de contêiner. As linhas a serem adicionadas são diferentes para contêineres x86-64 e contêineres ARM64.

> **Note**
>
> O agente do Lambda Insights só é compatível com tempos de execução do Lambda que usam o Amazon Linux 2.

Tópicos

- Implantação de imagem de contêiner x86-64
- Implantação de imagem de contêiner ARM64

Implantação de imagem de contêiner x86-64

Para habilitar o Lambda Insights em uma função do Lambda implantada como uma imagem de contêiner em execução em um contêiner X86-64, adicione as linhas a seguir ao Dockerfile. Essas linhas instalam o agente do Lambda Insights como uma extensão em sua imagem de contêiner.

```
RUN curl -O https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm && \
```

```
    rpm -U lambda-insights-extension.rpm && \
    rm -f lambda-insights-extension.rpm
```

Depois de criar sua função Lambda, atribua a política do IAM CloudWatchLambdaInsightsExecutionRolePolicy à função de execução da função, e o Lambda Insights estará habilitado na função Lambda baseada na imagem de contêiner.

> **Note**
>
> Para usar uma versão mais antiga da extensão do Lambda Insights, substitua a URL nos comandos anteriores por esta URL: `https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension.1.0.111.0.rpm`. Atualmente, apenas as versões 1.0.111.0 do Lambda Insights e posteriores estão disponíveis. Para ter mais informações, consulte Versões disponíveis da extensão do Lambda Insights.

Para verificar a assinatura do pacote do agente do Lambda Insights em um servidor Linux

1. Insira o comando a serguir para baixar a chave pública.

   ```
   shell$ wget https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
   lambda-insights-extension.gpg
   ```

2. Insira o comando a seguir para importar a chave pública para o token de autenticaçao.

   ```
   shell$ gpg --import lambda-insights-extension.gpg
   ```

   A saída será semelhante ao seguinte: Anote o valor da `key`, pois ele será necessário na próxima etapa. Neste exemplo de saída, a chave-valor é `848ABDC8`.

   ```
   gpg: key 848ABDC8: public key "Amazon Lambda Insights Extension" imported
   gpg: Total number processed: 1
   gpg: imported: 1  (RSA: 1)
   ```

3. Verifique a impressão digital inserindo o comando a seguir. Substitua `key-value` pelo valor da chave da etapa anterior.

   ```
   shell$  gpg --fingerprint key-value
   ```

A string de impressão digital na saída deste comando deve ser `E0AF FA11 FFF3 5BD7 349E E222 479C 97A1 848A BDC8`. Se a string não coincidir, não instale o agente e entre em contato com a AWS.

4. Depois de verificar a impressão digital, você pode usá-la para verificar a assinatura do pacote do agente do Lambda Insights. Baixe o arquivo de assinatura do pacote inserindo o comando a seguir.

```
shell$  wget https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm.sig
```

5. Verifique a assinatura inserindo o comando a seguir.

```
shell$ gpg --verify lambda-insights-extension.rpm.sig lambda-insights-extension.rpm
```

A saída deve ser a seguinte:

```
gpg: Signature made Thu 08 Apr 2021 06:41:00 PM UTC using RSA key ID 848ABDC8
gpg: Good signature from "Amazon Lambda Insights Extension"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E0AF FA11 FFF3 5BD7 349E  E222 479C 97A1 848A BDC8
```

Na saída esperada, poderá haver um aviso sobre uma assinatura confiável. Uma chave somente será confiável se você ou alguém em quem você confia a tiver assinado. Isso não significa que a assinatura é inválida, apenas que você não verificou a chave pública.

Caso a saída contenha `BAD signature`, confira se você executou as etapas corretamente. Se continuar recebendo a resposta `BAD signature`, entre em contato com a AWS e evite usar o arquivo baixado.

Exemplo de x86-64

Esta seção contém um exemplo de habilitação do Lambda Insights em uma função do Python Lambda baseada em imagem de contêiner.

Um exemplo de habilitação do Lambda Insights em uma implantação de imagem de contêiner do Lambda

1. Crie um Dockerfile semelhante ao seguinte:

```
FROM public.ecr.aws/lambda/python:3.8

// extra lines to install the agent here
RUN curl -O https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm && \
    rpm -U lambda-insights-extension.rpm && \
    rm -f lambda-insights-extension.rpm

COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Crie um arquivo Python chamado `index.py` que é semelhante ao seguinte:

```
def handler(event, context):
  return {
    'message': 'Hello World!'
  }
```

3. Coloque o Dockerfile e o `index.py` no mesmo diretório. Em seguida, nesse diretório, execute as seguintes etapas para criar a imagem do docker e carregá-la no Amazon ECR.

```
// create an ECR repository
aws ecr create-repository --repository-name test-repository
// build the docker image
docker build -t test-image .
// sign in to AWS
aws ecr get-login-password | docker login --username AWS --password-stdin
 "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com
// tag the image
docker tag test-image:latest "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/
test-repository:latest
// push the image to ECR
docker push "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/test-
repository:latest
```

4. Use a imagem do Amazon ECR que você acabou de criar para criar a função Lambda.

5. Adicione a política do IAM CloudWatchLambdaInsightsExecutionRolePolicy à função de execução da função.

Implantação de imagem de contêiner ARM64

Para habilitar o Lambda Insights em uma função do Lambda implantada como uma imagem de contêiner em execução em um contêiner AL2-aarch64 (que usa a arquitetura ARM64), adicione as linhas a seguir ao Dockerfile. Essas linhas instalam o agente do Lambda Insights como uma extensão em sua imagem de contêiner.

```
RUN curl -O https://lambda-insights-extension-arm64.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension-arm64.rpm && \
    rpm -U lambda-insights-extension-arm64.rpm && \
    rm -f lambda-insights-extension-arm64.rpm
```

Depois de criar sua função Lambda, atribua a política do IAM CloudWatchLambdaInsightsExecutionRolePolicy à função de execução da função, e o Lambda Insights estará habilitado na função Lambda baseada na imagem de contêiner.

> **Note**
>
> Para usar uma versão mais antiga da extensão do Lambda Insights, substitua a URL nos comandos anteriores por esta URL: `https://lambda-insights-extension-arm64.s3-ap-northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.1.0.229.0.rpm`. Atualmente, apenas as versões 1.0.229.0 do Lambda Insights e posteriores estão disponíveis. Para ter mais informações, consulte Versões disponíveis da extensão do Lambda Insights.

Para verificar a assinatura do pacote do agente do Lambda Insights em um servidor Linux

1. Insira o comando a serguir para baixar a chave pública.

```
shell$ wget https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/lambda-insights-extension.gpg
```

2. Insira o comando a seguir para importar a chave pública para o token de autenticaçao.

```
shell$ gpg --import lambda-insights-extension.gpg
```

A saída será semelhante ao seguinte: Anote o valor da `key`, pois ele será necessário na próxima etapa. Neste exemplo de saída, a chave-valor é `848ABDC8`.

```
gpg: key 848ABDC8: public key "Amazon Lambda Insights Extension" imported
gpg: Total number processed: 1
gpg: imported: 1  (RSA: 1)
```

3. Verifique a impressão digital inserindo o comando a seguir. Substitua `key-value` pelo valor da chave da etapa anterior.

```
shell$  gpg --fingerprint key-value
```

A string de impressão digital na saída deste comando deve ser `E0AF FA11 FFF3 5BD7 349E E222 479C 97A1 848A BDC8`. Se a string não coincidir, não instale o agente e entre em contato com a AWS.

4. Depois de verificar a impressão digital, você pode usá-la para verificar a assinatura do pacote do agente do Lambda Insights. Baixe o arquivo de assinatura do pacote inserindo o comando a seguir.

```
shell$  wget https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.rpm.sig
```

5. Verifique a assinatura inserindo o comando a seguir.

```
shell$ gpg --verify lambda-insights-extension-arm64.rpm.sig lambda-insights-
extension-arm64.rpm
```

A saída deve ser a seguinte:

```
gpg: Signature made Thu 08 Apr 2021 06:41:00 PM UTC using RSA key ID 848ABDC8
gpg: Good signature from "Amazon Lambda Insights Extension"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E0AF FA11 FFF3 5BD7 349E  E222 479C 97A1 848A BDC8
```

Na saída esperada, poderá haver um aviso sobre uma assinatura confiável. Uma chave somente será confiável se você ou alguém em quem você confia a tiver assinado. Isso não significa que a assinatura é inválida, apenas que você não verificou a chave pública.

Caso a saída contenha `BAD signature`, confira se você executou as etapas corretamente. Se continuar recebendo a resposta `BAD signature`, entre em contato com a AWS e evite usar o arquivo baixado.

Exemplo de ARM64

Esta seção contém um exemplo de habilitação do Lambda Insights em uma função do Python Lambda baseada em imagem de contêiner.

Um exemplo de habilitação do Lambda Insights em uma implantação de imagem de contêiner do Lambda

1. Crie um Dockerfile semelhante ao seguinte:

```
FROM public.ecr.aws/lambda/python:3.8
// extra lines to install the agent here
RUN curl -O https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.rpm && \
    rpm -U lambda-insights-extension-arm64.rpm && \
    rm -f lambda-insights-extension-arm64.rpm

COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Crie um arquivo Python chamado `index.py` que é semelhante ao seguinte:

```
def handler(event, context):
  return {
    'message': 'Hello World!'
  }
```

3. Coloque o Dockerfile e o `index.py` no mesmo diretório. Em seguida, nesse diretório, execute as seguintes etapas para criar a imagem do docker e carregá-la no Amazon ECR.

```
// create an ECR repository
aws ecr create-repository --repository-name test-repository
// build the docker image
docker build -t test-image .
// sign in to AWS
aws ecr get-login-password | docker login --username AWS --password-stdin
 "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com
```

```
// tag the image
docker tag test-image:latest "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/
test-repository:latest
// push the image to ECR
docker push "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/test-
repository:latest
```

4. Use a imagem do Amazon ECR que você acabou de criar para criar a função Lambda.
5. Adicione a política do IAM CloudWatchLambdaInsightsExecutionRolePolicy à função de execução da função.

Visualizar métricas do Lambda Insights

Depois de instalar a extensão Lambda Insights em uma função Lambda que foi invocada, você pode usar o console do CloudWatch para visualizar as métricas. É possível ter uma visão geral de várias funções ou focar em uma única função.

Para obter uma lista das métricas do Lambda Insights, consulte Métricas coletadas pelo Lambda Insights.

Para exibir a visão geral de várias funções das métricas do Lambda Insights

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, em Lambda Insights, escolha Multi-function (Várias funções).

   A parte superior da página exibe grafos com métricas agregadas de todas as suas funções do Lambda na região que têm o Lambda Insights habilitado. Na parte inferior da página há uma tabela que lista as funções.
3. Para filtrar por nome de função para reduzir o número de funções exibidas, digite parte do nome da função na caixa próxima à parte superior da página.
4. Para adicionar este gráfico a um painel como widget, escolha Add to dashboard (Adicionar ao painel).

Para visualizar as métricas de uma única função

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

2. No painel de navegação à esquerda, em Lambda Insights, escolha Single-function (Função única).

   A parte superior da página exibe grafos com métricas para a função selecionada.

3. Se o X-Ray estiver habilitado, você poderá escolher um único ID de rastreamento. Isso abre a página do mapa de rastreamento do X-Ray para essa invocação e, daí em diante, é possível reduzir o zoom para ver o rastreamento distribuído e os outros serviços envolvidos no processamento dessa transação específica. Para obter mais informações, sobre o mapa de rastreamento do X-Ray, consulte Using the X-Ray Trace Map.

4. Para abrir o CloudWatch Logs Insights e ampliar um erro específico, escolha View logs (Visualizar logs) pela tabela na parte inferior da página.

5. Para adicionar este gráfico a um painel como widget, escolha Add to dashboard (Adicionar ao painel).

Integração ao Application Insights

O Amazon CloudWatch Application Insights ajuda a monitorar suas aplicações, além de identificar e configurar as principais métricas, logs e alarmes nos recursos da aplicação e pilha de tecnologia. Para ter mais informações, consulte Amazon CloudWatch Application Insights.

Você pode habilitar o Application Insights para coletar dados adicionais de suas funções do Lambda. Se ainda não tiver feito isso, é possível habilitá-lo escolhendo Auto-configure Application Insights (Configurar automaticamente o Application Insights) na guia Application Insights abaixo da visualização de performance no painel Lambda Insights.

Se já tiver configurado o CloudWatch Application Insights para monitorar suas funções o Lambda, o painel do Application Insights aparecerá abaixo do painel do Lambda Insights, na guia Application Insights.

Métricas coletadas pelo Lambda Insights

O Lambda Insights coleta várias métricas das funções do Lambda onde está instalado. Algumas dessas métricas estão disponíveis como dados agregados de séries temporais no CloudWatch Metrics. Outras métricas não são agregadas em dados de séries temporais, mas podem ser encontradas nas entradas de log de formato de métrica incorporado com o CloudWatch Logs Insights.

As métricas a seguir estão disponíveis como dados agregados de séries temporais no CloudWatch Metrics no namespace `LambdaInsights`.

Nome da métrica	Dimensões	Descrição
`cpu_total_time`	function_name function_name, version	Soma de `cpu_system_time` e `cpu_user_time`. Unidade: milissegundos
`init_duration`	function_name function_name, version	A quantidade de tempo gasta na fase `init` do ciclo de vida do ambiente de execução do Lambda. Unidade: milissegundos
`memory_utilization`	function_name function_name, version	A memória máxima medida como uma porcentagem da memória alocada para a função. Unidade: Percentual
`rx_bytes`	function_name function_name, version	Número de bytes recebidos pela função. Unidade: bytes
`tmp_used`		A quantidade de espaço usada no diretório `/tmp`.

Nome da métrica	Dimensões	Descrição
		Unidade: bytes
`tx_bytes`	function_name function_name, version	O número de bytes enviados pela função. Unidade: bytes
`total_memory`	function_name function_name, version	A quantidade de memória alocada em sua função Lambda. É o mesmo que o tamanho da memória de sua função. Unidade: megabytes
`total_network`	function_name function_name, version	Soma de `rx_bytes` e `tx_bytes`. Mesmo para funções que não executam tarefas de E/S, esse valor geralmente é maior que zero por causa das chamadas de rede realizadas pelo runtime do Lambda. Unidade: bytes
`used_memory_max`	function_name function_name, version	A memória medida da sandbox da função. Unidade: megabytes

As métricas a seguir podem ser encontradas nas entradas de log de formato de métrica incorporado com o CloudWatch Logs Insights. Para obter mais informações sobre o CloudWatch Logs Insights, consulte Analisar dados de log com o CloudWatch Logs Insights.

Para obter mais informações sobre o formato de métrica incorporado, consulte Incorporação de métricas em logs.

Nome da métrica	Descrição
`cpu_system_time`	A quantidade de tempo que a CPU gastou para executar o código do kernel. Unidade: milissegundos
`cpu_total_time`	Soma de `cpu_system_time` e `cpu_user_time`. Unidade: milissegundos
`cpu_user_time`	A quantidade de tempo que a CPU gastou para executar o código do usuário. Unidade: milissegundos
`fd_max`	O número máximo de descritores de arquivos disponíveis. Unidade: Contagem
`fd_use`	O número máximo de descritores de arquivos em uso. Unidade: Contagem
`memory_utilization`	A memória máxima medida como uma porcentagem da memória alocada para a função. Unidade: Percentual
`rx_bytes`	Número de bytes recebidos pela função. Unidade: bytes
`tx_bytes`	O número de bytes enviados pela função.

Nome da métrica	Descrição
	Unidade: bytes
`threads_max`	O número de threads no processo da função. Como autor da função, você não controla o número inicial de threads criados pelo runtime. Unidade: Contagem
`tmp_max`	A quantidade de espaço disponível no diretório `/tmp`. Unidade: bytes
`total_memory`	A quantidade de memória alocada em sua função Lambda. É o mesmo que o tamanho da memória de sua função. Unidade: megabytes
`total_network`	Soma de `rx_bytes` e `tx_bytes`. Mesmo para funções que não executam tarefas de E/S, esse valor geralmente é maior que zero por causa das chamadas de rede realizadas pelo runtime do Lambda. Unidade: bytes
`used_memory_max`	A memória medida da sandbox da função. Unidade: bytes

Solução de problemas e problemas conhecidos

A primeira etapa para solucionar problemas é habilitar o log de depuração na extensão do Lambda Insights. Para isso, defina esta variável de ambiente em sua função Lambda: `LAMBDA_INSIGHTS_LOG_LEVEL=info`. Para obter mais informações, consulte Usar variáveis de ambiente do AWS Lambda.

A extensão emite logs no mesmo grupo de logs que sua função (`/aws/lambda/function-name`)). Revise esses logs para ver se o erro pode estar relacionado a um problema de instalação.

Não visualizo nenhuma métrica do Lambda Insights

Caso você não veja as métricas do Lambda Insights que espera visualizar, verifique as possibilidades a seguir:

- As métricas podem estar apenas atrasadas: se a função ainda não tiver sido invocada ou se os dados ainda não tiverem sido liberados, você não verá as métricas no CloudWatch. Para obter mais informações, consulte Probelmas conhecidos mais à frente, nesta seção.
- Confirme se a função Lambda tem as permissões corretas: verifique se a política do IAM CloudWatchLambdaInsightsExecutionRolePolicy está atribuída à função de execução do IAM.
- Verifique o runtime do Lambda: o Lambda Insights é compatível apenas com determinados runtimes do Lambda. Para ver uma lista dos tempos de execução compatíveis, consulte Lambda Insights.

  Por exemplo, para usar o Lambda Insights em Java 8, é necessário usar o runtime `java8.al2`, e não o runtime `java8`.
- Verificar o acesso à rede: a função Lambda pode estar em uma sub-rede privada da VPC sem acesso à Internet e você não tem um endpoint da VPC configurado para o CloudWatch Logs. Para ajudar a depurar esse problema, você pode definir a variável de ambiente `LAMBDA_INSIGHTS_LOG_LEVEL=info`.

Problemas conhecidos

O atraso de dados pode ser de até 20 minutos. Quando um manipulador de função é concluído, o Lambda congela a sandbox, que também congela a extensão do Lambda Insights. Enquanto a função está em execução, usamos uma estratégia de lotes adaptável baseada TPS da função para dados de saída. Porém, se a função parar de ser invocada por um período prolongado e ainda houver dados de evento no buffer, esses dados poderão ser atrasados até o Lambda desligar a sandbox ociosa. Quando o Lambda desliga a sandbox, liberamos os dados armazenados em buffer.

Exemplo de evento de telemetria

Cada invocação de uma função Lambda que tem o Lambda Insights habilitado grava um único evento de log no grupo de logs `/aws/lambda-insights`. Todo evento de log contém métricas

no formato de métrica incorporado. Para obter mais informações sobre o formato de métrica incorporado, consulte Incorporação de métricas em logs.

É possível usar os métodos a seguir para analisar esses eventos de log:

- A seção do Lambda Insights do console do CloudWatch, conforme explicado em Visualizar métricas do Lambda Insights.
- Registre consultas de eventos com o CloudWatch Logs Insights. Para obter mais informações, consulte Analisar dados de log com o CloudWatch Logs Insights.
- Métricas coletadas no namespace `LambdaInsights`, que você representa em grafos usando métricas do CloudWatch.

Veja a seguir um exemplo de evento de log do Lambda Insights com formato de métrica incorporado.

```
{
    "_aws": {
        "Timestamp": 1605034324256,
        "CloudWatchMetrics": [
            {
                "Namespace": "LambdaInsights",
                "Dimensions": [
                    [ "function_name" ],
                    [ "function_name", "version" ]
                ],
                "Metrics": [
                    { "Name": "memory_utilization", "Unit": "Percent" },
                    { "Name": "total_memory", "Unit": "Megabytes" },
                    { "Name": "used_memory_max", "Unit": "Megabytes" },
                    { "Name": "cpu_total_time", "Unit": "Milliseconds" },
                    { "Name": "tx_bytes", "Unit": "Bytes" },
                    { "Name": "rx_bytes", "Unit": "Bytes" },
                    { "Name": "total_network", "Unit": "Bytes" },
                    { "Name": "init_duration", "Unit": "Milliseconds" }
                ]
            }
        ],
        "LambdaInsights": {
            "ShareTelemetry": true
        }
    },
    "event_type": "performance",
```

```
    "function_name": "cpu-intensive",
    "version": "Blue",
    "request_id": "12345678-8bcc-42f7-b1de-123456789012",
    "trace_id": "1-5faae118-12345678901234567890",
    "duration": 45191,
    "billed_duration": 45200,
    "billed_mb_ms": 11571200,
    "cold_start": true,
    "init_duration": 130,
    "tmp_free": 538329088,
    "tmp_max": 551346176,
    "threads_max": 11,
    "used_memory_max": 63,
    "total_memory": 256,
    "memory_utilization": 24,
    "cpu_user_time": 6640,
    "cpu_system_time": 50,
    "cpu_total_time": 6690,
    "fd_use": 416,
    "fd_max": 32642,
    "tx_bytes": 4434,
    "rx_bytes": 6911,
    "timeout": true,
    "shutdown_reason": "Timeout",
    "total_network": 11345,
    "agent_version": "1.0.72.0",
    "agent_memory_avg": 10,
    "agent_memory_max": 10
}
```

Usar o Contributor Insights para analisar dados de alta cardinalidade

Você pode usar o Contributor Insights para analisar os dados de logs e criar séries temporais que exibem dados de colaboradores. É possível ver métricas sobre os principais colaboradores, o número total de colaboradores exclusivos e o uso deles. Isso ajuda a entender quem ou o que está afetando a performance do sistema e a localizar os principais talkers. Por exemplo, é possível encontrar hosts incorretos, identificar os usuários de rede mais pesados ou localizar os URLs que geraram mais erros.

Você pode criar suas regras do zero e, ao usar o AWS Management Console, também pode utilizar as regras de exemplo criadas pela AWS. As regras definem os campos de logs que você deseja usar para definir os colaboradores, como o `IpAddress`. Também é possível filtrar os dados de log para localizar e analisar o comportamento de colaboradores individuais.

O CloudWatch também fornece regras integradas que podem ser usadas para analisar métricas de outros produtos da AWS.

Todas as regras analisam dados de entrada em tempo real.

Se você fez login em uma conta configurada como uma conta de monitoramento na observabilidade entre contas do CloudWatch, poderá criar regras do Contributor Insights na conta de monitoramento que analisem os grupos de logs nas contas de origem e na conta de monitoramento. Você também pode criar uma única regra que analise os grupos de logs em várias contas. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

> **Note**
>
> Se você usar o Contributor Insights, você será cobrado por cada ocorrência de um evento de log que corresponda a uma regra. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

Tópicos

Criar uma regra do Contributor Insights

É possível criar regras para analisar os dados de log. Qualquer log em Common Log Format (CLF) ou JSON pode ser avaliado. Isso inclui os logs personalizados que seguem um desses formados e logs de produtos da AWS, como logs de fluxo da Amazon VPC, logs de consulta de DNS do Amazon

Route 53, logs de contêiner do Amazon ECS e logs do AWS CloudTrail, do Amazon SageMaker, do Amazon RDS, do AWS AppSync e do API Gateway.

Em uma regra, ao especificar valores ou nomes de campo, todas as correspondências diferenciam letras maiúsculas de minúsculas.

Você pode usar regras de exemplo internas ao criar uma regra ou criar sua própria regra do zero. O Contributor Insights inclui regras de exemplo para os seguintes tipos de logs:

- Logs do Amazon API Gateway
- Logs de consulta de DNS pública do Amazon Route 53
- Logs de consulta do Amazon Route 53 Resolver
- Logs do CloudWatch Container Insights
- VPC Flow Logs

Se você fez login em uma conta configurada como uma conta de monitoramento na observabilidade entre contas do CloudWatch, poderá criar regras do Contributor Insights para grupos de logs nas contas de origem vinculadas a essa conta de monitoramento, além de criar regras para grupos de logs na conta de monitoramento. Você também poderá configurar uma única regra que monitore grupos de logs em outras contas. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

> **Important**
>
> Quando você concede a permissão `cloudwatch:PutInsightRule` ao usuário, por padrão, ele pode criar uma regra que avalia qualquer grupo de logs no CloudWatch Logs. É possível adicionar condições de política do IAM que limitem essas permissões para que um usuário inclua e exclua grupos de logs específicos. Para ter mais informações, consulte Usar chaves de condição para limitar o acesso dos usuários do Contributor Insights aos grupos de log.

Como criar uma regra usando uma regra de exemplo interna

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights, Contributor Insights.
3. Escolha Create rule.

4. Em Select log group(s) (Selecionar grupos de logs), selecione os grupos de logs que você deseja que a regra monitore. Você pode selecionar até 20 grupos de logs. Se você fez login em uma conta de monitoramento configurada para a observabilidade entre contas do CloudWatch, poderá selecionar grupos de logs nas contas de origem e também configurar uma única regra para analisar grupos de logs em outras contas.

   - (Opcional) Para selecionar todos os grupos de logs que têm nomes que começam com uma string específica, escolha o menu suspenso Select by prefix match (Selecionar por correspondência de prefixo) e insira o prefixo. Se essa for uma conta de monitoramento, você poderá, opcionalmente, selecionar as contas a serem pesquisadas, do contrário, todas as contas serão selecionadas.

   > **Note**
   >
   > Você receberá cobranças por cada evento de log que corresponda à sua regra. Se escolher o menu suspenso Select by prefix match (Selecionar por correspondência com prefixo), esteja ciente da quantidade de grupos de logs com a qual o prefixo pode ter correspondência. Se você pesquisar por mais grupos de logs do que deseja, isso poderá gerar cobranças inesperadas. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

5. Em Rule type (Tipo de regra), escolha Sample rule (Exemplo de regra). Depois, escolha Select sample rule (Selecionar exemplo de regra) e selecione a regra.
6. No exemplo de regra, os campos Log format (Formato do log), Contribution (Contribuição), Filters (Filtros) e Aggregate on (Agregar em) estão preenchidos. Ajuste os valores se preferir.
7. Escolha Próximo.
8. Em Rule name (Nome da regra), insira um nome. Os caracteres válidos são a-z, A-Z, 0-9, (hífen), (sublinhado) e (ponto).
9. Escolha se deseja criar a regra em um estado ativado ou desativado. Se optar por habilitá-la, a regra começará a analisar os dados imediatamente. Você incorrerá em custos ao executar regras habilitadas. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

O Contributor Insights analisa somente novos eventos de log depois que uma regra é criada. Uma regra não pode processar eventos de log que foram processados anteriormente pelo CloudWatch Logs.

10. (Opcional) Em Tags (Etiquetas), adicione um ou mais pares chave-valor como tags para essa regra. As tags podem ajudar a identificar e organizar seus recursos da AWS e acompanhar seus custos da AWS. Para ter mais informações, consulte Etiquetar recursos do Amazon CloudWatch.
11. Escolha Criar.

Como criar uma regra do zero

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Contributor Insights.
3. Escolha Create rule.
4. Em Select log group(s) (Selecionar grupos de logs), selecione os grupos de logs que você deseja que a regra monitore. Você pode selecionar até 20 grupos de logs. Se você fez login em uma conta de monitoramento configurada para a observabilidade entre contas do CloudWatch, poderá selecionar grupos de logs nas contas de origem e também configurar uma única regra para analisar grupos de logs em outras contas.

    - (Opcional) Para selecionar todos os grupos de logs que têm nomes que começam com uma string específica, escolha o menu suspenso Select by prefix match (Selecionar por correspondência de prefixo) e insira o prefixo.

    > **Note**
    >
    > Você receberá cobranças por cada evento de log que corresponda à sua regra. Se escolher o menu suspenso Select by prefix match (Selecionar por correspondência com prefixo), esteja ciente da quantidade de grupos de logs com a qual o prefixo pode ter correspondência. Se você pesquisar por mais grupos de logs do que deseja, isso poderá gerar cobranças inesperadas. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

5. Em Rule type (Tipo de regra), escolha Custom rule (Regra personalizada).

6. Em Log format (Formato de log), selecione JSON ou CLF.
7. É possível terminar de criar a regra usando o assistente ou selecionando a guia Syntax (Sintaxe) e especificando a sintaxe da regra manualmente.

   Para continuar usando o assistente, faça o seguinte:

   a. Em Contribution (Contribuição), Key (Chave), insira um tipo de colaborador sobre o qual deseja relatar. O relatório exibe os principais valores desse tipo de colaborador.

      As entradas válidas são qualquer campo de log que tenha valores. Os exemplos incluem **`requestId`**, **`sourceIPaddress`** e **`containerID`**.

      Para obter informações sobre como encontrar os nomes de campos de log dos logs de determinado grupo de logs, consulte Localização de campos de log.

      Chaves maiores de 1 KB são truncadas para 1 1KB.

   b. (Opcional) Escolha Add new key (Adicionar nova chave) para adicionar mais chaves. É possível incluir até quatro chaves em uma regra. Se você inserir mais de uma chave, os colaboradores no relatório serão definidos por combinações de valor exclusivas das chaves. Por exemplo, se você especificar três chaves, cada combinação exclusiva de valores para as três chaves será contada como um colaborador exclusivo.

   c. (Opcional) Se você deseja adicionar um filtro que restringe o escopo dos resultados, escolha Add filter (Adicionar filtro). Em Match (Correspondência), insira o nome do campo de log pelo qual deseja filtrar. Em Condition (Condição), escolha o operador de comparação e insira um valor pelo qual deseja filtrar esse campo.

      É possível adicionar até quatro filtros em uma regra. Diversos filtros são unidos pela lógica AND (E), portanto, somente eventos de log que correspondem a todos os filtros são avaliados.

      > **Note**
      >
      > Arrays que seguem operadores de comparação, como `In`, `NotIn` ou `StartsWith`, pode incluir até dez valores de string. Para obter mais informações sobre a sintaxe de regras do Contributor Insights, consulte Sintaxe de regras do Contributor Insights.

   d. Em Aggregate on (Agregar em), selecione Count (Contagem) ou Sum (Soma). Escolher Count (Contagem) faz com que a classificação do colaborador seja baseada no número de ocorrências. Escolher Sum (Soma) faz com que a classificação seja baseada na soma agregada dos valores do campo especificado em Contribution (Contribuição), Value (Valor).

8. Para inserir a regra como um objeto JSON em vez de usar o assistente, faça o seguinte:

   a. Selecione a guia Syntax (Sintaxe).

   b. Em Rule body (Corpo da regra), insira o objeto JSON da regra. Para obter informações sobre a sintaxe da regra, consulte Sintaxe de regras do Contributor Insights.

9. Escolha Próximo.

10. Em Rule name (Nome da regra), insira um nome. Os caracteres válidos são A-Z, a-z, 0-9, "-", "_' e ".".

11. Escolha se deseja criar a regra em um estado ativado ou desativado. Se optar por habilitá-la, a regra começará a analisar os dados imediatamente. Você incorrerá em custos ao executar regras habilitadas. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

    O Contributor Insights analisa somente novos eventos de log depois que uma regra é criada. Uma regra não pode processar eventos de log que foram processados anteriormente pelo CloudWatch Logs.

12. (Opcional) Em Tags (Etiquetas), adicione um ou mais pares chave-valor como tags para essa regra. As tags podem ajudar a identificar e organizar seus recursos da AWS e acompanhar seus custos da AWS. Para ter mais informações, consulte Etiquetar recursos do Amazon CloudWatch.

13. Escolha Próximo.

14. Confirme as configurações que você inseriu e escolha Create rule (Criar regra).

É possível desativar, ativar ou excluir regras que você criou.

Como ativar, desativar ou excluir uma regra no Contributor Insights

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.

2. No painel de navegação, selecione Contributor Insights.

3. Na lista de regras, marque a caixa de seleção ao lado de uma única regra.

As regras integradas são criadas pelos serviços da AWS e não podem ser editadas, desativadas ou excluídas.

4. Selecione Actions (Ações) e escolha a opção desejada.

Localizar campos de log

Ao criar uma regra, é necessário saber os nomes dos campos das entradas de log em um grupo de logs.

Como localizar os campos de log em um grupo de logs

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, em Logs, escolha Insights.
3. Acima do editor de consultas, escolha um ou mais grupos de log para serem consultados.

   Quando você seleciona um grupo de logs, o CloudWatch Logs Insights automaticamente detecta campos nos dados no grupo de logs e os exibe no painel à direta, em Discovered fields (Campos detectados).

Sintaxe de regras do Contributor Insights

Esta seção explica a sintaxe das regras do Contributor Insights Use essa sintaxe somente quando estiver criando uma regra inserindo um bloco JSON. Se você usar o assistente para criar uma regra, não será necessário conhecer a sintaxe. Para obter mais informações sobre como criar regras usando o assistente, consulte Criar uma regra do Contributor Insights.

Todas as correspondências de regras a valores e nomes de campos de eventos de logs diferenciam letras maiúsculas de minúsculas.

O exemplo a seguir ilustra a sintaxe de logs JSON.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
```

```
        "API-Gateway-Access-Logs*",
        "Log-group-name2"
    ],
    "LogFormat": "JSON",
    "Contribution": {
        "Keys": [
            "$.ip"
        ],
        "ValueOf": "$.requestBytes",
        "Filters": [
            {
                "Match": "$.httpMethod",
                "In": [
                    "PUT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Campos nas regras do Contributor Insights

Schema

O valor de `Schema` para uma regra que analisa dados do CloudWatch Logs deverá ser sempre `{"Name": "CloudWatchLogRule", "Version": 1}`

LogGroupNames

Uma matriz de strings. Para cada elemento na matriz, é possível usar * no final de uma string para incluir todos os grupos de log com nomes que começam com esse prefixo.

Tenha cuidado com o uso de curingas em nomes de grupos de logs. Serão feitas cobranças por cada evento de log que corresponda a uma regra. Se você pesquisar acidentalmente mais grupos de logs do que pretendia, isso poderá gerar cobranças inesperadas. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

LogGroupARNs

Se você estiver criando essa regra em uma conta de monitoramento da observabilidade entre contas do CloudWatch, você pode usar `LogGroupARNs` para especificar grupos de logs nas contas de origem vinculadas à conta de monitoramento e para especificar grupos de logs na

própria conta de monitoramento. Você deve especificar `LogGroupNames` ou `LogGroupARNs` na regra, mas não ambos.

`LogGroupARNs` é uma matriz de strings. Para cada elemento da matriz, você tem a opção de usar * como curinga em determinadas situações. Por exemplo, você pode definir `arn:aws:logs:us-west-1:*:log-group/MyLogGroupName2` para especificar grupos de logs denominados `MyLogGroupName2` em todas as contas de origem e na conta de monitoramento, na região Oeste dos EUA (Norte da Califórnia). Você também pode definir `arn:aws:logs:us-west-1:111122223333:log-group/GroupNamePrefix*` para especificar todos os grupos de logs na região Oeste dos EUA (Norte da Califórnia) em 111122223333 que tenham nomes começando com `GroupNamePrefix`.

Você não pode especificar um ID de conta da AWS parcial como prefixo com um curinga.

Tenha cuidado com o uso de curingas com ARNs s de grupos de logs. Serão feitas cobranças por cada evento de log que corresponda a uma regra. Se você pesquisar acidentalmente mais grupos de logs do que pretendia, isso poderá gerar cobranças inesperadas. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

LogFormat

Os valores válidos são `JSON` e `CLF`.

Contribuição

Esse objeto inclui uma matriz `Keys` com até quatro membros, opcionalmente, um único `ValueOf` e, opcionalmente, uma matriz de até quatro `Filters`.

Chaves

Uma matriz de até quatro campos de log que são usados como dimensões para classificar colaboradores. Se você inserir mais de uma chave, cada combinação exclusiva de valores para as chaves será contada como um colaborador exclusivo. Os campos devem ser especificados usando a notação no formato de propriedade JSON.

ValueOf

(Opcional) Especifique isso somente quando estiver especificando `Sum` como o valor de `AggregateOn`. O `ValueOf` especifica um campo de log com valores numéricos. Nesse tipo de regra, os colaboradores são classificados pela soma do valor desse campo, em vez do número de ocorrências nas entradas do log. Por exemplo, se você quiser classificar os colaboradores pelo `BytesSent` total durante um período, defina `ValueOf` como `BytesSent` e especifique `Sum` em `AggregateOn`.

Filtros

(Opcional) Especifica uma matriz de até quatro filtros para restringir os eventos de log que serão incluídos no relatório. Se você especificar vários filtros, eles serão avaliados pelo Contributor Insights com um operador lógico AND (E). Você pode usar isso para excluir eventos de log irrelevantes na pesquisa ou escolher um único colaborador para ter o comportamento analisado.

Cada membro da matriz deve incluir um campo `Match` e um campo indicando o tipo de operador de correspondência que deve ser usado.

O campo `Match` especifica um campo de log para ser avaliado no filtro. O campo de log é especificado usando a notação de formato de propriedade JSON.

O campo de operador de correspondência deve ser um dos seguintes: `In`, `NotIn`, `StartsWith`, `GreaterThan`, `LessThan`, `EqualTo`, `NotEqualTo` ou `IsPresent`. Se o campo de operador for `In`, `NotIn` ou `StartsWith`, ele será seguido por uma matriz de valores de string a serem verificados. O Contributor Insights avalia a matriz de valores de string com um operador OR (OU). A matriz pode incluir até 10 valores de string.

Se o campo de operador for `GreaterThan`, `LessThan`, `EqualTo` ou `NotEqualTo`, ele será seguido por um único valor número com o qual será comparado.

Se o campo de operador for `IsPresent`, ele será seguido por `true` ou `false`. Esse operador corresponde a eventos de log dependendo se o campo de log especificado estava presente ou não no evento de log. O `isPresent` funciona somente com valores no nó folha de propriedades JSON. Por exemplo, um filtro que procura correspondências com `c-count` não avaliará um evento de log com um valor de `details.c-count.c1`.

Consulte o seguinte para exemplos de filtro:

```
{"Match": "$.httpMethod", "In": [ "PUT", ] }
{"Match": "$.StatusCode", "EqualTo": 200 }
{"Match": "$.BytesReceived", "GreaterThan": 10000}
{"Match": "$.eventSource", "StartsWith": [ "ec2", "ecs" ] }
```

AggregateOn

Os valores válidos são `Count` e `Sum`. Especifica se o relatório deve ser agregado com base em uma contagem de ocorrências ou em uma soma dos valores do campo especificada no campo `ValueOf`.

Notação de formato de propriedade JSON

Os campos `Keys`, `ValueOf` e `Match` seguem o formato de propriedade JSON sem notação de ponto, onde $ representa a raiz do objeto JSON. Isto é seguido por um ponto e por uma string alfanumérica com o nome da subpropriedade. Vários níveis de propriedade são compatíveis.

O primeiro caractere da string só pode ser A-Z ou a-z. Os caracteres a seguir da string podem ser A-Z, a-z ou 0-9.

A lista a seguir ilustra exemplos válidos do formato de propriedade JSON

```
$.userAgent
$.endpoints[0]
$.users[1].name
$.requestParameters.instanceId
```

Campo adicional em regras para logs em CLF

Os eventos de log em Common Log Format (CLF) não têm nomes para os campos como JSON. Para fornecer os campos a serem usados pelas regras do Contributor Insights, um log em CLF pode ser tratado como uma matriz com um índice que começa em 1. É possível especificar o primeiro campo como **"1"**, o segundo campo como **"2"** e assim por diante.

Para tornar uma regra para um log em CLF mais fácil de ler, é possível usar `Fields`. Isso permite fornecer um alias de nomeação para os locais de campo CLF. Por exemplo, você pode especificar que o local "4" é um endereço IP. Após especificado, `IpAddress` pode ser usado como uma propriedade em `Keys`, `ValueOf` e `Filters` na regra.

Veja a seguir um exemplo de uma regra para um log em que o CLF usa o campo `Fields`.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "IpAddress",
```

```
        "7": "StatusCode"
    },
    "Contribution": {
        "Keys": [
            "IpAddress"
        ],
        "Filters": [
            {
                "Match": "StatusCode",
                "EqualTo": 200
            }
        ]
    },
    "AggregateOn": "Count"
}
```

Exemplos de regras do Contributor Insights

Esta seção contém exemplos que ilustram casos de uso para regras do Contributor Insights.

Logs de fluxo da VPC: transferências de byte por endereço IP de origem e endereço IP de destino

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "srcaddr",
        "5": "dstaddr",
        "10": "bytes"
    },
    "Contribution": {
        "Keys": [
            "srcaddr",
            "dstaddr"
        ],
        "ValueOf": "bytes",
        "Filters": []
```

```
    },
    "AggregateOn": "Sum"
}
```

Logs de fluxo da VPC: número mais alto de solicitações HTTPS

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "5": "destination address",
        "7": "destination port",
        "9": "packet count"
    },
    "Contribution": {
        "Keys": [
            "destination address"
        ],
        "ValueOf": "packet count",
        "Filters": [
            {
                "Match": "destination port",
                "EqualTo": 443
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Logs de fluxo da VPC: conexões TCP rejeitadas

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
```

```
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Respostas do Route 53 NxDomain por endereço de origem

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.rcode",
                "StartsWith": [
```

```
                    "NXDOMAIN"
                ]
            }
        ],
        "Keys": [
            "$.srcaddr"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Consultas do Route 53 Resolver por nome de domínio

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [],
        "Keys": [
            "$.query_name"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Consultas do Route 53 Resolver por tipo de consulta e endereço de origem

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
```

```
        "Filters": [],
        "Keys": [
            "$.query_type",
            "$.srcaddr"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Visualizar relatórios do Contributor Insights

Para visualizar gráficos de dados do relatório e uma lista de classificação de colaboradores encontrados pelas regras, siga estas etapas.

Como visualizar os relatórios de regras

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Contributor Insights.
3. Na lista de regras, escolha o nome de uma regra.

   O gráfico exibe os resultados da regra pelas últimas três horas. A tabela abaixo do gráfico mostra os 10 principais colaboradores.
4. Para alterar o número de colaboradores mostrados na tabela, selecione Top 10 contributors (10 principais colaboradores) na parte superior do gráfico.
5. Para filtrar o gráfico a fim de que ele exiba somente os resultados de um único colaborador, escolha o colaborador na legenda da tabela. Para exibir novamente todos os colaboradores, escolha o mesmo colaborador novamente na legenda.
6. Para alterar o intervalo de tempo mostrado no relatório, escolha 15m, 30m, 1h, 2h, 3h ou custom (personalizado) na parte superior do gráfico.

   O intervalo de tempo máximo para o relatório é de 24 horas, mas é possível escolher uma janela de 24 horas que ocorreu até 15 dias atrás. Para escolher uma janela de tempo no passado, selecione custom (personalizado), absolute (absoluto) e especifique a janela de tempo.
7. Para alterar a duração do período usado para a agregação e a classificação de colaboradores, selecione period (período) na parte superior do gráfico. Visualizar um período mais longo

geralmente mostra um relatório mais suave, com poucos picos Escolher um período mais curto tem mais possibilidade de exibir picos.

8. Para adicionar este grafo a um painel do CloudWatch, escolha Add to dashboard (Adicionar ao painel).
9. Para abrir a janela de consulta do CloudWatch Logs Insights, com o grupo de log nesse relatório já carregado na caixa de consulta, selecione View logs (Visualizar logs).
10. Para exportar os dados do relatório para a área de transferência como um arquivo CSV, selecione Export (Exportar).

Criar gráfico de métricas geradas por regras

O Contributor Insights fornece uma função matemática de métrica, `INSIGHT_RULE_METRIC`. Você pode usar essa função para adicionar dados de um relatório do Contributor Insights a um grafo na guia Metrics (Métricas) do console do CloudWatch. Você também pode definir um alarme com base nessa função matemática. Para mais informações sobre funções matemáticas de métrica, consulte Usar matemática de métricas.

Para usar essa função matemática de métrica, é necessário estar conectado em uma conta que tenha as permissões `cloudwatch:GetMetricData` e `cloudwatch:GetInsightRuleReport`.

A sintaxe é `INSIGHT_RULE_METRIC(`*`ruleName`*`, `*`metricName`*`)`. *`ruleName`* é o nome de uma regra do Contributor Insights, e *`metricName`* é um dos valores na lista a seguir. O valor de *`metricName`* determina qual tipo de dados a função matemática retorna.

- `UniqueContributors`: o número de colaboradores exclusivos para cada ponto de dados.
- `MaxContributorValue`: o valor do principal colaborador para cada ponto de dados. A identidade do colaborador pode mudar para cada ponto de dados no gráfico.

  Se essa regra for agregada por `Count`, o principal colaborador de cada ponto de dados será o colaborador com mais ocorrências nesse período. Se a regra for agregada por `Sum`, o principal colaborador será o que tiver a maior soma no campo de log especificado pelo `Value` da regra durante esse período.
- `SampleCount`: o número de pontos de dados correspondentes pela regra.
- `Sum`: a soma dos valores de todos os colaboradores durante o período representado por esse ponto de dados.

- `Minimum`: o valor mínimo de uma única observação durante o período representado por esse ponto de dados.
- `Maximum`: o valor máximo de uma única observação durante o período representado por esse ponto de dados.
- `Average`: o valor médio de todos os colaboradores durante o período representado por esse ponto de dados.

Definir um alarme para os dados de métrica do Contributor Insights

É possível definir alarmes em métricas geradas pelo Contributor Insights usando a função `INSIGHT_RULE_METRIC`. Por exemplo, você pode criar um alarme com base na porcentagem de conexões de Transmission Control Protocol (TCP – Protocolo de controle de transmissão) rejeitadas. Para começar com esse tipo de alarme, você pode criar regras como as mostradas nos dois exemplos a seguir:

Exemplo de regra: "RejectedConnectionsRule"

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
```

```
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Exemplo de regra: "TotalConnectionsRule"

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            }
    "AggregateOn": "Sum"
}
```

Depois de criar suas regras, você pode selecionar a guia Metrics (Métricas), no Console do CloudWatch, e usar as seguintes expressões matemáticas de métrica de exemplo para representar gráficos dos dados que o Contributor Insights relata:

Exemplo: expressões matemáticas de métrica

```
e1 INSIGHT_RULE_METRIC("RejectedConnectionsRule", "Sum")
e2 INSIGHT_RULE_METRIC("TotalConnectionsRule", "Sum")
e3 (e1/e2)*100
```

No exemplo, a expressão matemática de métrica `e3` retorna todas as conexões TCP rejeitadas. Se quiser ser notificado quando 20% das conexões TCP forem rejeitadas, você poderá modificar a expressão, alterando o limite de `100` para `20`.

> **Note**
>
> Você pode criar um alarme em uma métrica que você está monitorando a partir da seção Metrics (Métricas). Enquanto estiver na guia Graphed metrics (Representar métricas em gráficos), você poderá selecionar o ícone Create alarm (Criar alarme) na coluna Actions (Ações). O ícone Create alarm (Criar alarme) parece um sino.

Para obter mais informações sobre como criar gráficos de métricas e usar funções matemáticas de métricas, consulte a seguinte seção: Adicionar uma expressão matemática a um gráfico do CloudWatch.

Usar as regras integradas do Contributor Insights

Você pode usar as regras internas do Contributor Insights para analisar métricas de outros produtos da AWS. Os produtos a seguir oferecem suporte a regras internas:

- Contributor Insights para Amazon DynamoDB no Guia do desenvolvedor do Amazon DynamoDB.
- Usar as regras integradas do Contributor Insights no Guia do AWS PrivateLink.

Amazon CloudWatch Application Insights

O Amazon CloudWatch Application Insights facilita a observação para suas aplicações e recursos da AWS subjacentes. Ele ajuda a configurar os melhores monitores para os recursos da aplicação, a

analisar dados continuamente para procurar sinais de problemas com suas aplicações. O Application Insights, que é desenvolvido pelo Sagemaker e outras tecnologias da AWS, fornece painéis automatizados que exibem problemas potenciais com aplicações monitoradas, que ajudam a isolar rapidamente problemas contínuos com suas aplicações e sua infraestrutura. A visibilidade melhorada da integridade de suas aplicações que o Application Insights fornece ajuda a reduzir tempo médio de reparo (MTTR) para solucionar os problemas de sua aplicação.

Quando você adiciona suas aplicações ao Amazon CloudWatch Application Insights, ele examina os recursos das aplicações, recomenda e configura métricas e logs no CloudWatch para componentes da aplicação. Os exemplos de componentes de aplicação incluem bancos de dados de backend do SQL Server e níveis do Microsoft IIS/Web. O Application Insights analisa os padrões da métrica usando dados históricos para detectar anomalias, além de detectar continuamente erros e exceções do sistema operacional da aplicação e logs de infraestrutura. Ele correlaciona essas observações usando uma combinação de algoritmos de classificação e regras integradas. Em seguida, cria automaticamente os painéis que exibem as observações relevantes e informações sobre a gravidade do problema para ajudar você a priorizar suas ações. Para problemas comuns nas pilhas de aplicações .NET e SQL, como a latência da aplicação, backups com falha do SQL Server, vazamentos de memória, solicitações HTTP grandes e operações de E/S canceladas, ele fornece insights adicionais que indicam a possível causa raiz do problema e as etapas para a resolução. A integração incorporada ao AWS SSM OpsCenter permite resolver problemas executando o documento relevante do Systems Manager Automation.

Seções

O que é o Amazon CloudWatch Application Insights?

O CloudWatch Application Insights ajudaa monitorar as aplicações que usam instâncias do Amazon EC2 juntamente com outros recursos de aplicações. Ele identifica e configura os principais logs de métricas e alarmes na pilha de tecnologia e nos recursos da aplicação (por exemplo, banco de dados Microsoft SQL Server, servidores web (IIS) e de aplicações, SO, load balancers e filas). Ele monitora continuamente os logs e as métricas para detectar e correlacionar anomalias e erros. Quando erros e anomalias são detectados, o Application Insights gera o CloudWatch Events que é possível usar para configurar notificações ou executar ações. Para auxiliar na solução de problemas, ele cria painéis automatizados para problemas detectados, que incluem anomalias de métricas correlacionadas e erros de log com insights adicionais para indicar uma potencial causa raiz do problema. Os painéis automatizados ajudam a tomar medidas corretivas rápidas para manter a integridade de suas aplicações e evitar o impacto nos usuários finais da aplicação. Ele também cria OpsItems para que você possa resolver problemas usando o AWS SSM OpsCenter.

É possível configurar contadores importantes, como transações de gravação espelhada por segundo, comprimento da fila de recuperação e atraso da transação, bem como logs de eventos do Windows no CloudWatch. Quando ocorre um evento de failover ou problema com sua workload de alta disponibilidade do SQL, como um acesso restrito para consultar um banco de dados de destino, o CloudWatch Application Insights oferece insights automatizados.

O CloudWatch Application Insights integra-se aoAWS Launch Wizard para fornecer uma experiência de configuração de monitoramento com um clique para implantar workloads de alta disponibilidade do SQL Server na AWS. Ao selecionar a opção para configurar o monitoramento e os insights com o Application Insights no console do Launch Wizard, o CloudWatch Application Insights configura automaticamente métricas, logs e alarmes relevantes no CloudWatch e inicia o monitoramento das workloadsrecém-implantadas. É possível exibir informações automatizadas e problemas detectados, juntamente com a integridade de suas workloads de alta disponibilidade do SQL Server, no console do CloudWatch.

Conteúdo

Atributos

O Application Insights fornece os recursos a seguir.

Configuração automática de monitores para recursos de aplicações

O CloudWatch Application Insights reduz o tempo necessário para configurar o monitoramento de suas aplicações. Ele faz isso verificando os recursos da aplicação, fornecendo uma lista personalizável de métricas e logs recomendados e configurando-os no CloudWatch para fornecer a visibilidade necessária dos recursos da aplicação, como o Amazon EC2 e os balanceadores de carga elásticos (ELB). Ele também configura alarmes dinâmicos em métricas monitoradas. Os alarmes são atualizados automaticamente com base nas anomalias detectadas nas duas semanas anteriores.

Notificação e detecção de problemas

O CloudWatch Application Insights detecta sinais de possíveis problemas com sua aplicação, como anomalias de métrica e erros de log. Ele correlaciona essas observações com possíveis problemas superficiais com sua aplicação. Em seguida, ele gera CloudWatch Events, que podem ser configurados para receber notificações ou executar ações. Isso elimina a necessidade de criar alarmes em métricas individuais ou erros de log.

Solução de problemas

O CloudWatch Application Insights cria painéis automáticos do CloudWatch para os problemas que são detectados. Os painéis mostram detalhes sobre o problema, incluindo as anomalias de métricas associadas e erros de log para ajudar a solucionar problemas. Eles também fornecem informações adicionais que indicam possíveis causas das anomalias e dos erros.

Conceitos

Os seguintes conceitos são importantes para entender como o Application Insights monitora sua aplicação.

Componente

Um agrupamento personalizado, individual ou agrupado automaticamente de recursos semelhantes que compõem uma aplicação. Recomendamos agrupar recursos semelhantes em componentes personalizados para melhor monitoramento.

Observação

Um evento individual (anomalia de métrica, log de erro ou exceção) que é detectado com uma aplicação ou recurso da aplicação.

Problema

Os problemas são detectados ao correlacionar, classificar e agrupar observações relacionadas.

Para ver definições de outros conceitos principais do CloudWatch Application Insights, consulte Conceitos do Amazon CloudWatch.

Definição de preço

O CloudWatch Application Insights configura as métricas e os logs recomendados para determinados recursos da aplicação usando as métricas, os logs e os eventos do CloudWatch para enviar notificações sobre os problemas detectados. Esses recursos são cobrados em sua conta da AWS de acordo com o preço do CloudWatch. Para os problemas detectados, OpsItems do SSM também são criados pelo Application Insights para notificar você sobre problemas. Além disso, o Application Insights cria parâmetros do SSM Parameter Store para configurar os agentes do CloudWatch nas instâncias. Os recursos do Systems Manager do Amazon EC2 são alterados de acordo com os preços do SSM. Você não é cobrado por assistência na configuração, monitoramento, análise de dados ou detecção de problemas.

Custos do CloudWatch Application Insights

Os custos do Amazon EC2 incluem o uso dos seguintes recursos:

- Agente do CloudWatch
  - Grupos de logs do agente do CloudWatch
  - Métricas do agente do CloudWatch
  - Grupos de logs do Prometheus (para workloads JMX)

Os custos de todos os recursos incluem o uso dos seguintes recursos:

- Alarmes do CloudWatch (maior parte do custo)
- Opsitems do SSM (custo mínimo)

Exemplo de cálculo de custos

Os custos neste exemplo são considerados de acordo com o cenário a seguir.

Você criou um grupo de recursos que inclui o seguinte:

- Uma instância do Amazon EC2 com o SQL Server instalado.
- Um volume do Amazon EBS anexado.

Quando você integra esse grupo de recursos com o CloudWatch Application Insights, a workload do SQL Server instalada na instância do Amazon EC2 é detectada. O CloudWatch Application Insights começa a monitorar as métricas a seguir.

As métricas a seguir são monitoradas para a instância do SQL Server:

- CPUUtilization
- StatusCheckFailed
- % de bytes confirmados em uso na memória
- Mbytes de memória disponíveis
- Total/s de bytes de interface de rede
- % de uso de arquivo de paginação
- % de tempo de disco do disco físico
- % de tempo de processador do processador
- SQLServer: proporção de acertos do cache do gerenciador de buffer
- SQLServer: expectativa de vida do gerenciador de buffer
- SQLServer: processos de estatísticas gerais bloqueados
- SQLServer: conexões de usuário de estatísticas gerais
- SQLServer: bloqueia o número de bloqueios/s
- SQLServer: solicitações em lote/s de estatísticas do SQL
- Tamanho da fila do processador do sistema

As métricas a seguir são monitoradas para os volumes anexados à instância do SQL Server:

- VolumeReadBytes
- VolumeWriteBytes
- VolumeReadOps
- VolumeWriteOps

- VolumeTotalReadTime
- VolumeTotalWriteTime
- VolumeIdleTime
- VolumeQueueLength
- VolumeThroughputPercentage
- VolumeConsumedReadWriteOps
- BurstBalance

Para esse cenário, os custos são calculados de acordo com a página CloudWatch pricing (Preços do CloudWatch) e a página SSM pricing (Preços do SSM):

- Métricas personalizadas

  Para esse cenário, 13 das métricas acima são emitidas para o CloudWatch usando o agente do CloudWatch. Essas métricas são tratadas como métricas personalizadas. O custo de cada métrica personalizada é de 0,3 USD/mês. O custo total dessas métricas personalizadas é de 13 * 0,3 USD = 3,90 USD/mês.
- Alarmes

  Para esse cenário, o CloudWatch Application Insights monitora 26 métricas no total, o que cria 26 alarmes. O custo de cada alarme é de 0,1 USD/mês. O custo total dos alarmes é de 26 * USD 0,1 = USD 2,60/mês.
- Ingestão de dados e logs de erros

  O custo da ingestão de dados é de 0,05 USD/GB e o armazenamento para o log de erros do SQL Server custa 0,03 USD/GB. O custo total da ingestão de dados e do log de erros é de 0,05 USD/GB + 0,03 USD/GB = 0,08 USD/GB.
- OpsItems do Amazon EC2 Systems Manager

  Um OpsItem do SSM é criado para cada problema detectado pelo CloudWatch Application Insights. Para n problemas na aplicação (onde n é o número de problemas), o custo total é de 0,00267 USD * n/mês.

Serviços relacionados

Os seguintes serviços são usados com o CloudWatch Application Insights:

Serviços relacionados da AWS

- O Amazon CloudWatch fornece visibilidade de todo o sistema com relação à utilização do recurso, performance da aplicação e integridade operacional. Ele coleta e rastreia métricas, envia notificações de alarmes, atualiza automaticamente os recursos que estão sendo monitorados com base nas regras definidas por você e permite monitorar suas próprias métricas personalizadas. O CloudWatch Application Insights é iniciado por meio do CloudWatch, especificamente nos painéis operacionais padrão do CloudWatch. Para mais informações, consulte o Guia do usuário do Amazon CloudWatch.
- O CloudWatch Container Insights coleta, agrega e resume métricas e logs das suas aplicações e microsserviços conteinerizados. Você pode usar o Container Insights para monitorar as plataformas Amazon ECS, Amazon Elastic Kubernetes Service e Kubernetes no Amazon EC2. Quando o Application Insights é habilitado nos consoles Container Insights ou Application Insights, o Application Insights exibe os problemas detectados no painel do Container Insights. Para obter mais informações, consulteContainer Insights
- O Amazon DynamoDB é um serviço de banco de dados NoSQL totalmente gerenciado que permite que você transfira as cargas administrativas de operação e escalabilidade de um banco de dados distribuído, para que não precise se preocupar com provisionamento, instalação e configuração de hardware, replicação, correção de software nem escalabilidade de cluster. Além disso, o DynamoDB oferece criptografia em repouso, o que elimina a carga e a complexidade operacionais envolvidas na proteção de dados confidenciais.
- O Amazon EC2 fornece capacidade de computação escalável na Nuvem AWS. É possível usar o Amazon EC2 para executar quantos servidores virtuais forem necessários, configurar a segurança e as redes e gerenciar o armazenamento. É possível ampliar ou reduzir para lidar com alterações nos requisitos ou com picos na popularidade, o que reduz a necessidade de prever o tráfego. Para obter mais informações, consulte o Guia do usuário do Amazon EC2 para instâncias do Linux ou Guia do Amazon EC2 para instâncias do Windows.
- O Amazon Elastic Block Store (Amazon EBS) oferece volumes de armazenamento em bloco para usar com instâncias do Amazon EC2. Os volumes do Amazon EBS se comportam como dispositivos de bloco brutos e não formatados. É possível montar esses volumes como dispositivos em suas instâncias. Os volumes do Amazon EBS que estão anexados a uma instância são expostos como volumes de armazenamento que persistem independentemente da vida útil da instância. É possível criar um sistema de arquivos sobre esses volumes ou utilizá-los da maneira que utilizaria um dispositivo de bloco (como um disco rígido). É possível alterar dinamicamente a configuração de um volume anexado a uma instância. Para obter mais informações, consulte o Manual do usuário da Amazon EBS.

- O Amazon EC2 Auto Scaling ajuda a garantir que você tenha o número correto de instâncias do EC2 disponíveis para processar a carga da aplicação. Para obter mais informações, consulte o Guia do usuário do Amazon EC2 Auto Scaling.
- O Elastic Load Balancing distribui aplicações de entrada ou tráfego de rede em vários destinos, como instâncias do EC2, contêineres e endereços IP em várias zonas de disponibilidade. Para obter mais informações, consulte o Manual do usuário do Elastic Load Balancing.
- O IAM é um serviço da Web que ajuda a controlar com segurança o acesso de seus usuários aos recursos da AWS. Use o IAM para controlar quem pode usar seus recursos da AWS (autenticação) e controlar os recursos que eles podem usar e como podem usá-los (autorização). Para obter mais informações, consulte Autenticação e controle de acesso para o Amazon CloudWatch.
- O AWS Lambda permite criar aplicações sem servidor compostas de funções acionadas por eventos e implantá-las automaticamente usando o CodePipeline e o AWS CodeBuild. Para obter mais informações, consulte Aplicações do AWS Lambda.
- O AWS Launch Wizard for SQL Server reduz o tempo necessário para implantar a solução de alta disponibilidade do SQL Server na nuvem. Você insere seus requisitos de aplicações, incluindo performance, número de nós e conectividade no console do serviço, e o AWS Launch Wizard identifica os recursos da AWS corretos para implantar e executar sua aplicação SQL Server Always On.
- O AWS Resource Groups ajuda a organizar os recursos que compõem a aplicação. Com o Resource Groups, é possível gerenciar e automatizar tarefas em um grande número de recursos de uma só vez. Somente um Grupo de recursos pode ser registrado para uma única aplicação. Para obter mais informações, consulte o Manual do usuário do AWS Resource Groups.
- O Amazon SQS oferece uma fila hospedada segura, durável e disponível que permite integrar e desacoplar sistemas de software e componentes distribuídos. Para obter mais informações, consulte o Guia do usuário do Amazon SQS.
- O AWS Step Functions é um compositor de função sem servidor que permite sequenciar uma variedade de produtos e recursos da AWS, inclusive funções do AWS Lambda, em fluxos de trabalho estruturados e visuais. Para mais informações, consulte o Guia do usuário do AWS Step Functions.
- O AWS SSM OpsCenter agrega e padroniza OpsItems entre serviços, fornecendo dados de investigação contextual sobre cada OpsItem, OpsItems relacionados e recursos relacionados. O OpsCenter também fornece documentos (runbooks) do Systems Manager Automation que podem ser usados para resolver problemas rapidamente. É possível especificar dados personalizados e pesquisáveis para cada OpsItem. Também é possível visualizar relatórios de resumo gerados

automaticamente sobre os OpsItems por status e origem. Para mais informações, consulte o Guia do usuário do AWS Systems Manager.

- O Amazon API Gateway é um produto da AWS para criação, publicação, manutenção, monitoramento e proteção de APIs REST e WebSocket em qualquer escala. Os desenvolvedores de API podem criar APIs que acessem a AWS ou outros serviços da Web, bem como dados armazenados na Nuvem AWS. Para obter mais informações, consulte o Manual do usuário do Amazon API Gateway.

> **Note**
>
> O Application Insights e compatível apenas com protocolos API REST (v1 do serviço API Gateway).

- O Amazon Elastic Container Service (Amazon ECS) é um serviço de orquestração de contêiner totalmente gerenciado. Você pode usar o Amazon ECS para executar suas aplicações mais sigilosas e essenciais à missão. Para obter mais informações, consulte o Guia do desenvolvedor do Amazon Elastic Container Service.
- O Amazon Elastic Kubernetes Service (Amazon EKS) é um serviço gerenciado que você pode usar para executar o Kubernetes na AWS, eliminando a necessidade de instalar e manter seus próprios nós ou seu ambiente de gerenciamento do Kubernetes. O Kubernetes é um sistema de código aberto para automatizar a implantação, a escalabilidade e o gerenciamento de aplicações em contêineres. Para obter mais informações, consulte o Manual do usuário do Amazon EKS.
- Kubernetes no Amazon EC2. O Kubernetes é um software de código aberto que ajuda a implantar e gerenciar aplicações em contêineres em escala. O Kubernetes gerencia clusters de instâncias de computação do Amazon EC2 e executa contêineres nessas instâncias com processos para implantação, manutenção e escalabilidade. Com o Kubernetes, é possível executar qualquer tipo de aplicação em contêineres com o mesmo conjunto de ferramentas on-premises e na nuvem. Para obter mais informações, consulte Documentação do Kubernetes: conceitos básicos.
- O Amazon FSx ajuda a iniciar e executar sistemas de arquivos bastante utilizados que são totalmente gerenciados pela AWS. Com o Amazon FSx, é possível aproveitar os conjuntos de recursos e a performance de sistemas de arquivos comuns de código aberto e licenciados comercialmente para evitar tarefas administrativas demoradas. Para mais informações, consulte a documentação do Amazon FSx.
- O Amazon Simple Notification Service (SNS) é um serviço de mensagens totalmente gerenciado para comunicação entre aplicações e de aplicação para pessoa. Você pode configurar o Amazon SNS para monitoramento pelo Application Insights. Quando o Amazon SNS é configurado como

um recurso de monitoramento, o Application Insights rastreia as métricas do SNS para ajudar a determinar a causa de possíveis problemas e falhas em mensagens do SNS.

- O Amazon Elastic File System (Amazon EFS) é um sistema de arquivos NFS elástico totalmente gerenciado para uso com serviços e recursos on-premises da Nuvem AWS. Ele foi criado para escalar até petabytes sob demanda sem interromper os aplicativos. Ele aumenta e diminui automaticamente à medida que arquivos são adicionados e removidos, o que elimina a necessidade de provisionar e gerenciar a capacidade para acomodar o crescimento. Para obter mais informações, consulte a documentação do Amazon Elastic File System.

Serviços relacionados de terceiros

- Para algumas workloads e aplicações monitoradas no Application Insights, o Prometheus JMX Exporter é instalado usando o AWS Systems Manager Distributor para que o CloudWatch Application Insights possa recuperar métricas específicas do Java. Quando você escolhe monitorar uma aplicação Java, o Application Insights instala automaticamente o Prometheus JMX Exporter para você.

Componentes da aplicação com suporte

O CloudWatch Application Insights verifica seu grupo de recursos para identificar os componentes da aplicação. Os componentes podem ser individuais, agrupados automaticamente (como instâncias em um grupo de Auto Scaling ou por trás de um load balancer) ou personalizados (agrupando instâncias do Amazon EC2 individuais).

Os seguintes componentes são compatíveis com o CloudWatch Application Insights:

Componentes da AWS

- Amazon EC2
- Amazon EBS
- Amazon RDS
- Elastic Load Balancing: Application Load Balancer e Classic Load Balancer (todas as instâncias de destino desses balanceadores de carga são identificadas e configuradas).
- Grupos do Amazon EC2 Auto Scaling: AWS Auto Scaling (os grupos do Auto Scaling são configurados dinamicamente para todas as instâncias de destino; se a aplicação aumentar a escala na vertical, o CloudWatch Application Insights configurará automaticamente as novas

instâncias). Os grupos do Auto Scaling não têm suporte para grupos de recursos baseados em pilha do CloudFormation.
- AWS Lambda
- Amazon Simple Queue Service (Amazon SQS)
- Tabela do Amazon DynamoDB
- Métricas de bucket do Amazon S3
- AWS Step Functions
- Etapas da API REST do Amazon API Gateway
- Amazon Elastic Container Service (Amazon ECS): cluster, serviço e tarefa
- Amazon Elastic Kubernetes Service (Amazon EKS): cluster
- Kubernetes no Amazon EC2: cluster do Kubernetes em execução no EC2
- Tópico do Amazon SNS

Atualmente, nenhum outro recurso do tipo de componente é rastreado pelo CloudWatch Application Insights. Se um tipo de componente que é compatível não aparecer na sua aplicação do Application Insights, pode ser que ele já esteja registrado e sendo gerenciado por outra aplicação sua monitorada pelo Application Insights.

Pilhas de tecnologia compatíveis

É possível usar o CloudWatch Application Insights para monitorar suas aplicações executadas em sistemas operacionais Windows Server e Linux selecionando a opção de menu suspenso da camada de aplicações para uma das seguintes tecnologias:

- Front-end: servidor web do Microsoft Internet Information Services (IIS)
- Nível do operador:
  - NET Framework
  - .NET Core
- Aplicações:
  - Java
  - Implantações padrão, distribuídas e de alta disponibilidade do SAP NetWeaver
- Active Directory
- SharePoint

- Bancos de dados:
  - Microsoft SQL Server em execução no Amazon RDS ou no Amazon EC2 (incluindo configurações de alta disponibilidade do SQL Server. Consulte Exemplos de configuração do componente).
  - MySQL em execução no Amazon RDS, no Amazon Aurora ou no Amazon EC2
  - PostgreSQL em execução no Amazon RDS ou no Amazon EC2
  - Tabela do Amazon DynamoDB
  - Oracle em execução no Amazon RDS ou no Amazon EC2
  - Banco de dados SAP HANA em uma única instância do Amazon EC2 e várias instâncias do EC2
  - Configuração de alta disponibilidade do banco de dados SAP HANA entre zonas de disponibilidade
  - Banco de dados SAP Sybase ASE em uma única instância do Amazon EC2
  - Configuração de alta disponibilidade do banco de dados SAP Sybase ASE entre zonas de disponibilidade

Se nenhuma das pilhas de tecnologia listadas acima se aplicar aos recursos da aplicação, será possível monitorar a pilha de aplicações escolhendo Custom (Personalizar) no menu suspenso da camada de aplicação na página Manage monitoring (Gerenciar monitoramento).

Trabalhos do Amazon CloudWatch Application Insights

Esta seção contém informações sobre como o CloudWatch Application Insights funciona, incluindo:

- Como Application Insights monitora aplicações
- Retenção de dados
- Cotas
- Pacotes do AWSSystems Manager (SSM) usados pelo CloudWatch Application Insights
- Documentos do AWS Systems Manager (SSM) usados pelo CloudWatch Application Insights

Como Application Insights monitora aplicações

O Application Insights monitora aplicações conforme é mostrado a seguir.

Configuração e descoberta de aplicação

Na primeira vez em que uma aplicação é adicionada ao CloudWatch Application Insights, ele verifica os componentes da aplicação com relação aos principais logs, métricas e outras fontes de dados recomendados que devem ser monitorados para a aplicação. É possível, então, configurar a aplicação com base nessas recomendações.

Pré-processamento de dados

O CloudWatch Application Insights analisa continuamente as fontes de dados que estão sendo monitoradas em todos os recursos da aplicação para detectar anomalias de métrica e erros de log (observações).

Detecção de problema inteligente

O mecanismo do CloudWatch Application Insights detecta problemas em sua aplicação ao correlacionar observações usando algoritmos de classificação e regras integradas. Para auxiliar na solução de problemas, ele cria painéis automatizados do CloudWatch, que incluem informações contextuais sobre os problemas.

Alerta e ação

Quando o CloudWatch Application Insights detecta um problema em sua aplicação, ele gera CloudWatch Events para notificar você sobre o problema. Consulte CloudWatch Events do Application Insights e notificações de problemas detectados para obter mais informações sobre como configurar esses Eventos.

Exemplo de cenário

Você tem uma aplicação ASP .NET baseado em um banco de dados do SQL Server. Repentinamente, o banco de dados começa a apresentar mau funcionamento por causa de alta pressão de memória. Isso causa uma degradação na performance da aplicação e possivelmente erros HTTP 500 nos servidores web e no load balancer.

Com o CloudWatch Application Insights e sua análise inteligente, é possível identificar a camada da aplicação que está causando o problema conferindo o painel criado dinamicamente que exibe as métricas relacionadas e os trechos do arquivo de log. Nesse caso, o problema pode estar na camada de banco de dados do SQL.

Retenção de dados

O CloudWatch Application Insights retém problemas por 55 dias e observações por 60 dias.

Cotas

Para obter cotas padrão do CloudWatch Application Insights, consulte endpoints e cotas do Amazon CloudWatch Application Insights. A menos que especificado de outra forma, cada cota é por região da AWS. Entre em contato com o AWS Support para solicitar aumento na cota de serviço. Muitos serviços têm cotas que não podem ser alteradas. Para obter mais informações sobre as cotas de um serviço específico, consulte a documentação desse serviço.

Pacotes do AWSSystems Manager (SSM) usados pelo CloudWatch Application Insights

Os pacotes listados nesta seção são usados pelo Application Insights e podem ser gerenciados e implantados de forma independente com o AWS Systems Manager Distributor. Para obter mais informações sobre o SSM Distributor, consulteAWS Systems Manager Distributor no Manual do usuário do AWS Systems Manager.

Pacotes:

- AWSObservabilityExporter-JMXExporterInstallAndConfigure
- AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure
- AWSObservabilityExporter-HAClusterExporterInstallAndConfigure
- AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure
- AWSObservabilityExporter-SQLExporterInstallAndConfigure

`AWSObservabilityExporter-JMXExporterInstallAndConfigure`

É possível recuperar métricas Java específicas de workloads a partir do Prometheus JMX Exporter para que o Application Insights configure e monitore alarmes. No console do Application Insights, na página Manage monitoring (Gerenciar monitoramento) Página, JAVA application (Aplicação JAVA) da lista suspensa Application tier (Nível da aplicação). Em JAVA Prometheus exporter configuration (Configuração do JAVA Prometheus Exporter), selecione seu Collection method (Método de coleta) e JMX port number (Número da porta JMX).

Para usar o AWS Systems Manager Distributor para empacotar, instalar e configurar o pacote do Prometheus JMX Exporter fornecido pela AWS de modo independente do Application Insights, realize as etapas a seguir.

Pré-requisitos para usar o pacote SSM do Prometheus JMX Exporter

- SSM Agent versão 2.3.1550.0 ou posterior instalado
- A variável de ambiente JAVA_HOME está definida

Instalar e configurar o pacote **`AWSObservabilityExporter-JMXExporterInstallAndConfigure`**

O pacote `AWSObservabilityExporter-JMXExporterInstallAndConfigure` é um pacote do SSM Distributor que você pode usar para instalar e configurar o Prometheus JMX Exporter. Quando as métricas Java são enviadas pelo Prometheus JMX Exporter, é possível configurar o atendente do CloudWatch ppara recuperar as métricas do serviço CloudWatch.

1. Com base em suas preferências, prepare o arquivo de configuração YAML do Prometheus JMX Exporter localizado no repositório do Prometheus no GitHub. O exemplo de configuração e as descrições de opções podem ser usados para orientar você.
2. Copie o arquivo de configuração YAML do Prometheus JMX Exporter codificado como Base64 para um novo parâmetro SSM em SSM Parameter Store (Armazenamento de parâmetros do SSM).
3. Navegue até o console do SSM Distributor e abra a guia Owned by Amazon (Propriedade da Amazon). Selecione AWSObservabilityExporter-JMXExporterInstallAndConfigure e escolha Install one time (Instalar uma vez).
4. Atualize o parâmetro do SSM criado na primeira etapa substituindo “Additional Arguments” (Argumentos adicionais) por:

```
{
  "SSM_EXPORTER_CONFIGURATION": "{{ssm:<SSM_PARAMETER_STORE_NAME>}}",
  "SSM_EXPOSITION_PORT": "9404"
}
```

> **Note**
>
> A porta 9404 é a porta padrão usada para enviar métricas do Prometheus JMX. É possível atualizá-la.

Exemplo: configurar o atendente do CloudWatch para recuperar métricas do Java

1. Instale o JMX Exporter do Prometheus, conforme descrito no procedimento anterior. Em seguida, confira o status da porta para verificar que ele está instalado corretamente em sua instância.

   Exemplo de instalação bem-sucedida na instância do Windows

   ```
   PS C:\> curl http://localhost:9404 (http://localhost:9404/)
   StatusCode : 200
   StatusDescription : OK
   Content : # HELP jvm_info JVM version info
   ```

   Exemplo de instalação bem-sucedida na instância do Linux

   ```
   $ curl localhost:9404
   # HELP jmx_config_reload_failure_total Number of times configuration have failed to
    be reloaded.
   # TYPE jmx_config_reload_failure_total counter
   jmx_config_reload_failure_total 0.0
   ```

2. Crie o arquivo YAML de detecção de serviço do Prometheus. O exemplo de arquivo de detecção de serviço abaixo executa o seguinte:

   - Especifica a porta de host do Prometheus JMX Exporter como `localhost: 9404`.
   - Anexa rótulos (`Application`, `ComponentName` e `InstanceId`) para as métricas, que podem ser definidas como dimensões métricas do CloudWatch.

   ```
   $ cat prometheus_sd_jmx.yaml
   - targets:
     - 127.0.0.1:9404
     labels:
       Application: myApp
       ComponentName: arn:aws:elasticloadbalancing:us-
   east-1:123456789012:loadbalancer/app/sampl-Appli-MMZW8E3GH4H2/aac36d7fea2a6e5b
       InstanceId: i-12345678901234567
   ```

3. Crie o arquivo YAML de configuração do Prometheus JMX Exporter. O exemplo de arquivo de configuração a seguir especifica:

- O intervalo de trabalho de recuperação de métricas e o período de tempo limite.
- Os trabalhos de recuperação de métricas (`jmx` e `sap`), também conhecido como “scraping”, que incluem o nome do trabalho, o máximo de séries temporais retornadas por vez e o caminho do arquivo de detecção de serviço.

```
$ cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    file_sd_configs:
      - files: ["/tmp/prometheus_sd_jmx.yaml"]
  - job_name: sap
    sample_limit: 10000
    file_sd_configs:
      - files: ["/tmp/prometheus_sd_sap.yaml"]
```

4. Verifique se o atendente do CloudWatch está instalado em sua instância do Amazon EC2 e se a versão é 1.247346.1b249759 ou posterior. Para instalar o atendente do CloudWatch em sua instância do EC2, consulte Instalar o atendente do CloudWatch. Para verificar a versão, consulte Encontrar informações sobre versões do atendente do CloudWatch.
5. Configure o atendente do CloudWatch. Para obter mais informações sobre como configurar o arquivo de configuração do atendente do CloudWatch, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch. O exemplo do arquivo de configuração do atendente do CloudWatch abaixo executa o seguinte:

   - Especifica o caminho do arquivo de configuração do Prometheus JMX Exporter.
   - Especifica o grupo de logs de destino em que os logs de métrica do EMF serão publicados.
   - Especifica dois conjuntos de dimensões para cada nome da métrica.
   - Envia 8 métricas do CloudWatch (4 nomes da métricas * 2 conjuntos de dimensões por nome da métrica).

```
{
   "logs":{
      "logs_collected":{
```

```
        ....
    },
    "metrics_collected":{
       "prometheus":{
          "cluster_name":"prometheus-test-cluster",
          "log_group_name":"prometheus-test",
          "prometheus_config_path":"/tmp/prometheus.yaml",
          "emf_processor":{
             "metric_declaration_dedup":true,
             "metric_namespace":"CWAgent",
             "metric_unit":{
                "jvm_threads_current":"Count",
                "jvm_gc_collection_seconds_sum":"Second",
                "jvm_memory_bytes_used":"Bytes"
             },
             "metric_declaration":[
                {
                   "source_labels":[
                      "job"
                   ],
                   "label_matcher":"^jmx$",
                   "dimensions":[
                      [
                         "InstanceId",
                         "ComponentName"
                      ],
                      [
                         "ComponentName"
                      ]
                   ],
                   "metric_selectors":[
                      "^java_lang_threading_threadcount$",
                      "^java_lang_memory_heapmemoryusage_used$",
                      "^java_lang_memory_heapmemoryusage_committed$"
                   ]
                }
             ]
          }
       }
    }
 },
 "metrics":{
    ....
 }
```

```
}
```

AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure

É possível recuperar métricas SAP HANA específicas de workloads a partir do exportador de banco de dados Prometheus HANA para que o Application Insights configure e monitore alarmes. Para obter mais informações, consulte Configurar seu banco de dados SAP HANA para monitoramento neste guia.

Para usar o AWS Systems Manager Distributor para empacotar, instalar e configurar o pacote do exortador de banco de dados Prometheus HANA fornecido pela AWS de maneira independente do Application Insights, realize as etapas a seguir.

Pré-requisitos para usar o pacote SSM do exportador de banco de dados Prometheus HANA

- SSM Agent versão 2.3.1550.0 ou posterior instalado
- Banco de dados SAP HANA
- Sistema operacional Linux (SUSE Linux, RedHat Linux)
- Um segredo com credenciais de monitoramento de banco de dados SAP HANA usando AWS Secrets Manager. Crie um segredo no formato de pares chave-valor, especifique o nome de usuário da chave e insira o usuário do banco de dados para o valor. Adicione uma segunda senha da chave e, em seguida, digite a senha para o valor. Para obter mais informações sobre criação de segredos, consulte Criar um segredo no Guia do usuário do AWS Secrets Manager. O segredo deve ser formatado da seguinte forma:

```
{
  "username": "<database_user>",
  "password": "<database_password>"
}
```

Instalar e configurar o pacote **AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure**

O pacote `AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure` é um pacote do SSM Distributor que você pode usar para instalar e configurar o exportador de banco de dados HANA do Prometheus. Quando as métricas Java são enviadas pelo exportador de banco de

dados HANA do Prometheus, é possível configurar o atendente do CloudWatch para recuperar as métricas do serviço CloudWatch.

1. Criar um parâmetro do SSM no Parameter Store do SSM para armazenar as configurações do Exportador. Veja a seguir um exemplo do valor de parâmetro.

```
{\"exposition_port\":9668,\"multi_tenant\":true,\"timeout\":600,\"hana\":{\"host\":
\"localhost\",\"port\":30013,\"aws_secret_name\":\"HANA_DB_CREDS\",\"scale_out_mode
\":true}}
```

> **Note**
>
> Neste exemplo, a exportação é executada somente na instância do Amazon EC2 com o banco de dados SYSTEM ativo. Ele permanecerá ocioso nas outras instâncias do EC2 para evitar métricas duplicadas. O exportador pode recuperar todas as informações do inquilino do banco de dados a partir do banco de dados SYSTEM.

2. Criar um parâmetro do SSM no Parameter Store do SSM para armazenar as consultas de métricas do exportador. O pacote pode aceitar mais de um parâmetro de métricas. Cada parâmetro deve ter um formato de objeto JSON válido. Veja a seguir um exemplo do valor de parâmetro.

```
{\"SELECT MAX(TIMESTAMP) TIMESTAMP, HOST, MEASURED_ELEMENT_NAME CORE,
 SUM(MAP(CAPTION, 'User Time', TO_NUMBER(VALUE), 0)) USER_PCT, SUM(MAP(CAPTION,
 'System Time', TO_NUMBER(VALUE), 0)) SYSTEM_PCT, SUM(MAP(CAPTION, 'Wait
 Time', TO_NUMBER(VALUE), 0)) WAITIO_PCT, SUM(MAP(CAPTION, 'Idle Time', 0,
 TO_NUMBER(VALUE))) BUSY_PCT, SUM(MAP(CAPTION, 'Idle Time', TO_NUMBER(VALUE), 0))
 IDLE_PCT FROM sys.M_HOST_AGENT_METRICS WHERE MEASURED_ELEMENT_TYPE = 'Processor'
 GROUP BY HOST, MEASURED_ELEMENT_NAME;\":{\"enabled\":true,\"metrics\":[{\"name\":
\"hanadb_cpu_user\",\"description\":\"Percentage of CPU time spent by HANA DB in user
 space, over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":
\"USER_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_system
\",\"description\":\"Percentage of CPU time spent by HANA DB in Kernel space,
 over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":
\"SYSTEM_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_waitio
\",\"description\":\"Percentage of CPU time spent by HANA DB in IO mode, over the
 last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":\"WAITIO_PCT\",
\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_busy\",\"description
\":\"Percentage of CPU time spent by HANA DB, over the last minute (in seconds)\",
\"labels\":[\"HOST\",\"CORE\"],\"value\":\"BUSY_PCT\",\"unit\":\"percent\",\"type\":
```

```
\"gauge\"},{\"name\":\"hanadb_cpu_idle\",\"description\":\"Percentage of CPU time not
 spent by HANA DB, over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE
\"],\"value\":\"IDLE_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"}]}}
```

Para obter mais informações sobre consultas de métricas, consulte o repositório SUSE / hanadb_exporter no GitHub.

3. Navegue até o console do SSM Distributor e abra a guia Owned by Amazon (Propriedade da Amazon). Selecione AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure* e escolha Install one time (Instalar uma vez).
4. Atualize o parâmetro do SSM criado na primeira etapa substituindo “Additional Arguments” (Argumentos adicionais) por:

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<*SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>*}}",
  "SSM_SID": "<SAP_DATABASE_SID>",
  "SSM_EXPORTER_METRICS_1": "{{ssm:<SSM_FIRST_METRICS_PARAMETER_STORE_NAME>}}",
  "SSM_EXPORTER_METRICS_2": "{{ssm:<SSM_SECOND_METRICS_PARAMETER_STORE_NAME>}}"
}
```

5. Selecione as instâncias do Amazon EC2 com banco de dados SAP HANA e escolha Run (Executar).

`AWSObservabilityExporter-HAClusterExporterInstallAndConfigure`

Você pode recuperar métricas de cluster de alta disponibilidade (HA) específicas de workloads do exportador de clusters HANA do Prometheus para o Application Insights a fim de configurar e monitorar alarmes para uma configuração de alta disponibilidade do banco de dados SAP HANA. Para obter mais informações, consulte Configurar seu banco de dados SAP HANA para monitoramento neste guia.

Para usar o AWS Systems Manager Distributor para empacotar, instalar e configurar o pacote do exportador de cluster de HA do Prometheus fornecido pela AWS de modo independente do Application Insights, realize as etapas a seguir.

Pré-requisitos para usar o pacote SSM do Prometheus HA cluster exporter

- SSM Agent versão 2.3.1550.0 ou posterior instalado
- Cluster HA para Pacemaker, Corosync, SBD e DRBD
- Sistema operacional Linux (SUSE Linux, RedHat Linux)

Instalar e configurar o pacote **AWSObservabilityExporter-HAClusterExporterInstallAndConfigure**

O pacote `AWSObservabilityExporter-HAClusterExporterInstallAndConfigure` é um pacote do SSM Distributor que você pode usar para instalar e configurar o Prometheus HA Cluster Exporter. Quando as métricas de cluster são enviadas pelo exportador do banco de dados HANA do Prometheus, é possível configurar o atendente do CloudWatch para recuperar as métricas do serviço CloudWatch.

1. Criar um parâmetro do SSM no Parameter Store do SSM para armazenar as configurações do exportador no formato JSON. Veja a seguir um exemplo do valor de parâmetro.

   ```
   {\"port\":\"9664\",\"address\":\"0.0.0.0\",\"log-level\":\"info\",\"crm-mon-path
   \":\"/usr/sbin/crm_mon\",\"cibadmin-path\":\"/usr/sbin/cibadmin\",\"corosync-
   cfgtoolpath-path\":\"/usr/sbin/corosync-cfgtool\",\"corosync-quorumtool-path\":\"/
   usr/sbin/corosync-quorumtool\",\"sbd-path\":\"/usr/sbin/sbd\",\"sbd-config-path\":
   \"/etc/sysconfig/sbd\",\"drbdsetup-path\":\"/sbin/drbdsetup\",\"enable-timestamps
   \":false}
   ```

   Para obter mais informações sobre a configuração do exportador, consulte o repositório `ClusterLabs / ha_cluster_exporter` no GitHub.
2. Navegue até o console do SSM Distributor e abra a guia Owned by Amazon (Propriedade da Amazon). Selecione AWSOBservabilityExporter-HaclusterExporterInstallAndConfigure* e escolha Install one time (Instalar uma vez).
3. Atualize o parâmetro do SSM criado na primeira etapa substituindo “Additional Arguments” (Argumentos adicionais) por:

   ```
   {
     "SSM_EXPORTER_CONFIG": "{{ssm:<*SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>*}}"
   }
   ```
4. Selecione as instâncias do Amazon EC2 com banco de dados SAP HANA e escolha Run (Executar).

AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure

Você pode recuperar, do exportador do Prometheus para host do SAP, as métricas do SAP NetWeaver específicas da workload, para que o Application Insights configure e monitore alarmes

para as implantações distribuídas e de alta disponibilidade do SAP NetWeaver. Para ter mais informações, consulte Comece a usar o Amazon CloudWatch Application Insights.

Para usar o AWS Systems Manager Distributor para empacotar, instalar e configurar o pacote do exportador do host SAP independentemente do Application Insights, conclua as etapas a seguir.

Pré-requisitos para usar o pacote do SSM de exportador do host SAP Prometheus

- SSM Agent versão 2.3.1550.0 ou posterior instalado
- Servidores de aplicações do SAP NetWeaver
- Sistema operacional Linux (SUSE Linux, RedHat Linux)

Instalar e configurar o pacote **`AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`**

O pacote `AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure` é um pacote do SSM Distributor que você pode usar para instalar e configurar o exportador de métricas Prometheus do SAP NetWeaver. Quando as métricas do SAP NetWeaver são enviadas pelo exportador do Prometheus, o agente do CloudWatch pode ser configurado para recuperar as métricas para o serviço CloudWatch.

1. Criar um parâmetro do SSM no Parameter Store do SSM para armazenar as configurações do exportador no formato JSON. Veja a seguir um exemplo do valor de parâmetro.

   ```
   {\"address\":\"0.0.0.0\",\"port\":\"9680\",\"log-level\":\"info\",\"is-HA\":false}
   ```

   - endereço

     O endereço de destino para o qual o Prometheus envia métricas. O valor padrão é `localhost`.

   - port

     A porta de destino para a qual enviar métricas do Prometheus. O valor padrão é `9680`.

   - is-HA

     `true` para implantações de alta disponibilidade do SAP NetWeaver. Para todas as outras implantações, o valor é `false`.

2. Navegue até o console do SSM Distributor e abra a guia Owned by Amazon (Propriedade da Amazon). Selecione AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure e escolha Install one time (Instalar uma vez).
3. Atualize o parâmetro do SSM criado na primeira etapa substituindo “Additional Arguments” (Argumentos adicionais) por:

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>}}",
  "SSM_SID": "<SAP_DATABASE_SID>",
  "SSM_INSTANCES_NUM": "<instances_number seperated by comma>"
}
```

   Exemplo

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:exporter_config_paramter}}",
  "SSM_INSTANCES_NUM": "11,12,10",
  "SSM_SID": "PR1"
}
```

4. Selecione as instâncias do Amazon EC2 com aplicações do SAP NetWeaver e escolha Run (Executar).

> **Note**
>
> O exportador do Prometheus processa as métricas do SAP NetWeaver em um endpoint local. O endpoint local só pode ser acessado pelos usuários do sistema operacional na instância do Amazon EC2. Assim, após a instalação do pacote do exportador, as métricas ficam disponíveis para todos os usuários do sistema operacional. O endpoint local padrão é o `localhost:9680/metrics`.

`AWSObservabilityExporter-SQLExporterInstallAndConfigure`

É possível recuperar métricas de SQL Server específicas de workloads a partir do Exportador do Prometheus SQL para que o Application Insights monitore os principais alarmes.

Para usar o Distribuidor do AWS Systems Manager para empacotar, instalar e configurar o pacote do exportador do SQL independentemente do Application Insights, conclua as etapas a seguir.

Pré-requisitos para usar o pacote SSM do exportador do Prometheus SQL

- SSM Agent versão 2.3.1550.0 ou posterior instalado
- Instância do Amazon EC2 executando o SQL Server no Windows com a autenticação de usuário do SQL Server ativada.
- Um usuário do SQL Server com as permissões a seguir:

```
GRANT VIEW ANY DEFINITION TO
```

```
GRANT VIEW SERVER STATE TO
```

- Um segredo contendo a string de conexão do banco de dados usando AWS Secrets Manager. Para obter mais informações sobre criação de segredos, consulte Criar um segredo no Guia do usuário do AWS Secrets Manager. O segredo deve ser formatado da seguinte forma:

```
{
  "data_source_name":"sqlserver://<username>:<password>@localhost:1433"
}
```

> **Note**
>
> Se a senha ou o nome de usuário contiverem caracteres especiais, você deverá codificar com porcento os caracteres especiais para garantir uma conexão com êxito ao banco de dados.

Instalar e configurar o pacote **AWSObservabilityExporter-SQLExporterInstallAndConfigure**

O pacote `AWSObservabilityExporter-SQLExporterInstallAndConfigure` é um pacote do SSM Distributor que pode ser usado para instalar e configurar o exportador de métricas SQL Prometheus. Quando as métricas são enviadas pelo exportador do Prometheus, é possível configurar o agente do CloudWatch para recuperar as métricas do serviço CloudWatch.

1. Com base nas suas preferências, prepare a configuração YAML do SQL Exporter. O exemplo de configuração a seguir tem uma única métrica configurada. Use a configuração de exemplo para atualizar a configuração com métricas adicionais ou criar sua própria configuração.

```
---
global:
  scrape_timeout_offset: 500ms
  min_interval: 0s
  max_connections: 3
  max_idle_connections: 3
target:
  aws_secret_name: <SECRET_NAME>
  collectors:
    - mssql_standard
collectors:
  - collector_name: mssql_standard
    metrics:
    - metric_name: mssql_batch_requests
      type: counter
      help: 'Number of command batches received.'
      values: [cntr_value]
      query: |
        SELECT cntr_value
        FROM sys.dm_os_performance_counters WITH (NOLOCK)
        WHERE counter_name = 'Batch Requests/sec'
```

2. Copie o arquivo de configuração YAML do exportador do Prometheus SQL codificado como Base64 para um novo parâmetro SSM em Armazenamento de parâmetros do SSM.
3. Navegue até o console do SSM Distributor e abra a guia Owned by Amazon (Propriedade da Amazon). Selecione AWSObservabilityExporter-SQLExporterInstallAndConfigure e escolha Instalar uma vez.
4. Substitua os "Argumentos adicionais" pelas informações a seguir. O `SSM_PARAMETER_NAME` é o nome do parâmetro que foi criado na Etapa 2.

```
{
  "SSM_EXPORTER_CONFIGURATION":
    "{{ssm:<SSM_PARAMETER_STORE_NAME>}}",
    "SSM_PROMETHEUS_PORT": "9399",
    "SSM_WORKLOAD_NAME": "SQL"
}
```

5. Selecione a instância do Amazon EC2 com o banco de dados SQL Server e escolha executar.

Documentos do AWS Systems Manager (SSM) usados pelo CloudWatch Application Insights

O Application Insights usa os documentos do SSM listados nesta seção para definir as ações que o AWS Systems Manager executa em suas instâncias gerenciadas. Esses documentos utilizam a funcionalidade `Run Command` do Systems Manager para automatizar as tarefas necessárias para a execução das funcionalidades de monitoramento do Application Insights. As programações de execuções para esses documentos são mantidas pelo Application Insights e não podem ser alteradas.

Para obter mais informações sobre os documentos do SSM, consulte Documentos do AWS Systems Manager no Guia do usuário do AWS Systems Manager.

Documentos gerenciados pelo CloudWatch Application Insights

A tabela apresentada a seguir lista os documentos do SSM que são gerenciados pelo Application Insights.

Nome do documento	Descrição	Programação de execução
`AWSEC2-DetectWorkload`	Detecta automaticamente as aplicações em execução no seu ambiente de aplicações que podem ser configuradas para serem monitoradas pelo Application Insights.	Esse documento é executado de hora em hora no ambiente de aplicações para obter detalhes atualizados das aplicações.
`AWSEC2-CheckPerformanceCounterSets`	Verifica se os namespaces do contador de performance estão habilitados nas instâncias do Windows do Amazon EC2.	Esse documento é executado de hora em hora no seu ambiente de aplicações e vai monitorar somente as métricas do contador de performance se os namespaces correspondentes estiverem habilitados.
`AWSEC2-ApplicationInsightsCloudwatch`	Instala e configura o agente do CloudWatch com base na configuração de monitoram	Esse documento é executado a cada 30 minutos para garantir que a configuração do

Nome do documento	Descrição	Programação de execução
`AgentInstallAndCon figure`	ento dos componentes da aplicação.	agente do CloudWatch esteja sempre precisa e atualizad a. O documento também é executado imediatamente após uma alteração ser realizada na configuração de monitoramento da aplicação, como a adição ou a remoção de métricas ou a atualização das configurações de log.

Documentos gerenciados pelo AWS Systems Manager

Os documentos apresentados a seguir são usados pelo CloudWatch Application Insights e gerenciados pelo Systems Manager.

`AWS-ConfigureAWSPackage`

O Application Insights usa esse documento para instalar e desinstalar pacotes de distribuidores exportadores do Prometheus, para coletar métricas específicas da workload e para habilitar o monitoramento abrangente de workloads em instâncias do Amazon EC2 do cliente. O CloudWatch Application Insights instalará os pacotes de distribuidores exportadores do Prometheus somente se a workload de destino correlacionada estiver em execução na sua instância.

A tabela apresentada a seguir lista os pacotes de distribuidores exportadores do Prometheus e as workloads de destino correlacionadas.

Nome do pacote de distribuidor exportador do Prometheus	Workload de destino
`AWSObservabilityExporter-HA ClusterExporterInstallAndCo nfigure`	SAP HANA HA
`AWSObservabilityExporter-JM XExporterInstallAndConfigure`	Java/JMX

Nome do pacote de distribuidor exportador do Prometheus	Workload de destino
`AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure`	SAP HANA
`AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`	NetWeaver
`AWSObservabilityExporter-SQLExporterInstallAndConfigure`	SQL Server (Windows) e SAP ASE (Linux)

`AmazonCloudWatch-ManageAgent`

O Application Insights usa esse documento para gerenciar o status e a configuração do agente do CloudWatch em suas instâncias e para coletar métricas e logs internos no nível do sistema usando as instâncias do Amazon EC2 em sistemas operacionais.

Comece a usar o Amazon CloudWatch Application Insights

Para começar a usar o CloudWatch Application Insights, verifique se você cumpriu os pré-requisitos descritos abaixo e criou uma política do IAM. Em seguida, você pode começar a usar o link do console para habilitar o CloudWatch Application Insights. Para configurar os recursos da aplicação, siga as etapas em Instalar, configurar e gerenciar sua aplicação para monitoramento.

Conteúdo

Acessar o CloudWatch Application Insights

Você pode acessar o CloudWatch Application Insights, por meio de uma das seguintes interfaces:

- Console do CloudWatch. Para adicionar monitores às aplicações, escolha Application Insights em Insights no painel de navegação esquerdo do console do CloudWatch. Depois que a aplicação estiver configurado, você poderá usar o console do CloudWatch para visualizar e analisar problemas que forem detectados.
- Interface de linha de comando da AWS (AWS CLI). Você pode usar a AWS CLI para acessar as operações de API da AWS. Para obter mais informações, consulte Instalar a AWS Command Line Interface no Manual do usuário da AWS Command Line Interface. Para obter mais informações sobre as APIs do Application Insights, consulte a Referência de API do Amazon CloudWatch Application Insights.

Pré-requisitos

É necessário concluir os seguintes pré-requisitos para configurar uma aplicação com o CloudWatch Application Insights:

- Capacitação do AWS Systems Manager: instale o Systems Manager Agent (SSM Agent) nas suas instâncias do Amazon EC2 e capacite-as para o SSM. Para obter informações sobre como instalar o SSM Agent, consulte Configurando o AWS Systems Manager, no Guia do usuário do AWS Systems Manager.
- Perfil de instância do Amazon EC2: você deve anexar os perfis de instância do Amazon EC2 a seguir para habilitar o Systems Manager
  - Você deve anexar o perfil do `AmazonSSMManagedInstanceCore` para habilitar o Systems Manager. Para obter mais informações, consulte exemplos de políticas baseadas em identidade do AWS Systems Manager.
  - Você deve anexar a política do `CloudWatchAgentServerPolicy` para permitir que métricas e logs de instâncias sejam emitidos por meio do CloudWatch. Para obter mais informações, consulte Criar perfis e usuários do IAM para uso com o agente do CloudWatch.
- Grupos de recursos do AWS – Para integrar suas aplicações ao CloudWatch Application Insights, crie um grupo de recursos da AWS que inclua todos os recursos associados usados por sua pilha de aplicações. Isso inclui Application Load Balancers, instâncias do Amazon EC2 que executam o IIS e o front-end da Web, níveis de operador do .NET e bancos de dados SQL Server. Para obter mais informações sobre componentes de aplicações e pilhas de tecnologia compatíveis com o Application Insights, consulte Componentes da aplicação com suporte. O CloudWatch Application Insights inclui automaticamente grupos do Auto Scaling usando as mesmas pilhas do CloudFormation ou etiquetas que seu grupo de recursos, pois grupos do Auto Scaling não são

compatíveis com o grupos de recursos do CloudFormation. Para obter mais informações, consulte Conceitos básicos do AWS Resource Groups.

- Permissões do IAM: para usuários que não têm acesso administrativo, você deve criar uma política do AWS Identity and Access Management (IAM) que permita ao Application Insights criar um perfil vinculado ao serviço e anexá-lo à identidade do usuário. Para obter informações sobre como criar uma política do IAM, consulte Política do IAM.
- Perfil vinculado ao serviço: o Application Insights usa os perfis vinculadas ao serviço do AWS Identity and Access Management (IAM). Ao criar a primeira aplicação com o Application Insights, é criada uma função vinculada ao serviço no console do Application Insights. Para ter mais informações, consulte Usar funções vinculadas ao serviço do CloudWatch Application Insights.
- Suporte às métricas de contador de performance para instâncias do Windows do EC2: para monitorar as métricas de contador de performance em suas instâncias do Windows do Amazon EC2, os contadores de performance devem ser instalados nas instâncias. Para métricas de contador de performance e nomes de conjuntos de contadores de performance correspondentes, consulte Métricas de contador de performance. Para obter mais informações sobre contadores de performance, consulte Contadores de performance.
- Agente do Amazon CloudWatch: o Application Insights instala e configura o agente do CloudWatch. Se você tiver o agente CloudWatch instalado, o Application Insights manterá sua configuração. Para evitar um conflito de mesclagem, remova a configuração de recursos que você deseja usar no Application Insights do arquivo de configuração do agente do CloudWatch existente. Para ter mais informações, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

Política do IAM

Para usar o CloudWatch Application Insights, é necessário criar uma política do AWS Identity and Access Management (IAM) e anexá-la ao usuário, grupo ou perfil. Para obter mais informações sobre usuários, grupos e perfis, consulte Identidades do IAM (usuários, grupos e perfis). A política do IAM define as permissões do usuário.

Para criar uma política do IAM usando o console

Para criar uma política do IAM usando o console do IAM, execute as seguintes etapas.

1. Acesse o console do IAM. No painel de navegação à esquerda, selecione Policies (Políticas).
2. Na parte superior da página, selecione Create policy (Criar política).

3. Selecione a guia JSON.
4. Copie e cole o documento JSON a seguir na guia JSON.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "applicationinsights:*",
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "resource-groups:ListGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

5. Selecione Review Policy (Revisar política).
6. Insira um Name (Nome) para a política, por exemplo, "AppInsightsPolicy". Opcionalmente, insira uma Description (Descrição).
7. Selecione Create Policy (Criar política).
8. No painel de navegação esquerdo, escolha Grupos de usuários, Usuários ou Perfis.
9. Selecione o nome do grupo de usuários, usuário ou perfil ao qual deseja anexar a política.
10. Selecione Add permissions (Adicionar permissões).
11. Selecione Attach existing policies directly (Anexar políticas existentes diretamente).
12. Procure a política que acabou de criar e marque a caixa de seleção à esquerda do nome da política.
13. Selecione Next: Review (Próximo: revisar).
14. Certifique-se de que a política correta está listada e selecione Add permissions (Adicionar permissões).
15. Faça login com o usuário associado à política que você acabou de criar ao usar o CloudWatch Application Insights.

Para criar uma política do IAM usando a AWS CLI

Para criar uma política do IAM usando a AWS CLI, execute a operação create-policy na linha de comando utilizando o documento JSON acima como um arquivo na pasta atual.

Para criar uma política do IAM usando o AWS Tools for Windows PowerShell

Para criar uma política do IAM usando o AWS Tools for Windows PowerShell, execute o cmdlt New-IAMPolicy utilizando documento JSON acima como um arquivo na pasta atual.

Permissões de função do IAM para integração de aplicações baseadas em conta

Se você quiser integrar todos os recursos em sua conta e optar por não usar a Política de gerenciamento do Application Insights para acesso total à funcionalidade do Application Insights, você deve anexar as seguintes permissões à sua função do IAM para que o Application Insights possa descobrir todos os recursos em sua conta:

```
"ec2:DescribeInstances"
"ec2:DescribeNatGateways"
"ec2:DescribeVolumes"
"ec2:DescribeVPCs"
"rds:DescribeDBInstances"
"rds:DescribeDBClusters"
"sqs:ListQueues"
"elasticloadbalancing:DescribeLoadBalancers"
"autoscaling:DescribeAutoScalingGroups"
"lambda:ListFunctions"
"dynamodb:ListTables"
"s3:ListAllMyBuckets"
"sns:ListTopics"
"states:ListStateMachines"
"apigateway:GET"
"ecs:ListClusters"
"ecs:DescribeTaskDefinition"
"ecs:ListServices"
"ecs:ListTasks"
"eks:ListClusters"
"eks:ListNodegroups"
"fsx:DescribeFileSystems"
"route53:ListHealthChecks"
"route53:ListHostedZones"
"route53:ListQueryLoggingConfigs"
"route53resolver:ListFirewallRuleGroups"
"route53resolver:ListFirewallRuleGroupAssociations"
"route53resolver:ListResolverEndpoints"
```

```
"route53resolver:ListResolverQueryLogConfigs"
"route53resolver:ListResolverQueryLogConfigAssociations"
"logs:DescribeLogGroups"
"resource-explorer:ListResources"
```

Instalar, configurar e gerenciar sua aplicação para monitoramento

Esta seção fornece as etapas para instalar, configurar e gerenciar a aplicação do CloudWatch Application Insights usando o console, a AWS CLI e o AWS Tools for Windows PowerShell.

Tópicos

Instalar, configurar e gerenciar sua aplicação para monitoramento no console do CloudWatch

Esta seção fornece as etapas para instalar, configurar e gerenciar a aplicação para monitoramento no console do CloudWatch.

Procedimentos do console

Adicionar e configurar uma aplicação

Adicionar e configurar uma aplicação no console do CloudWatch

Para começar a usar o CloudWatch Application Insights no console do CloudWatch, siga estas etapas.

1. Início. Abra a página inicial do console do CloudWatch. No painel de navegação esquerdo, em Insights, escolha Application Insights. Essa página exibe a lista de aplicações que são monitoradas pelo Application Insights do CloudWatch, além dos status de monitoramento.

2. Adicione uma aplicação. Para configurar o monitoramento para sua aplicação, escolha Add an application (Adicionar uma aplicação). Ao escolher Add an application (Adicionar uma aplicação), você será direcionado para Choose Application Type (Escolher o tipo de aplicação).

   - Aplicação baseada em grupo de recursos. Ao selecionar essa opção, você pode escolher quais grupos de recursos monitorar nessa conta. Para usar várias aplicações em um componente, você deve usar o monitoramento baseado em grupos de recursos.
   - Aplicação baseada em conta. Ao selecionar essa opção, você poderá monitorar todos os recursos dessa conta. Se você quiser monitorar todos os recursos em uma conta, recomendamos usar esta opção em vez da a opção baseada em grupo de recursos, pois o processo de integração de aplicações é mais rápido.

   > **Note**
   >
   > Não é possível combinar o monitoramento baseado em grupo de recursos com o monitoramento baseado em conta usando o Application Insights. Para alterar o tipo de aplicação, você deve excluir todas aquelas que estão sendo monitoradas e Choose Application Type (Escolher o tipo de aplicação).

   Quando você adiciona sua primeira aplicação para monitoramento, o CloudWatch Application Insights cria uma função vinculada ao serviço em sua conta, o que dá permissões ao Application Insights para chamar outros serviços da AWS em seu nome. Para obter mais informações sobre a função vinculada ao serviço criada em sua conta pelo Application Insights, consulte Usar funções vinculadas ao serviço do CloudWatch Application Insights.

3. Resource-based application monitoring

   1. Selecione o grupo de recursos.Na página Specify application details (Especificar os detalhes da aplicação), selecione na lista suspensa o grupo de recursos da AWS que contém os recursos da sua aplicação. Esses recursos incluem servidores front-end, load balancers, grupos de Auto Scaling e servidores de banco de dados.

      Se você não criou um grupo de recursos para sua aplicação, pode fazer isso escolhendo Create new resource group (Criar grupo de recursos). Para obter mais informações sobre a criação de grupos de recursos, consulte Guia do usuário do AWS Resource Groups.

   2. Monitorar CloudWatch Events. Marque a caixa de seleção para integrar o monitoramento do Application Insights com o CloudWatch Events para obter insights do Amazon EBS,

Amazon EC2,AWS CodeDeploy, Amazon ECS, APIs e notificações do AWS Health, Amazon RDS, Amazon S3 eAWS Step Functions.

3. Integração ao Systems Manager OpsCenter da AWS. Para visualizar e receber notificações quando problemas forem detectados em aplicações selecionadas, selecione a caixa de seleção Generate Systems Manager OpsCenter OpsItems for remedial actions (Gerar OpsItems do Systems Manager OpsCenter para ações corretivas). Para rastrear as operações executadas para resolver itens de trabalho operacionais (OpsItems) relacionados aos recursos da AWS, forneça o ARN do tópico do SNS.
4. Etiquetas (opcionais). O CloudWatch Application Insights oferece suporte a grupos de recurso baseados em etiquetas e baseados no CloudFormation (com exceção dos grupos do Auto Scaling). Para obter mais informações, consulte Trabalhar com o Tag Editor.
5. Escolha Próximo.

   Um ARN será gerado para a aplicação no formato a seguir.

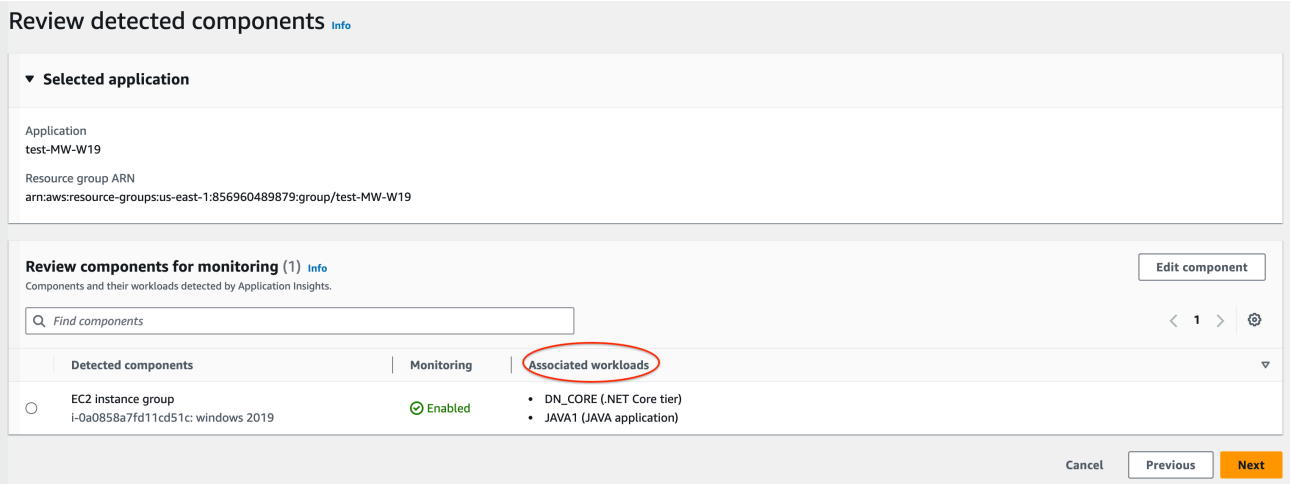
   ```
   arn:partition:applicationinsights:region:account-id:application/resource-group/resource-group-name
   ```

   Exemplo

   ```
   arn:aws:applicationinsights:us-east-1:123456789012:application/resource-group/my-resource-group
   ```

6. Na página Revisar componentes detectados, em Revisar componentes para monitoramento, a tabela lista os componentes detectados e suas workloads detectadas associadas.

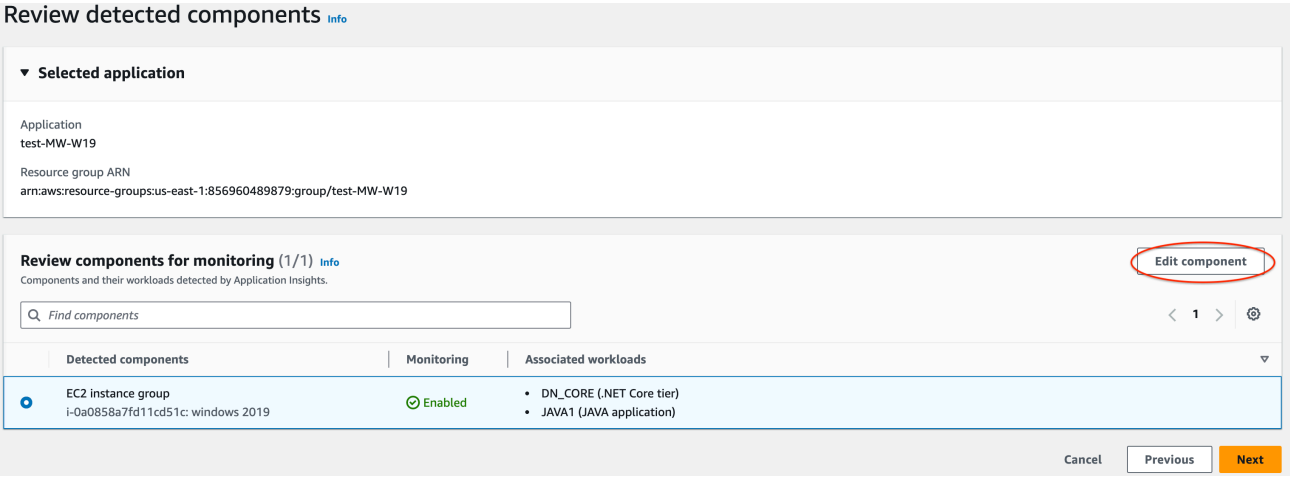
   > **Note**
   >
   > Para componentes que ofereçam suporte a várias workloads personalizadas, é possível monitorar até cinco workloads para cada componente. Essas workloads serão monitoradas separadamente do componente.



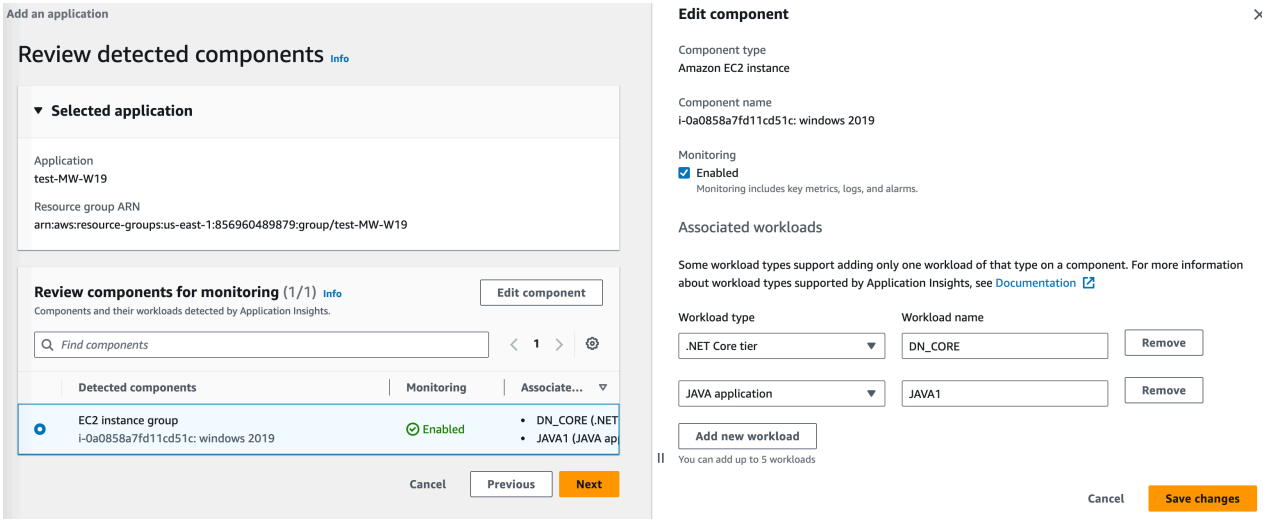
Em Workloads associadas, há várias mensagens possíveis que aparecem se uma workload não estiver listada.

- Não foi possível detectar workloads: ocorreu um problema ao tentar detectar workloads. Certifique-se de ter concluído Pré-requisitos. Se você precisar adicionar workloads, escolha Editar componente.
- Nenhuma workload trabalho detectada: não detectamos nenhuma workload. Talvez seja necessário adicionar workloads. Para fazer isso, escolha Editar componente.
- Não aplicável: o componente não oferece suporte a workloads personalizadas e será monitorado com métricas, alarmes e logs padrão. Você não pode adicionar workloads a esses componentes.

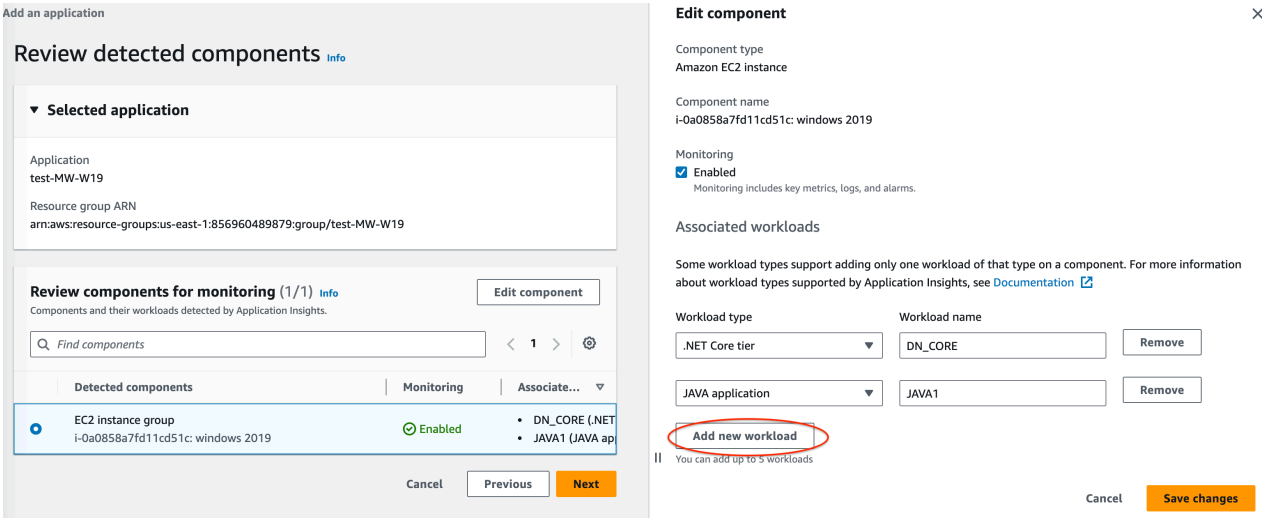
7. Para editar um componente, selecione um componente e, em seguida, escolha Editar componente. Um painel lateral se abrirá com as workloads detectadas no componente. Nesse painel, é possível editar os detalhes do componente e adicionar novas workloads.



- Para editar o tipo ou nome da workload, use a lista suspensa.

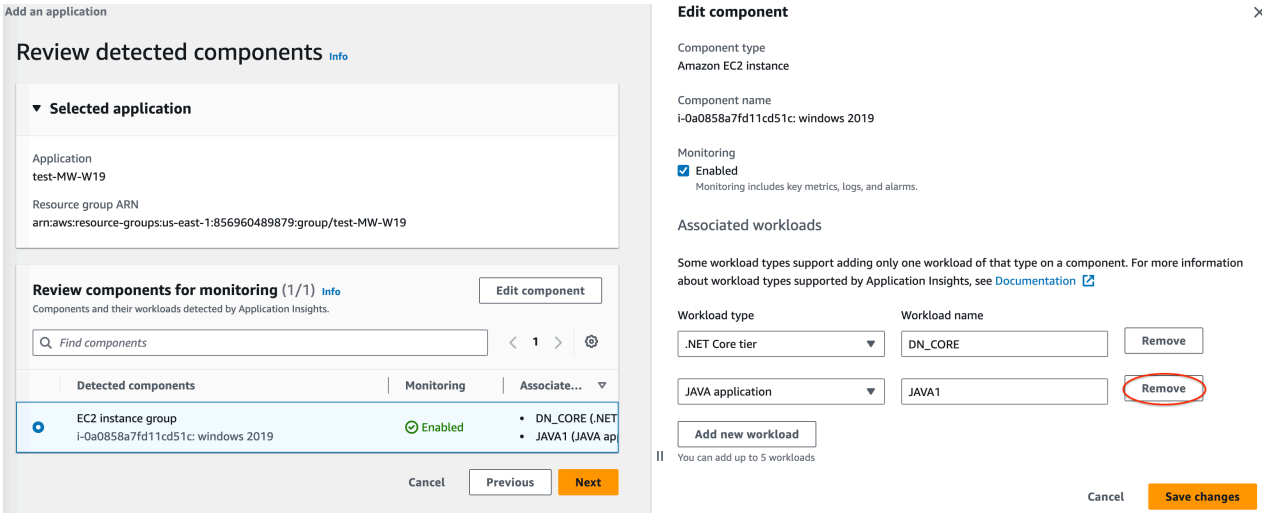


- Para adicionar uma workload ao componenyte, escolha Adicionar nova workload.

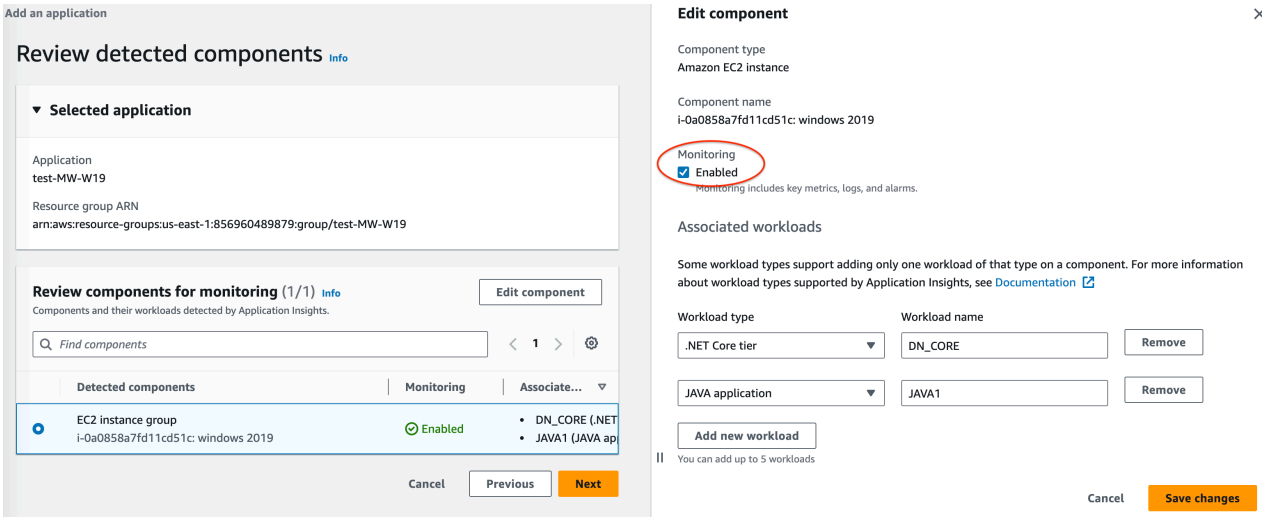


  - Se Adicionar nova workload não aparecer, esse componente não oferece suporte a várias workloads.
  - Se o título Workloads associadas não aparecer, esse componente não oferece suporte a workloads personalizadas.

- Para remover uma workload, escolha Remover ao lado da workload que você deseja remover do monitoramento.



- Para desativar o monitoramento de todo o componente, desmarque a caixa de seleção Monitoramento.



- Quando terminar de editar o componente, escolha Salvar alterações no canto inferior direito. Todas as alterações nas workloads de um componente serão visíveis na tabela Revisar componentes para monitoramento, em Workloads associadas.

8. Na página Revisar componentes detectados, escolha Avançar.
9. A página Especificar detalhes do componente inclui todos os componentes com workloads associadas personalizáveis da etapa anterior.

> **Note**
>
> Se um cabeçalho de um componente tiver uma tag opcional, detalhes adicionais das workloads nesse componente serão opcionais.

Se um componente não aparecer nessa página, o componente não terá detalhes adicionais que possam ser especificados nesta etapa.

10. Escolha Próximo.
11. Na página Revisar e enviar, revise todos os detalhes do componente monitorado e da workload.
12. Selecione Enviar.

Account-based application monitoring

1. Application name (Nome da aplicação). Insira um nome para sua aplicação baseada em conta.
2. Monitoramento automatizado de novos recursos. Por padrão, o Application Insights usa as configurações recomendadas para configurar o monitoramento de componentes de recursos que são adicionados à sua conta após a integração da aplicação. Você pode excluir o monitoramento de recursos adicionados após a integração da aplicação desmarcando a caixa de seleção.
3. Monitorar CloudWatch Events. Marque a caixa de seleção para integrar o monitoramento do Application Insights com o CloudWatch Events para obter insights do Amazon EBS, Amazon EC2,AWS CodeDeploy, Amazon ECS, APIs e notificações do AWS Health, Amazon RDS, Amazon S3 eAWS Step Functions.
4. Integração ao Systems Manager OpsCenter da AWS. Para visualizar e receber notificações quando problemas forem detectados em aplicações selecionadas, selecione a caixa de seleção Generate Systems Manager OpsCenter OpsItems for remedial actions (Gerar OpsItems do Systems Manager OpsCenter para ações corretivas). Para rastrear as operações executadas para resolver itens de trabalho operacionais (OpsItems) relacionados aos recursos da AWS, forneça o ARN do tópico do SNS.
5. Etiquetas (opcionais). O CloudWatch Application Insights oferece suporte a grupos de recurso baseados em etiquetas e baseados no CloudFormation (com exceção dos grupos do Auto Scaling). Para obter mais informações, consulte Trabalhar com o Tag Editor.
6. Recursos descobertos. Todos os recursos descobertos em sua conta são adicionados a esta lista. Se o Application Insights não conseguir descobrir todos os recursos da conta, uma mensagem de erro será exibida na parte superior da página. Esta mensagem inclui um link para a documentação sobre como adicionar as permissões necessárias.

7. Escolha Próximo.

   Um ARN será gerado para a aplicação no formato a seguir.

   ```
   arn:partition:applicationinsights:region:account-id:application/
   TBD/application-name
   ```

   Exemplo

   ```
   arn:aws:applicationinsights:us-east-1:123456789012:application/TBD/my-
   application
   ```

4. Depois de enviar a configuração de monitoramento da aplicação, você será direcionado para a página de detalhes da aplicação, onde poderá visualizar o Application summary (Resumo da aplicação), a lista Monitored components (Componentes monitorados) e Unmonitored components (Componentes não monitorados). Além disso, selecionando as guias ao lado de Components (Componentes), você verá o Configuration history (Histórico de configuração), os Log patterns (Padrões de log) e qualquer Tag (Etiqueta) que tenha aplicado.

   Para visualizar insights da aplicação, escolha View Insights (Visualizar Insights).

   É possível atualizar suas seleções para monitoramento e integração do CloudWatch Events ao AWS Systems Manager OpsCenter escolhendo Edit (Editar).

   Em Components (Componentes), é possível selecionar o menu Actions (Ações) para criar, modificar ou desagrupar um grupo de instâncias.

   É possível gerenciar o monitoramento de componentes, incluindo camada de aplicação, grupos de log, logs de eventos, métricas e alarmes personalizados. Para isso, selecione o marcador ao lado de um componente e escolha Manage monitoring (Gerenciar o monitoramento).

Habilitar o Application Insights para monitoramento de recursos do Amazon ECS e do Amazon EKS

Você pode habilitar o Application Insights para monitorar aplicações e microsserviços em contêiner no console do Container Insights. O Application Insights oferece suporte ao monitoramento dos seguintes recursos:

- Clusters do Amazon ECS
- Serviços do Amazon ECS

- Tarefas do Amazon ECS
- Clusters do Amazon EKS

Quando o Application Insights está ativado, ele fornece métricas e registros recomendados, detecta possíveis problemas, gera CloudWatch Events e cria painéis automáticos para suas aplicações e microsserviços em contêiner.

Você pode habilitar o Application Insights para recursos em contêiner nos consoles Container Insights ou Application Insights.

Habilitar insights de aplicações no console Container Insights

No console do Container Insights, no painel Performance Monitoring (Monitoramento da performance) do Container Insights, escolha Auto-configure Application Insights (Configurar automaticamente o Application Insights). Quando o Application Insights está habilitado, ele exibe detalhes sobre problemas detectados.

Habilitar o Application Insights a partir do console

Quando clusters do ECS aparecem na lista de componentes, o Application Insights permite automaticamente o monitoramento adicional de contêiner com o Container Insights

Em clusters do EKS, é possível habilitar o monitoramento adicional com o Container Insights para fornecer informações de diagnóstico, como falhas de reinicialização de contêiner, para ajudar você a isolar e resolver problemas. São necessárias etapas adicionais para configurar o Container Insights para o EKS. Consulte Configurar o Container Insights no Amazon EKS e no Kubernetes para obter mais informações sobre as etapas do Container Insights no EKS.

O monitoramento adicional para EKS com o Container Insights é compatível com instâncias Linux com EKS.

Para obter mais informações sobre a compatibilidade do Container Insights com clusters ECS e EKS, consulte Container Insights.

Desabilitar o monitoramento de um componente da aplicação

Para desabilitar o monitoramento de um componente da aplicação, na página de detalhes da aplicação, selecione o componente para o qual deseja desabilitar o monitoramento. Escolha Actions (Ações) e, em seguida, Remove from monitoring (Remover do monitoramento).

Excluir uma aplicação

Para excluir uma aplicação, mo painel do CloudWatch, no painel de navegação esquerdo, escolha Application Insights em Insights. Selecione a aplicação que você deseja excluir. Em Actions (Ações), escolha Delete application (Excluir aplicação). Isso exclui o monitoramento e exclui todos os monitores salvos para os componentes da aplicação. Os recursos da aplicação não são excluídos.

Instalar, configurar e gerenciar sua aplicação para monitoramento usando a linha de comando

Esta seção apresenta as etapas necessárias para instalar, configurar e gerenciar a aplicação para monitoramento usando a AWS CLI e o AWS Tools for Windows PowerShell.

Procedimentos para a linha de comando

- Adicionar e gerenciar uma aplicação
- Gerenciar e atualizar o monitoramento
- Configurar o monitoramento para grupos de disponibilidade Always On do SQL
- Configurar o monitoramento para MySQL RDS
- Configurar o monitoramento para MySQL EC2
- Configurar o monitoramento para o PostgreSQL RDS
- Configurar o monitoramento para o PostgreSQL EC2
- Configurar o monitoramento para o Oracle RDS
- Configurar o monitoramento para o Oracle EC2

Adicionar e gerenciar uma aplicação

Você pode adicionar, obter informações sobre, gerenciar e configurar sua aplicação do Application Insights usando a linha de comando.

Tópicos

- Adicionar uma aplicação
- Descrever uma aplicação
- Listar os componentes em uma aplicação
- Descrever um componente
- Agrupar recursos semelhantes em um componente personalizado
- Desagrupar um componente personalizado

- Atualizar uma aplicação
- Atualizar um componente personalizado

Adicionar uma aplicação

Adicionar uma aplicação usando a AWS CLI

Para usar a AWS CLI para adicionar uma aplicação ao grupo de recursos chamado `my-resource-group`, com o OpsCenter habilitado para entregar o opsItem criado ao ARN do tópico do SNS `arn:aws:sns:us-east-1:123456789012:MyTopic`, use o comando a seguir.

```
aws application-insights create-application --resource-group-name my-resource-group --
ops-center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Adicionar uma aplicação usando o AWS Tools for Windows PowerShell

Para usar o AWS Tools for Windows PowerShell para adicionar uma aplicação ao grupo de recursos chamado `my-resource-group`, com o OpsCenter habilitado para entregar o opsItem criado ao ARN do tópico do SNS `arn:aws:sns:us-east-1:123456789012:MyTopic`, use o comando a seguir.

```
New-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Descrever uma aplicação

Descrever uma aplicação usando a AWS CLI

Para usar o AWS CLI para descrever uma aplicação criada em um grupo de recursos chamado `my-resource-group`, use o comando a seguir.

```
aws application-insights describe-application --resource-group-name my-resource-group
```

Descrever uma aplicação usando o AWS Tools for Windows PowerShell

Para usar o AWS Tools for Windows PowerShell para descrever uma aplicação criada em um grupo de recursos chamado `my-resource-group`, use o comando a seguir.

```
Get-CWAIApplication -ResourceGroupName my-resource-group
```

Listar os componentes em uma aplicação

Listar os componentes de uma aplicação usando a AWS CLI

Para usar o AWS CLI para listar os componentes criados em um grupo de recursos chamado `my-resource-group`, use o comando a seguir.

```
aws application-insights list-components --resource-group-name my-resource-group
```

Listar os componentes de uma aplicação usando o AWS Tools for Windows PowerShell

Para usar o AWS Tools for Windows PowerShell para listar os componentes criados em um grupo de recursos chamado `my-resource-group`, use o comando a seguir.

```
Get-CWAIComponentList -ResourceGroupName my-resource-group
```

Descrever um componente

Descrever um componente usando a AWS CLI

É possível usar o seguinte comando da AWS CLI para descrever um componente chamado `my-component` que pertence a uma aplicação criada em um grupo de recursos chamado `my-resource-group`.

```
aws application-insights describe-component --resource-group-name my-resource-group --
component-name my-component
```

Descrever um componente usando AWS Tools for Windows PowerShell

É possível usar o seguinte comando da AWS Tools for Windows PowerShell para descrever um componente chamado `my-component` que pertence a uma aplicação criada em um grupo de recursos chamado `my-resource-group`.

```
Get-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

Agrupar recursos semelhantes em um componente personalizado

Recomendamos agrupar recursos semelhantes, como instâncias de servidor web do .NET, em componentes personalizados para integração mais fácil e melhores monitoramento e insights. Atualmente, o CloudWatch Application Insights oferece suporte a grupos personalizados para instâncias do EC2.

Para agrupar recursos em um componente personalizado usando a AWS CLI

Para usar a AWS CLI para agrupar três instâncias (`arn:aws:ec2:us-east-1:123456789012:instance/i-11111`, `arn:aws:ec2:us-east-1:123456789012:instance/i-22222` e `arn:aws:ec2:us-east-1:123456789012:instance/i-33333`) em um componente personalizado chamado `my-component` para uma aplicação criada para o grupo de recursos chamado `my-resource-group`, use o comando a seguir.

```
aws application-insights create-component --resource-group-name my-
resource-group --component-name my-component --resource-list arn:aws:ec2:us-
east-1:123456789012:instance/i-11111 arn:aws:ec2:us-east-1:123456789012:instance/
i-22222 arn:aws:ec2:us-east-1:123456789012:instance/i-33333
```

Para agrupar recursos em um componente personalizado usando AWS Tools for Windows PowerShell

Para usar AWS Tools for Windows PowerShell para agrupar três instâncias (`arn:aws:ec2:us-east-1:123456789012:instance/i-11111`, `arn:aws:ec2:us-east-1:123456789012:instance/i-22222` e `arn:aws:ec2:us-east-1:123456789012:instance/i-33333`) em um componente personalizado chamado `my-component` para uma aplicação criada para o grupo de recursos chamado `my-resource-group`, use o comando a seguir.

```
New-CWAIComponent -ResourceGroupName my-resource-group -ComponentName my-component
 -ResourceList arn:aws:ec2:us-east-1:123456789012:instance/i-11111,arn:aws:ec2:us-
east-1:123456789012:instance/i-22222,arn:aws:ec2:us-east-1:123456789012:instance/
i-33333
```

Desagrupar um componente personalizado

Para desagrupar um componente personalizado usando a AWS CLI

Para usar o AWS CLI para desagrupar um componente personalizado chamado `my-component` em uma aplicação criada no grupo de recursos `my-resource-group`, use o comando a seguir.

```
aws application-insights delete-component --resource-group-name my-resource-group --
component-name my-new-component
```

Para desagrupar um componente personalizado usando o AWS Tools for Windows PowerShell

Para usar o AWS Tools for Windows PowerShell para desagrupar um componente personalizado chamado `my-component` em uma aplicação criada no grupo de recursos `my-resource-group`, use o comando a seguir.

```
Remove-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

Atualizar uma aplicação

Atualizar uma aplicação usando a AWS CLI

É possível usar a AWS CLI para atualizar uma aplicação para gerar OpsItems do AWS Systems Manager OpsCenter para problemas detectados na aplicação e associar os OpsItems criados ao tópico `arn:aws:sns:us-east-1:123456789012:MyTopic` do SNS usando o comando a seguir.

```
aws application-insights update-application --resource-group-name my-resource-group --
ops-center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Atualizar uma aplicação usando o AWS Tools for Windows PowerShell

É possível usar o AWS Tools for Windows PowerShell para atualizar uma aplicação para gerar OpsItems do AWS SSM OpsCenter para problemas detectados na aplicação e associar os OpsItems criados ao tópico `arn:aws:sns:us-east-1:123456789012:MyTopic` do SNS usando o comando a seguir.

```
Update-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Atualizar um componente personalizado

Atualizar um componente personalizado usando a AWS CLI

Você pode usar a AWS CLI para atualizar um componente personalizado chamado `my-component` com um novo nome de componente, `my-new-component`, e um grupo atualizado de instâncias usando o comando a seguir.

```
aws application-insights update-component --resource-group-name my-resource-
group --component-name my-component --new-component-name my-new-component --
resource-list arn:aws:ec2:us-east-1:123456789012:instance/i-44444 arn:aws:ec2:us-
east-1:123456789012:instance/i-55555
```

Atualizar um componente personalizado usando o AWS Tools for Windows PowerShell

Você pode usar a AWS Tools for Windows PowerShell para atualizar um componente personalizado chamado `my-component` com um novo nome de componente, `my-new-component`, e um grupo atualizado de instâncias usando o comando a seguir.

```
Update-CWAIComponent -ComponentName my-component -NewComponentName my-new-
component -ResourceGroupName my-resource-group -ResourceList arn:aws:ec2:us-
east-1:123456789012:instance/i-44444,arn:aws:ec2:us-east-1:123456789012:instance/
i-55555
```

Gerenciar e atualizar o monitoramento

Você pode gerenciar e atualizar o monitoramento da aplicação Application Insights usando a linha de comando.

Tópicos

- Listar problemas com sua aplicação
- Descrever um problema de aplicação
- Descrever as anomalias ou os erros associados a um problema
- Descrever uma anomalia ou um erro com a aplicação
- Descrever as configurações de monitoramento de um componente
- Descrever a configuração de monitoramento recomendada de um componente
- Atualizar as configurações de monitoramento de um componente
- Remover um grupo de recursos especificado do monitoramento do Application Insights

Listar problemas com sua aplicação

Listar problemas com sua aplicação usando a AWS CLI

Para usar a AWS CLI para listar problemas em sua aplicação detectados entre 1.000 e 10.000 milissegundos desde a época Unix para uma aplicação criada em um grupo de recursos chamado `my-resource-group`, use o comando a seguir.

```
aws application-insights list-problems --resource-group-name my-resource-group --start-
time 1000 --end-time 10000
```

Listar problemas com sua aplicação usando o AWS Tools for Windows PowerShell

Para usar a AWS Tools for Windows PowerShell para listar problemas em sua aplicação detectados entre 1.000 e 10.000 milissegundos desde a época Unix para uma aplicação criada em um grupo de recursos chamado `my-resource-group`, use o comando a seguir.

```
$startDate = "8/6/2019 3:33:00"
$endDate = "8/6/2019 3:34:00"
Get-CWAIProblemList -ResourceGroupName my-resource-group -StartTime $startDate -
EndTime $endDate
```

Descrever um problema de aplicação

Descrever um problema de aplicação usando a AWS CLI

Para usar a AWS CLI para descrever uma falha com o ID do problema `p-1234567890`, use o comando a seguir.

```
aws application-insights describe-problem –problem-id p-1234567890
```

Descrever um problema de aplicação usando o AWS Tools for Windows PowerShell

Para usar a AWS Tools for Windows PowerShell para descrever uma falha com o ID do problema `p-1234567890`, use o comando a seguir.

```
Get-CWAIProblem -ProblemId p-1234567890
```

Descrever as anomalias ou os erros associados a um problema

Descrever as anomalias ou os erros associados a um problema usando a AWS CLI

Para usar a AWS CLI para descrever as anomalias ou os erros associados a uma falha com o ID do problema `p-1234567890`, use o comando a seguir.

```
aws application-insights describe-problem-observations --problem-id -1234567890
```

Descrever as anomalias ou os erros associados a um problema usando o AWS Tools for Windows PowerShell

Para usar a AWS Tools for Windows PowerShell para descrever as anomalias ou os erros associados a uma falha com o ID do problema `p-1234567890`, use o comando a seguir.

```
Get-CWAIProblemObservation -ProblemId p-1234567890
```

Descrever uma anomalia ou um erro com a aplicação

Descrever uma anomalia ou um erro na aplicação usando a AWS CLI

Para usar a AWS CLI para descrever uma anomalia ou um erro na aplicação com o ID de observação `o-1234567890`, use o comando a seguir.

```
aws application-insights describe-observation –observation-id o-1234567890
```

Descrever uma anomalia ou um erro na aplicação usando o AWS Tools for Windows PowerShell

Para usar a AWS Tools for Windows PowerShell para descrever uma anomalia ou um erro na aplicação com o ID de observação `o-1234567890`, use o comando a seguir.

```
Get-CWAIObservation -ObservationId o-1234567890
```

Descrever as configurações de monitoramento de um componente

Descrever as configurações de monitoramento de um componente usando a AWS CLI

Para usar a AWS CLI para descrever a configuração do monitoramento de um componente chamado `my-component` em uma aplicação criada no grupo de recursos `my-resource-group`, use o comando a seguir.

```
aws application-insights describe-component-configuration –resource-group-name my-
resource-group –component-name my-component
```

Descrever as configurações de monitoramento de um componente usando o AWS Tools for Windows PowerShell

Para usar o AWS Tools for Windows PowerShell para descrever a configuração do monitoramento de um componente chamado `my-component` em uma aplicação criada no grupo de recursos `my-resource-group`, use o comando a seguir.

```
Get-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-
resource-group
```

Para obter mais informações sobre a configuração do componente e, por exemplo, arquivos JSON, consulte Trabalhar com configurações de componentes.

Descrever a configuração de monitoramento recomendada de um componente

Descrever a configuração de monitoramento recomendada de um componente usando a AWS CLI

Quando o componente faz parte de uma aplicação .NET Worker, é possível usar a AWS CLI para descrever a configuração de monitoramento recomendada de um componente chamado `my-component` em uma aplicação criada no grupo de recursos `my-resource-group` usando o comando a seguir.

```
aws application-insights describe-component-configuration-recommendation --resource-
group-name my-resource-group --component-name my-component --tier DOT_NET_WORKER
```

Descrever a configuração de monitoramento recomendada de um componente usando o AWS Tools for Windows PowerShell

Quando o componente faz parte de uma aplicação .NET Worker, é possível usar a AWS Tools for Windows PowerShell para descrever a configuração de monitoramento recomendada de um componente chamado `my-component` em uma aplicação criada no grupo de recursos `my-resource-group` usando o comando a seguir.

```
Get-CWAIComponentConfigurationRecommendation -ComponentName my-component -
ResourceGroupName my-resource-group -Tier DOT_NET_WORKER
```

Para obter mais informações sobre a configuração do componente e, por exemplo, arquivos JSON, consulte Trabalhar com configurações de componentes.

Atualizar as configurações de monitoramento de um componente

Atualizar as configurações de monitoramento de um componente usando a AWS CLI

Para usar a AWS CLI para atualizar o componente chamado `my-component` em uma aplicação criada no grupo de recursos chamado `my-resource-group`, use o comando a seguir. O comando inclui estas ações:

1. Habilitar o monitoramento do componente.
2. Definir o nível do componente como `.NET Worker`.
3. Atualizar a configuração JSON do componente para ler o arquivo local `configuration.txt`.

```
aws application-insights update-component-configuration --resource-group-name my-
resource-group --component-name my-component --tier DOT_NET_WORKER --monitor --
component-configuration "file://configuration.txt"
```

Atualizar as configurações de monitoramento de um componente usando a AWS Tools for Windows PowerShell

Para usar a AWS Tools for Windows PowerShell para atualizar o componente chamado `my-component` em uma aplicação criada no grupo de recursos chamado `my-resource-group`, use o comando a seguir. O comando inclui estas ações:

1. Habilitar o monitoramento do componente.
2. Definir o nível do componente como `.NET Worker`.
3. Atualizar a configuração JSON do componente para ler o arquivo local `configuration.txt`.

```
[string]$config = Get-Content -Path configuration.txt
Update-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-
resource-group -Tier DOT_NET_WORKER -Monitor 1 -ComponentConfiguration $config
```

Para obter mais informações sobre a configuração do componente e, por exemplo, arquivos JSON, consulte Trabalhar com configurações de componentes.

Remover um grupo de recursos especificado do monitoramento do Application Insights

Remover um grupo de recursos especificado do monitoramento do Application Insights usando a AWS CLI

Para usar a AWS CLI para remover uma aplicação criada no grupo de recursos chamado `my-resource-group` do monitoramento, use o comando a seguir.

```
aws application-insights delete-application --resource-group-name my-resource-group
```

Remover um grupo de recursos especificado do monitoramento do Application Insights usando a AWS Tools for Windows PowerShell

Para usar a AWS Tools for Windows PowerShell para remover uma aplicação criada no grupo de recursos chamado `my-resource-group` do monitoramento, use o comando a seguir.

```
Remove-CWAIApplication -ResourceGroupName my-resource-group
```

Configurar o monitoramento para grupos de disponibilidade Always On do SQL

1. Crie uma aplicação para o grupo de recursos com as instâncias do SQL HA EC2.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Defina as instâncias do EC2 que representam o cluster de HA do SQL criando um componente de aplicação.

```
aws application-insights create-component --resource-group-name
 "<RESOURCE_GROUP_NAME>" --component-name SQL_HA_CLUSTER --resource-list
  "arn:aws:ec2:<REGION>:<ACCOUNT_ID>:instance/<CLUSTER_INSTANCE_1_ID>"
 "arn:aws:ec2:<REGION>:<ACCOUNT_ID>:instance/<CLUSTER_INSTANCE_2_ID>
```

3. Configure o componente de HA do SQL.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "SQL_HA_CLUSTER" --
monitor --tier SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP --monitor  --component-
configuration '{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
```

```
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Buffer cache hit ratio",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Page life expectancy",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics Processes blocked",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics User Connections",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Locks Number of Deadlocks/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:SQL Statistics Batch Requests/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica File Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log remaining for undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Send Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Mirrored Write Transaction/
sec",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Recovery Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redo Bytes Remaining",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redone Bytes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Total Log requiring undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Transaction Delay",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP-
<RESOURCE_GROUP_NAME>",
      "logPath" : "C:\\Program Files\\Microsoft SQL Server\\MSSQL**.MSSQLSERVER\
\MSSQL\\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
```

```
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}'
```

> **Note**
>
> O Application Insights deve ingerir logs de eventos da aplicação (nível de informações) para detectar atividades do cluster, como failover.

Configurar o monitoramento para MySQL RDS

1. Crie uma aplicação para o grupo de recursos com a instância de banco de dados MySQL do RDS.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. O log de erros está habilitado por padrão. O log de consulta lento pode ser habilitado usando grupos de parâmetros de dados. Para obter mais informações, consulte Acessar os logs gerais e de consultas lentas do MySQL.

- `set slow_query_log = 1`
- `set log_output = FILE`

3. Exporte os logs a serem monitorados para os logs do CloudWatch. Para obter mais informações, consulte Publicar logs MySQL no CloudWatch Logs.
4. Configure o componente MySQL RDS.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "<DB_COMPONENT_NAME>"
 --monitor --tier DEFAULT --monitor  --component-configuration "{\"alarmMetrics\":
[{\"alarmMetricName\":\"CPUUtilization\",\"monitor\":true}],\"logs\":[{\"logType\":
\"MYSQL\",\"monitor\":true},{\"logType\": \"MYSQL_SLOW_QUERY\",\"monitor\":false}]}"
```

Configurar o monitoramento para MySQL EC2

1. Crie uma aplicação para o grupo de recursos com as instâncias do SQL HA EC2.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. O log de erros está habilitado por padrão. O log de consulta lento pode ser habilitado usando grupos de parâmetros de dados. Para obter mais informações, consulte Acessar os logs gerais e de consultas lentas do MySQL.
   - `set slow_query_log = 1`
   - `set log_output = FILE`
3. Configure o componente MySQL EC2.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "<DB_COMPONENT_NAME>"
 --monitor --tier MYSQL --monitor  --component-configuration "{\"alarmMetrics\":
[{\"alarmMetricName\":\"CPUUtilization\",\"monitor\":true}],\"logs\":[{\"logGroupName
\":\"<UNIQUE_LOG_GROUP_NAME>\",\"logPath\":\"C:\\\\ProgramData\\\\MySQL\\\\MySQL
 Server **\\\\Data\\\\<FILE_NAME>.err\",\"logType\":\"MYSQL\",\"monitor\":true,
\"encoding\":\"utf-8\"}]}"
```

Configurar o monitoramento para o PostgreSQL RDS

1. Crie uma aplicação para o grupo de recursos com a instância de banco de dados PostgreSQL RDS.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. A publicação de logs do PostgreSQL no CloudWatch não é habilitada por padrão. Para habilitar o monitoramento, abra o console do RDS e selecione o banco de dados a ser monitorado. Escolha Modify (Modificar) no canto superior direito e marque a caixa de seleção rotulada com o log do PostgreSQL. Selecione Continue (Continuar) para salvar essa configuração.
3. Os logs do PostgreSQL são exportados para o CloudWatch.
4. Configure o componente PostgreSQL RDS .

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier DEFAULT --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\": \"CPUUtilization\",
         \"monitor\": true
      }
   ],
   \"logs\":[
      {
         \"logType\": \"POSTGRESQL\",
         \"monitor\": true
      }
   ]
}"
```

Configurar o monitoramento para o PostgreSQL EC2

1. Crie uma aplicação para o grupo de recursos com a instância do PostgreSQL EC2.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Configure o componente PostgreSQL EC2.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier POSTGRESQL --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\":\"CPUUtilization\",
         \"monitor\":true
      }
   ],
   \"logs\":[
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/var/lib/pgsql/data/log/\",
         \"logType\":\"POSTGRESQL\",
         \"monitor\":true,
         \"encoding\":\"utf-8\"
      }
   ]
}"
```

Configurar o monitoramento para o Oracle RDS

1. Crie uma aplicação para o grupo de recursos com a instância de banco de dados Oracle RDS.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. A publicação de logs do Oracle no CloudWatch não é habilitada por padrão. Para habilitar o monitoramento, abra o console do RDS e selecione o banco de dados a ser monitorado. Escolha Modify (Modificar) no canto superior direito e marque as caixas de seleção rotuladas com o log Alert e Listener. Selecione Continue (Continuar) para salvar essa configuração.
3. Os logs Oracle são exportados para o CloudWatch.
4. Configure o componente Oracle RDS.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier DEFAULT --component-configuration
```

```
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\": \"CPUUtilization\",
         \"monitor\": true
      }
   ],
   \"logs\":[
      {
         \"logType\": \"ORACLE_ALERT\",
         \"monitor\": true
      },
      {
         \"logType\": \"ORACLE_LISTENER\",
         \"monitor\": true
      }
   ]
}"
```

Configurar o monitoramento para o Oracle EC2

1. Crie uma aplicação para o grupo de recursos com a instância Oracle EC2.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Configure o componente Oracle EC2.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier ORACLE --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\":\"CPUUtilization\",
         \"monitor\":true
      }
   ],
   \"logs\":[
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/opt/oracle/diag/rdbms/*/*/trace\",
```

```
         \"logType\":\"ORACLE_ALERT\",
         \"monitor\":true,
      },
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/opt/oracle/diag/tnslsnr/$HOSTNAME/listener/trace/\",
         \"logType\":\"ORACLE_ALERT\",
         \"monitor\":true,
      }
   ]
}"
```

CloudWatch Events do Application Insights e notificações de problemas detectados

Para cada aplicação que é adicionada ao Application Insights, um evento do CloudWatch é publicado para os seguintes eventos, com base no melhor esforço:

- Criação do problema. Emitido quando o CloudWatch Application Insights detecta um novo problema.
  - Tipo de detalhe: "Application Insights Problem Detected" (Detectado problema do Application Insights)
  - Detalhe:
    - `problemId`: o ID do problema detectado.
    - `region`: a região da AWS onde o problema foi criado.
    - `resourceGroupName`: o grupo de recursos da aplicação registrado para o qual o problema foi detectado.
    - `status`: o status do problema. O status e as definições possíveis são os seguintes:
      - `In progress`: um novo problema foi identificado. O problema ainda está recebendo observações.
      - `Recovering`: o problema está se estabilizando. Você pode resolver manualmente o problema quando ele está nesse estado.
      - `Resolved`: o problema está resolvido. Não há novas observações sobre esse problema.
      - `Recurring`: o problema foi resolvido nas últimas 24 horas. Ele foi reaberto devido a observações adicionais.
    - `severity`: a gravidade do problema.
    - `problemUrl`: o URL do console para o problema.

- Atualização do problema. Emitido quando o problema é atualizado com uma nova observação ou quando uma observação existente é atualizada e o problema é atualizado subsequentemente. As atualizações incluem uma resolução ou o encerramento do problema.
  - Tipo de detalhe: "Application Insights Problem Updated" (Problema do Application Insights atualizado)
  - Detalhe:
    - `problemId`: o ID do problema criado.
    - `region`: a região da AWS onde o problema foi criado.
    - `resourceGroupName`: o grupo de recursos da aplicação registrado para o qual o problema foi detectado.
    - `status`: o status do problema.
    - `severity`: a gravidade do problema.
    - `problemUrl`: o URL do console para o problema.

Como receber notificações para eventos de problema gerados por uma aplicação

No console do CloudWatch, selecione Regras em Eventos, no painel de navegação esquerdo. Na página Regras , selecione Criar regra. Escolha o Amazon CloudWatch Application Insights na lista suspensa Service Name (Nome do serviço) e escolha o Event Type (Tipo de evento). Depois, escolha Adicionar destino e selecione o destino e os parâmetros, por exemplo, um tópico do SNS ou uma função do Lambda.

Ações por meio do AWS Systems Manager. O CloudWatch Application Insights oferece integração incorporada ao Systems Manager OpsCenter. Se você optar por usar essa integração para sua aplicação, um OpsItem será criado no console do OpsCenter para cada problema detectado com a aplicação. No console do OpsCenter, é possível visualizar informações resumidas sobre o problema detectado pelo CloudWatch Application Insights e escolher um runbook do Systems Manager Automation para executar ações corretivas ou identificar ainda mais processos do Windows que estejam causando problemas de recursos na aplicação.

Observabilidade do Application Insights entre contas

Com a observabilidade entre contas do CloudWatch Application Insights, é possível monitorar e solucionar problemas de aplicações que abranjam várias contas da AWS em uma região.

É possível usar o Amazon CloudWatch Observability Access Manager para configurar uma ou mais de suas contas da AWS como conta de monitoramento. Você fornecerá à conta de monitoramento a capacidade de visualizar dados em sua conta de origem criando um coletor em sua conta de monitoramento. Você usa o coletor para criar um link da sua conta de origem para sua conta de monitoramento. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

Recursos necessários do

Para a funcionalidade adequada de observabilidade entre contas do CloudWatch Application Insights, certifique-se de que os tipos a seguir de telemetria sejam compartilhados por meio do CloudWatch Observability Access Manager.

- Aplicações no CloudWatch Application Insights
- Métricas no Amazon CloudWatch
- Grupos de logs no Amazon CloudWatch Logs
- Rastreamentos no AWS X-Ray

Trabalhar com configurações de componentes

Uma configuração de componente é um arquivo de texto em formato JSON que descreve as definições de configuração do componente. Esta seção fornece um exemplo de fragmento do modelo, descrições de seções de configuração do componente e exemplos de configurações do componente.

Tópicos

- Fragmento do modelo de configuração do componente
- Seções de configuração do componente
- Exemplos de configuração do componente

Fragmento do modelo de configuração do componente

O exemplo a seguir mostra um fragmento de modelo no formato JSON.

```
{
  "alarmMetrics" : [
    list of alarm metrics
```

```
  ],
  "logs" : [
    list of logs
  ],
  "processes" : [
   list of processes
  ],
  "windowsEvents" : [
    list of windows events channels configurations
  ],
  "alarms" : [
    list of CloudWatch alarms
  ],
  "jmxPrometheusExporter": {
    JMX Prometheus Exporter configuration
  },
  "hanaPrometheusExporter": {
      SAP HANA Prometheus Exporter configuration
  },
  "haClusterPrometheusExporter": {
      HA Cluster Prometheus Exporter configuration
  },
  "netWeaverPrometheusExporter": {
      SAP NetWeaver Prometheus Exporter configuration
  },
  "subComponents" : [
    {
      "subComponentType" : "AWS::EC2::Instance" ...
      component nested instances configuration
    },
    {
      "subComponentType" : "AWS::EC2::Volume" ...
      component nested volumes configuration
    }
  ]
}
```

Seções de configuração do componente

Uma configuração do componente contém várias seções principais. As seções de uma configuração de componente podem estar listadas em qualquer ordem.

- alarmMetrics (opcional)

Uma lista de métricas a serem monitoradas para o componente. Todos os tipos de componentes podem ter uma seção alarmMetrics.

- logs (opcional)

  Uma lista de logs a serem monitorados para o componente. Somente as instâncias do EC2 podem ter uma seção logs.

- processos (opcional)

  Uma lista de processos a serem monitorados para o componente. Somente as instâncias do EC2 podem ter uma seção de processos.

- subComponents (opcional)

  Configuração de instância aninhada de subComponent do volume para o componente. Os tipos de componentes a seguir podem ter instâncias aninhadas e uma seção subComponent: ELB, ASG, instâncias do EC2 agrupadas personalizadas e instâncias do EC2.

- alarmes (opcional)

  Uma lista de alarmes a serem monitorados para o componente. Todos os tipos de componentes podem ter uma seção de alarmes.

- windowsEvents (opcional)

  Uma lista de eventos do Windows a serem monitorados para o componente. Somente o Windows em instâncias do EC2 tem uma seção `windowsEvents`.

- JMXPrometheusExporter (opcional)

  Configuração do JMX Protheus Exporter.

- hanaPrometheusExporter (opcional)

  Configuração do SAP HANA Prometheus Exporter.

- haClusterPrometheusExporter (opcional)

  Configuração do HA Cluster Prometheus Exporter.

- netWeaverPrometheusExporter (opcional)

  Configuração do exportador do Prometheus para SAP NetWeaver.

- sapAsePrometheusExporter (opcional)

Configuração do SAP ASE Prometheus Exporter.

O exemplo a seguir mostra a sintaxe para o fragmento da seção subComponents no formato JSON.

```
[
  {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [
      list of alarm metrics
    ],
    "logs" : [
      list of logs
    ],
    "processes": [
      list of processes
    ],
    "windowsEvents" : [
      list of windows events channels configurations
    ]
  },
  {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [
      list of alarm metrics
    ]
  }
]
```

Propriedades da seção de configuração de componentes

Esta seção descreve as propriedades de cada seção de configuração do componente

Seções

- Métrica
- Log
- Processar
- JMX Prometheus Exporter
- HANA Prometheus Exporter
- HA Cluster Prometheus Exporter

- Exportador do Prometheus para NetWeaver
- SAP ASE Prometheus Exporter
- Eventos do Windows
- Alarme

Métrica

Define uma métrica a ser monitorada para o componente.

JSON

```
{
  "alarmMetricName" : "monitoredMetricName",
  "monitor" : true/false
}
```

Properties

- alarmMetricName (obrigatório)

  O nome da métrica a ser monitorada para o componente. Para saber sobre as métricas com suporte pelo Application Insights, consulte Logs e métricas compatíveis com o Amazon CloudWatch Application Insights.
- monitor (opcional)

  Booliano para indicar se a métrica deve ser monitorada. O valor padrão é `true`.

Log

Define um log a ser monitorado para o componente.

JSON

```
{
  "logGroupName" : "logGroupName",
  "logPath" : "logPath",
  "logType" : "logType",
  "encoding" : "encodingType",
  "monitor" : true/false
```

```
}
```

Properties

- logGroupName (obrigatório)

  O nome do grupo de logs do CloudWatch a ser associado ao log monitorado. Para obter as restrições de nome do grupo de logs, consulte CreateLogGroup.

- logPath (necessário para componentes de instância do EC2; não necessário para componentes que não usam o CloudWatch Agent, como o AWS Lambda)

  O caminho dos logs a serem monitorados. O caminho do log deve ser um caminho de arquivo absoluto do sistema Windows. Para obter mais informações, consulte Arquivo de configuração do atendente do CloudWatch: seção Logs.

- logType (obrigatório)

  O tipo de log decide os padrões de log em relação aos quais o Application Insights analisa o log. O tipo de log é selecionado a partir de:

  - SQL_SERVER
  - MYSQL
  - MYSQL_SLOW_QUERY
  - POSTGRESQL
  - ORACLE_ALERT
  - ORACLE_LISTENER
  - IIS
  - APPLICATION
  - WINDOWS_EVENTS
  - WINDOWS_EVENTS_ACTIVE_DIRECTORY
  - WINDOWS_EVENTS_DNS
  - WINDOWS_EVENTS_IIS
  - WINDOWS_EVENTS_SHAREPOINT
  - SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP
  - SQL_SERVER_FAILOVER_CLUSTER_INSTANCE
  - DEFAULT

  - CUSTOM
  - STEP_FUNCTION
  - API_GATEWAY_ACCESS
  - API_GATEWAY_EXECUTION
  - SAP_HANA_LOGS
  - SAP_HANA_TRACE
  - SAP_HANA_HIGH_AVAILABILITY
  - SAP_NETWEAVER_DEV_TRACE_LOGS
  - PACEMAKER_HIGH_AVAILABILITY
- encoding (opcional)

  O tipo de codificação dos logs a serem monitorados. A codificação especificada deve ser incluída na lista de codificações compatíveis com atendentes do CloudWatch. Se não for fornecida, o CloudWatch Application Insights usará o tipo de codificação padrão utf-8, exceto:

  - SQL_SERVER: codificação utf-16
  - IIS: codificação ascii
- monitor (opcional)

  Booliano que indica se os logs devem ser monitorados. O valor padrão é `true`.

Processar

Define um processo a ser monitorado para o componente.

JSON

```
{
  "processName" : "monitoredProcessName",
  "alarmMetrics" : [
      list of alarm metrics
  ]
}
```

Properties

- processName (obrigatório)

O nome do processo a ser monitorado para o componente. O nome do processo não deve conter uma raiz de processo, como `sqlservr` ou `sqlservr.exe`.

- alarmMetrics (obrigatório)

  Uma lista de métricas a serem monitoradas para esse processo. Para visualizar as métricas de processo compatíveis com o CloudWatch Application Insights, consulte Amazon Elastic Compute Cloud (EC2) .

JMX Prometheus Exporter

Define as configurações do JMX Prometheus Exporter.

JSON

```
"JMXPrometheusExporter": {
  "jmxURL" : "JMX URL",
  "hostPort" : "The host and port",
  "prometheusPort" : "Target port to emit Prometheus metrics"
}
```

Properties

- jmxURL (opcional)

  Uma URL completa do JMX para se conectar.

- hostPort (opcional)

  O host e a porta à qual se conectar por meio de JMX remoto. Apenas um de `jmxURL` e `hostPort` pode ser especificado.

- prometheusPort (opcional)

  A porta de destino para a qual enviar métricas do Prometheus. Se não for especificada, será usadaa porta padrão 9404.

HANA Prometheus Exporter

Define as configurações do HANA Prometheus Exporter.

JSON

```
"hanaPrometheusExporter": {
    "hanaSid": "SAP HANA  SID",
    "hanaPort": "HANA database port",
    "hanaSecretName": "HANA secret name",
    "prometheusPort": "Target port to emit Prometheus metrics"
}
```

Properties

- hanaSid

  O ID do sistema SAP (SID) de três caracteres do sistema SAP HANA.

- hanaPort

  A porta de banco de dados HANA pela qual o exportador consultará métricas HANA.

- hanaseCretName

  O segredo do AWS Secrets Manager que armazena credenciais de usuário de monitoramento HANA. O exportador HANA Prometheus usa essas credenciais para se conectar ao banco de dados e consultar métricas HANA.

- prometheusPort (opcional)

  A porta de destino para a qual o Prometheus envia métricas. Se não for especificada, será usada a porta padrão 9668.

HA Cluster Prometheus Exporter

Define as configurações do HA Cluster Prometheus Exporter

JSON

```
"haClusterPrometheusExporter": {
    "prometheusPort": "Target port to emit Prometheus metrics"
}
```

Properties

- prometheusPort (opcional)

A porta de destino para a qual o Prometheus envia métricas. Se não for especificada, será usada a porta padrão 9664.

Exportador do Prometheus para NetWeaver

Define as configurações do exportador Prometheus do NetWeaver.

JSON

```
"netWeaverPrometheusExporter": {
    "sapSid": "SAP NetWeaver  SID",
    "instanceNumbers": [ "Array of instance Numbers of SAP NetWeaver system "],
"prometheusPort": "Target port to emit Prometheus metrics"
}
```

Properties

- sapSid

  O ID do sistema SAP (SID) de três caracteres do sistema SAP NetWeaver.
- instanceNumbers

  Matriz dos números de instância do sistema SAP NetWeaver.

  Exemplo: `"instanceNumbers": [ "00", "01"]`
- prometheusPort (opcional)

  A porta de destino para a qual enviar as métricas do Prometheus. Se não for especificada, a porta padrão 9680 será usada.

SAP ASE Prometheus Exporter

Define as configurações do SAP ASE Prometheus Exporter.

JSON

```
"sapASEPrometheusExporter": {
    "sapAseSid": "SAP ASE SID",
    "sapAsePort": "SAP ASE database port",
```

```
    "sapAseSecretName": "SAP ASE secret name",
    "prometheusPort": "Target port to emit Prometheus metrics",
    "agreeToEnableASEMonitoring": true
}
```

Properties

- sapAseSid

  O ID do sistema SAP (SID) de três caracteres do sistema do SAP ASE.
- sapAsePort

  A porta do banco de dados do SAP ASE pela qual o exportador consultará as métricas do ASE.
- sapAseSecretName

  O segredo do AWS Secrets Manager que armazena as credenciais de usuário de monitoramento do ASE. O SAP ASE Prometheus Exporter usa essas credenciais para se conectar ao banco de dados e consultar as métricas do ASE.
- prometheusPort (opcional)

  A porta de destino para a qual o Prometheus envia métricas. Se não for especificada, será usada a porta padrão 9399. Se houver outro banco de dados ASE que esteja usando a porta padrão, usaremos 9499.

Eventos do Windows

Define os eventos do Windows a serem registrados em log.

JSON

```
{
  "logGroupName" : "logGroupName",
  "eventName" : "eventName",
  "eventLevels" : ["ERROR","WARNING","CRITICAL","INFORMATION","VERBOSE"],
  "monitor" : true/false
}
```

Properties

- logGroupName (obrigatório)

O nome do grupo de logs do CloudWatch a ser associado ao log monitorado. Para obter as restrições de nome do grupo de logs, consulte CreateLogGroup.

- eventName (obrigatório)

  O tipo de eventos do Windows a serem registrados em log. É equivalente ao nome do canal de log de eventos do Windows. Por exemplo, System, Security, CustomEventName etc. Esse campo é necessário para cada tipo de evento do Windows a ser registrado em log.

- eventLevels (obrigatório)

  Os níveis de evento a serem registrados. Você deve especificar cada nível a ser registrado em log. Os valores possíveis incluem `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` e `VERBOSE`. Esse campo é necessário para cada tipo de evento do Windows a ser registrado em log.

- monitor (opcional)

  Booliano que indica se os logs devem ser monitorados. O valor padrão é `true`.

Alarme

Define um alarme do CloudWatch a ser monitorado para o componente.

JSON

```
{
  "alarmName" : "monitoredAlarmName",
  "severity" : HIGH/MEDIUM/LOW
}
```

Properties

- alarmName (obrigatório)

  O nome do alarme do CloudWatch a ser monitorado para o componente.

- gravidade (opcional)

  Indica o grau de interrupção quando o alarme dispara.

Exemplos de configuração do componente

Os exemplos a seguir mostram configurações de componentes no formato JSON para serviços relevantes.

Exemplo de configurações do componente

- Tabela do Amazon DynamoDB
- Amazon EC2 Auto Scaling (ASG)
- Cluster do Amazon EKS
- Instância do Amazon Elastic Compute Cloud (EC2)
- Amazon Elastic Container Service (Amazon ECS)
- Serviços do Amazon ECS
- Tarefas do Amazon ECS
- Amazon Elastic File System (Amazon EFS)
- Amazon FSx
- Amazon Relational Database Service (RDS) Aurora MySQL
- Instância do banco de dados relacional (RDS) da Amazon
- Verificação de integridade do Amazon Route 53
- Zona hospedada do Amazon Route 53
- Endpoint do Amazon Route 53 Resolver
- Configurações de logs de consulta do Amazon Route 53 Resolver
- Bucket do Amazon S3
- Amazon Simple Queue Service (SQS)
- Tópico do Amazon SNS
- Amazon Virtual Private Cloud (Amazon VPC)
- Gateways de conversão de endereços de rede (NAT) da Amazon VPC
- Etapas da API REST do API Gateway
- Elastic Load Balancing da aplicação
- Funções do AWS Lambda
- Grupo de regras do AWS Network Firewall
- associação do grupos de regras AWS Network Firewall
- AWS Step Functions

- Instâncias do Amazon EC2 agrupadas pelo cliente
- Elastic Load Balancing
- Java
- Kubernetes no Amazon EC2
- RDS MariaDB e o RDS MySQL
- RDS Oracle
- RDS PostgreSQL
- SAP ASE no Amazon EC2
- Alta disponibilidade do SAP ASE no Amazon EC2
- SAP HANA no Amazon EC2
- SAP HANA High Availability no Amazon EC2
- SAP NetWeaver no Amazon EC2
- Alta disponibilidade do SAP NetWeaver no Amazon EC2
- Grupos de disponibilidade Always On do SQL
- Instância de cluster de failover do SQL

Tabela do Amazon DynamoDB

O exemplo a seguir mostra uma configuração de componente no formato JSON para a tabela do Amazon DynamoDB.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "SystemErrors",
      "monitor": false
    },
    {
      "alarmMetricName": "UserErrors",
      "monitor": false
    },
    {
      "alarmMetricName": "ConsumedReadCapacityUnits",
      "monitor": false
    },
    {
      "alarmMetricName": "ConsumedWriteCapacityUnits",
```

```
      "monitor": false
    },
    {
      "alarmMetricName": "ReadThrottleEvents",
      "monitor": false
    },
    {
      "alarmMetricName": "WriteThrottleEvents",
      "monitor": false
    },
    {
      "alarmMetricName": "ConditionalCheckFailedRequests",
      "monitor": false
    },
    {
      "alarmMetricName": "TransactionConflict",
      "monitor": false
    }
  ],
  "logs": []
}
```

Amazon EC2 Auto Scaling (ASG)

O exemplo a seguir mostra uma configuração de componente no formato JSON para o Amazon EC2 Auto Scaling (ASG).

```
{
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUCreditBalance"
      }, {
        "alarmMetricName" : "EBSIOBalance%"
      }
    ],
    "subComponents" : [
      {
        "subComponentType" : "AWS::EC2::Instance",
        "alarmMetrics" : [
          {
            "alarmMetricName" : "CPUUtilization"
          }, {
            "alarmMetricName" : "StatusCheckFailed"
```

```
          }
        ],
        "logs" : [
          {
            "logGroupName" : "my_log_group",
            "logPath" : "C:\\LogFolder\\*",
            "logType" : "APPLICATION"
          }
        ],
        "processes" : [
          {
            "processName" : "my_process",
            "alarmMetrics" : [
              {
                  "alarmMetricName" : "procstat cpu_usage",
                  "monitor" : true
              }, {
                  "alarmMetricName" : "procstat memory_rss",
                  "monitor" : true
              }
            ]
      }
  ],
        "windowsEvents" : [
          {
            "logGroupName" : "my_log_group_2",
            "eventName" : "Application",
            "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ]
          }
        ]
      }, {
        "subComponentType" : "AWS::EC2::Volume",
        "alarmMetrics" : [
          {
            "alarmMetricName" : "VolumeQueueLength"
          }, {
            "alarmMetricName" : "BurstBalance"
          }
        ]
      }
    ],
    "alarms" : [
      {
        "alarmName" : "my_asg_alarm",
```

```
        "severity" : "LOW"
      }
    ]
  }
```

Cluster do Amazon EKS

O exemplo a seguir mostra uma configuração de componente no formato JSON para o cluster do Amazon EKS.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName": "cluster_failed_node_count",
          "monitor":true
       },
       {
          "alarmMetricName": "node_cpu_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName": "node_cpu_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_filesystem_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_memory_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName": "node_memory_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_network_total_bytes",
          "monitor":true
       },
       {
          "alarmMetricName": "pod_cpu_reserved_capacity",
          "monitor":true
```

```
      },
      {
         "alarmMetricName": "pod_cpu_utilization",
         "monitor":true
      },
      {
         "alarmMetricName": "pod_cpu_utilization_over_pod_limit",
         "monitor":true
      },
      {
         "alarmMetricName": "pod_memory_reserved_capacity",
         "monitor":true
      },
      {
         "alarmMetricName": "pod_memory_utilization",
         "monitor":true
      },
      {
         "alarmMetricName": "pod_memory_utilization_over_pod_limit",
         "monitor":true
      },
      {
         "alarmMetricName": "pod_network_rx_bytes",
         "monitor":true
      },
      {
         "alarmMetricName": "pod_network_tx_bytes",
         "monitor":true
      }
   ],
   "logs":[
      {
         "logGroupName": "/aws/containerinsights/kubernetes/application",
         "logType":"APPLICATION",
         "monitor":true,
         "encoding":"utf-8"
      }
   ],
   "subComponents":[
      {
         "subComponentType":"AWS::EC2::Instance",
         "alarmMetrics":[
            {
               "alarmMetricName":"CPUUtilization",
```

```
                "monitor":true
            },
            {
                "alarmMetricName":"StatusCheckFailed",
                "monitor":true
            },
            {
                "alarmMetricName":"disk_used_percent",
                "monitor":true
            },
            {
                "alarmMetricName":"mem_used_percent",
                "monitor":true
            }
        ],
        "logs":[
            {
                "logGroupName":"APPLICATION-KubernetesClusterOnEC2-IAD",
                "logPath":"",
                "logType":"APPLICATION",
                "monitor":true,
                "encoding":"utf-8"
            }
        ],
        "processes" : [
          {
              "processName" : "my_process",
              "alarmMetrics" : [
                  {
                      "alarmMetricName" : "procstat cpu_usage",
                      "monitor" : true
                  }, {
                      "alarmMetricName" : "procstat memory_rss",
                      "monitor" : true
                  }
              ]
          }
      ],
        "windowsEvents":[
            {
                "logGroupName":"my_log_group_2",
                "eventName":"Application",
                "eventLevels":[
                   "ERROR",
```

```
                "WARNING",
                "CRITICAL"
             ],
             "monitor":true
          }
       ]
    },
    {
       "subComponentType":"AWS::AutoScaling::AutoScalingGroup",
       "alarmMetrics":[
          {
             "alarmMetricName":"CPUCreditBalance",
             "monitor":true
          },
          {
             "alarmMetricName":"EBSIOBalance%",
             "monitor":true
          }
       ]
    },
    {
       "subComponentType":"AWS::EC2::Volume",
       "alarmMetrics":[
          {
             "alarmMetricName":"VolumeReadBytes",
             "monitor":true
          },
          {
             "alarmMetricName":"VolumeWriteBytes",
             "monitor":true
          },
          {
             "alarmMetricName":"VolumeReadOps",
             "monitor":true
          },
          {
             "alarmMetricName":"VolumeWriteOps",
             "monitor":true
          },
          {
             "alarmMetricName":"VolumeQueueLength",
             "monitor":true
          },
          {
```

```
                "alarmMetricName":"BurstBalance",
                "monitor":true
             }
          ]
       }
    ]
}
```

> **Note**
>
> - A seção `subComponents` de `AWS::EC2::Instance`, `AWS::EC2::Volume` e `AWS::AutoScaling::AutoScalingGroup`aplica-se somente ao cluster do Amazon EKS em execução no tipo de inicialização do EC2.
> - A seção `windowsEvents` de `AWS::EC2::Instance` em `subComponents` aplica-se somente ao Windows em execução em instâncias do Amazon EC2.

Instância do Amazon Elastic Compute Cloud (EC2)

O exemplo a seguir mostra uma configuração de componente no formato JSON para uma instância do Amazon EC2.

> **Important**
>
> Quando uma instância do Amazon EC2 entra em um estado `stopped`, ela é removida do monitoramento. Quando retorna a um estado `running`, ela é adicionada à lista de Unmonitored components (Componentes não monitorados) na página Application details (Detalhes da aplicação) do console do CloudWatch Application Insights. Se o monitoramento automático de novos recursos estiver habilitado para a aplicação, a instância será adicionada à lista de Monitored components (Componentes monitorados). No entanto, os logs e as métricas são definidos para o padrão da workload. A configuração anterior de log e métricas não é salva.

```
{
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUUtilization",
        "monitor" : true
```

```
      }, {
        "alarmMetricName" : "StatusCheckFailed"
      }
    ],
    "logs" : [
      {
        "logGroupName" : "my_log_group",
        "logPath" : "C:\\LogFolder\\*",
        "logType" : "APPLICATION",
        "monitor" : true
      },
      {
        "logGroupName" : "my_log_group_2",
        "logPath" : "C:\\LogFolder2\\*",
        "logType" : "IIS",
        "encoding" : "utf-8"
      }
    ],
    "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ],
    "windowsEvents" : [
      {
        "logGroupName" : "my_log_group_3",
        "eventName" : "Application",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
      }, {
        "logGroupName" : "my_log_group_4",
        "eventName" : "System",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
    }],
```

```
     "alarms" : [
      {
        "alarmName" : "my_instance_alarm_1",
        "severity" : "HIGH"
      },
      {
        "alarmName" : "my_instance_alarm_2",
        "severity" : "LOW"
      }
    ],
     "subComponents" : [
     {
       "subComponentType" : "AWS::EC2::Volume",
       "alarmMetrics" : [
       {
         "alarmMetricName" : "VolumeQueueLength",
         "monitor" : "true"
       },
       {
         "alarmMetricName" : "VolumeThroughputPercentage",
         "monitor" : "true"
       },
       {
         "alarmMetricName" : "BurstBalance",
         "monitor" : "true"
       }
    }]
  }
```

Amazon Elastic Container Service (Amazon ECS)

O exemplo a seguir mostra uma configuração de componente no formato JSON para o Amazon Elastic Container Service (Amazon ECS).

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"CpuUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilized",
          "monitor":true
```

```
        },
        {
           "alarmMetricName":"NetworkRxBytes",
           "monitor":true
        },
        {
           "alarmMetricName":"NetworkTxBytes",
           "monitor":true
        },
        {
           "alarmMetricName":"RunningTaskCount",
           "monitor":true
        },
        {
           "alarmMetricName":"PendingTaskCount",
           "monitor":true
        },
        {
           "alarmMetricName":"StorageReadBytes",
           "monitor":true
        },
        {
           "alarmMetricName":"StorageWriteBytes",
           "monitor":true
        }
     ],
     "logs":[
        {
           "logGroupName":"/ecs/my-task-definition",
           "logType":"APPLICATION",
           "monitor":true
        }
     ],
     "subComponents":[
        {
           "subComponentType":"AWS::ElasticLoadBalancing::LoadBalancer",
           "alarmMetrics":[
              {
                 "alarmMetricName":"HTTPCode_Backend_4XX",
                 "monitor":true
              },
              {
                 "alarmMetricName":"HTTPCode_Backend_5XX",
                 "monitor":true
```

```
            },
            {
               "alarmMetricName":"Latency",
               "monitor":true
            },
            {
               "alarmMetricName":"SurgeQueueLength",
               "monitor":true
            },
            {
               "alarmMetricName":"UnHealthyHostCount",
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::ElasticLoadBalancingV2::LoadBalancer",
         "alarmMetrics":[
            {
               "alarmMetricName":"HTTPCode_Target_4XX_Count",
               "monitor":true
            },
            {
               "alarmMetricName":"HTTPCode_Target_5XX_Count",
               "monitor":true
            },
            {
               "alarmMetricName":"TargetResponseTime",
               "monitor":true
            },
            {
               "alarmMetricName":"UnHealthyHostCount",
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Instance",
         "alarmMetrics":[
            {
               "alarmMetricName":"CPUUtilization",
               "monitor":true
            },
            {
```

```
                "alarmMetricName":"StatusCheckFailed",
                "monitor":true
            },
            {
                "alarmMetricName":"disk_used_percent",
                "monitor":true
            },
            {
                "alarmMetricName":"mem_used_percent",
                "monitor":true
            }
        ],
        "logs":[
            {
                "logGroupName":"my_log_group",
                "logPath":"/mylog/path",
                "logType":"APPLICATION",
                "monitor":true
            }
        ],
        "processes" : [
            {
                "processName" : "my_process",
                "alarmMetrics" : [
                    {
                        "alarmMetricName" : "procstat cpu_usage",
                        "monitor" : true
                    }, {
                        "alarmMetricName" : "procstat memory_rss",
                        "monitor" : true
                    }
                ]
            }
        ],
        "windowsEvents":[
            {
                "logGroupName":"my_log_group_2",
                "eventName":"Application",
                "eventLevels":[
                    "ERROR",
                    "WARNING",
                    "CRITICAL"
                ],
                "monitor":true
```

```
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Volume",
         "alarmMetrics":[
            {
               "alarmMetricName":"VolumeQueueLength",
               "monitor":"true"
            },
            {
               "alarmMetricName":"VolumeThroughputPercentage",
               "monitor":"true"
            },
            {
               "alarmMetricName":"BurstBalance",
               "monitor":"true"
            }
         ]
      }
   ]
}
```

> **Note**
>
> - A seção `subComponents` de `AWS::EC2::Instance` e `AWS::EC2::Volume` aplica-se somente aos clusters do Amazon ECS com serviço ECS ou tarefa ECS em execução no tipo de inicialização do EC2.
> - A seção `windowsEvents` de `AWS::EC2::Instance` em `subComponents` aplica-se somente ao Windows em execução em instâncias do Amazon EC2.

Serviços do Amazon ECS

O exemplo a seguir mostra uma configuração de componente no formato JSON para um serviço do Amazon ECS.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"CPUUtilization",
```

```
            "monitor":true
         },
         {
            "alarmMetricName":"MemoryUtilization",
            "monitor":true
         },
         {
            "alarmMetricName":"CpuUtilized",
            "monitor":true
         },
         {
            "alarmMetricName":"MemoryUtilized",
            "monitor":true
         },
         {
            "alarmMetricName":"NetworkRxBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"NetworkTxBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"RunningTaskCount",
            "monitor":true
         },
         {
            "alarmMetricName":"PendingTaskCount",
            "monitor":true
         },
         {
            "alarmMetricName":"StorageReadBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"StorageWriteBytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName":"/ecs/my-task-definition",
            "logType":"APPLICATION",
            "monitor":true
```

```
        }
    ],
    "subComponents":[
        {
            "subComponentType":"AWS::ElasticLoadBalancing::LoadBalancer",
            "alarmMetrics":[
                {
                    "alarmMetricName":"HTTPCode_Backend_4XX",
                    "monitor":true
                },
                {
                    "alarmMetricName":"HTTPCode_Backend_5XX",
                    "monitor":true
                },
                {
                    "alarmMetricName":"Latency",
                    "monitor":true
                },
                {
                    "alarmMetricName":"SurgeQueueLength",
                    "monitor":true
                },
                {
                    "alarmMetricName":"UnHealthyHostCount",
                    "monitor":true
                }
            ]
        },
        {
            "subComponentType":"AWS::ElasticLoadBalancingV2::LoadBalancer",
            "alarmMetrics":[
                {
                    "alarmMetricName":"HTTPCode_Target_4XX_Count",
                    "monitor":true
                },
                {
                    "alarmMetricName":"HTTPCode_Target_5XX_Count",
                    "monitor":true
                },
                {
                    "alarmMetricName":"TargetResponseTime",
                    "monitor":true
                },
                {
```

```
                "alarmMetricName":"UnHealthyHostCount",
                "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Instance",
         "alarmMetrics":[
            {
               "alarmMetricName":"CPUUtilization",
               "monitor":true
            },
            {
               "alarmMetricName":"StatusCheckFailed",
               "monitor":true
            },
            {
               "alarmMetricName":"disk_used_percent",
               "monitor":true
            },
            {
               "alarmMetricName":"mem_used_percent",
               "monitor":true
            }
         ],
         "logs":[
            {
               "logGroupName":"my_log_group",
               "logPath":"/mylog/path",
               "logType":"APPLICATION",
               "monitor":true
            }
         ],
         "processes" : [
            {
               "processName" : "my_process",
               "alarmMetrics" : [
               {
                   "alarmMetricName" : "procstat cpu_usage",
                   "monitor" : true
               }, {
                   "alarmMetricName" : "procstat memory_rss",
                   "monitor" : true
               }
```

```
            ]
        }
    ],
          "windowsEvents":[
             {
                "logGroupName":"my_log_group_2",
                "eventName":"Application",
                "eventLevels":[
                   "ERROR",
                   "WARNING",
                   "CRITICAL"
                ],
                "monitor":true
             }
          ]
       },
       {
          "subComponentType":"AWS::EC2::Volume",
          "alarmMetrics":[
             {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":"true"
             },
             {
                "alarmMetricName":"VolumeThroughputPercentage",
                "monitor":"true"
             },
             {
                "alarmMetricName":"BurstBalance",
                "monitor":"true"
             }
          ]
       }
    ]
}
```

Note

- A seção `subComponents` de `AWS::EC2::Instance` e `AWS::EC2::Volume` aplica-se somente ao Amazon ECS em execução no tipo de inicialização do EC2.

- A seção `windowsEvents` de `AWS::EC2::Instance` em `subComponents` aplica-se somente ao Windows em execução em instâncias do Amazon EC2.

Tarefas do Amazon ECS

O exemplo a seguir mostra uma configuração de componente no formato JSON para uma tarefa do Amazon ECS.

```
{
    "logs":[
       {
          "logGroupName":"/ecs/my-task-definition",
          "logType":"APPLICATION",
          "monitor":true
       }
    ],
    "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ]
 }
```

Amazon Elastic File System (Amazon EFS)

O exemplo a seguir mostra uma configuração de componente no formato JSON para o Amazon EFS.

```
{
   "alarmMetrics": [
     {
       "alarmMetricName": "BurstCreditBalance",
       "monitor": true
```

```
    },
    {
      "alarmMetricName": "PercentIOLimit",
      "monitor": true
    },
    {
      "alarmMetricName": "PermittedThroughput",
      "monitor": true
    },
    {
      "alarmMetricName": "MeteredIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "TotalIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataReadIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "MetadataIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "ClientConnections",
      "monitor": true
    },
    {
      "alarmMetricName": "TimeSinceLastSync",
      "monitor": true
    },
    {
      "alarmMetricName": "Throughput",
      "monitor": true
    },
    {
      "alarmMetricName": "PercentageOfPermittedThroughputUtilization",
      "monitor": true
```

```
    },
    {
      "alarmMetricName": "ThroughputIOPS",
      "monitor": true
    },
    {
      "alarmMetricName": "PercentThroughputDataReadIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "PercentThroughputDataWriteIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "PercentageOfIOPSDataReadIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "PercentageOfIOPSDataWriteIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "AverageDataReadIOBytesSize",
      "monitor": true
    },
    {
      "alarmMetricName": "AverageDataWriteIOBytesSize",
      "monitor": true
    }
  ],
  "logs": [
   {
   "logGroupName": "/aws/efs/utils",
   "logType": "EFS_MOUNT_STATUS",
   "monitor": true,
   }
  ]
}
```

Amazon FSx

O exemplo a seguir mostra uma configuração de componente no formato JSON para o Amazon FSx.

```
{
```

```
  "alarmMetrics": [
    {
      "alarmMetricName": "DataReadBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataReadOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "MetadataOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "FreeStorageCapacity",
      "monitor": true
    }
  ]
}
```

Amazon Relational Database Service (RDS) Aurora MySQL

O exemplo a seguir mostra uma configuração de componente no formato JSON do Amazon RDS Aurora MySQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    },
    {
      "alarmMetricName": "CommitLatency",
      "monitor": true
    }
```

```
  ],
  "logs": [
    {
      "logType": "MYSQL",
      "monitor": true,
    },
    {
      "logType": "MYSQL_SLOW_QUERY",
      "monitor": false
    }
  ]
}
```

Instância do banco de dados relacional (RDS) da Amazon

O exemplo a seguir mostra uma configuração de componente no formato JSON para uma instância do Amazon RDS.

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "BurstBalance",
      "monitor" : true
    }, {
      "alarmMetricName" : "WriteThroughput",
      "monitor" : false
    }
  ],

  "alarms" : [
    {
      "alarmName" : "my_rds_instance_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

Verificação de integridade do Amazon Route 53

O exemplo a seguir mostra uma configuração de componente no formato JSON para verificação de integridade do Amazon Route 53.

```
{
```

```
  "alarmMetrics": [
    {
      "alarmMetricName": "ChildHealthCheckHealthyCount",
      "monitor": true
    },
    {
      "alarmMetricName": "ConnectionTime",
      "monitor": true
    },
    {
      "alarmMetricName": "HealthCheckPercentageHealthy",
      "monitor": true
    },
    {
      "alarmMetricName": "HealthCheckStatus",
      "monitor": true
    },
    {
      "alarmMetricName": "SSLHandshakeTime",
      "monitor": true
    },
    {
      "alarmMetricName": "TimeToFirstByte",
      "monitor": true
    }
  ]
}
```

Zona hospedada do Amazon Route 53

O exemplo a seguir mostra uma configuração de componente no formato JSON para zona hospedada do Amazon Route 53.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "DNSQueries",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECInternalFailure",
      "monitor": true
    },
```

```
    {
      "alarmMetricName": "DNSSECKeySigningKeysNeedingAction",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeyMaxNeedingActionAge",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeyAge",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logGroupName":"/hosted-zone/logs",
      "logType": "ROUTE53_DNS_PUBLIC_QUERY_LOGS",
      "monitor": true
    }
  ]
}
```

Endpoint do Amazon Route 53 Resolver

O exemplo a seguir mostra uma configuração de componente no formato JSON para o endpoint Amazon Route 53 Resolver.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "EndpointHealthyENICount",
      "monitor": true
    },
    {
      "alarmMetricName": "EndpointUnHealthyENICount",
      "monitor": true
    },
    {
      "alarmMetricName": "InboundQueryVolume",
      "monitor": true
    },
    {
      "alarmMetricName": "OutboundQueryVolume",
```

```
      "monitor": true
    },
    {
      "alarmMetricName": "OutboundQueryAggregateVolume",
      "monitor": true
    }
  ]
}
```

Configurações de logs de consulta do Amazon Route 53 Resolver

O exemplo a seguir mostra uma configuração de componente no formato JSON para a configuração de logs de consulta do Amazon Route 53 Resolver.

```
{
  "logs": [
    {
      "logGroupName": "/resolver-query-log-config/logs",
      "logType": "ROUTE53_RESOLVER_QUERY_LOGS",
      "monitor": true
    }
  ]
}
```

Bucket do Amazon S3

O exemplo a seguir mostra uma configuração de componentes no formato JSON para um bucket do Amazon S3.

```
{
    "alarmMetrics" : [
        {
            "alarmMetricName" : "ReplicationLatency",
            "monitor" : true
        }, {
            "alarmMetricName" : "5xxErrors",
            "monitor" : true
        }, {
            "alarmMetricName" : "BytesDownloaded"
            "monitor" : true
        }
    ]
```

```
}
```

Amazon Simple Queue Service (SQS)

O exemplo a seguir mostram configurações de componentes no formato JSON para o Amazon Simple Queue Service.

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "ApproximateAgeOfOldestMessage"
    }, {
      "alarmMetricName" : "NumberOfEmptyReceives"
    }
  ],
  "alarms" : [
    {
      "alarmName" : "my_sqs_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

Tópico do Amazon SNS

O exemplo a seguir mostra uma configuração de componente no formato JSON para um tópico do Amazon SNS.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "NumberOfNotificationsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFilteredOut-InvalidAttributes",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFilteredOut-NoMessageAttributes",
      "monitor": true
    },
    {
```

```
      "alarmMetricName": "NumberOfNotificationsFailedToRedriveToDlq",
      "monitor": true
    }
  ]
}
```

Amazon Virtual Private Cloud (Amazon VPC)

O exemplo a seguir mostra uma configuração de componente no formato JSON para a Amazon VPC.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "NetworkAddressUsage",
      "monitor": true
    },
    {
      "alarmMetricName": "NetworkAddressUsagePeered",
      "monitor": true
    },
    {
      "alarmMetricName": "VPCFirewallQueryVolume",
      "monitor": true
    }
  ]
}
```

Gateways de conversão de endereços de rede (NAT) da Amazon VPC

O exemplo a seguir mostra uma configuração de componentes no formato JSON para gateways de NAT.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ErrorPortAllocation",
      "monitor": true
    },
    {
      "alarmMetricName": "IdleTimeoutCount",
      "monitor": true
    }
```

```
  ]
}
```

Etapas da API REST do API Gateway

O exemplo a seguir mostra uma configuração de componente no formato JSON para etapas da API REST do API Gateway.

```
{
     "alarmMetrics" : [
         {
             "alarmMetricName" : "4XXError",
             "monitor" : true
         },
         {
             "alarmMetricName" : "5XXError",
             "monitor" : true
         }
     ],
    "logs" : [
        {
            "logType" : "API_GATEWAY_EXECUTION",
            "monitor" : true
        },
        {
            "logType" : "API_GATEWAY_ACCESS",
            "monitor" : true
        }
    ]
}
```

Elastic Load Balancing da aplicação

O exemplo a seguir mostra uma configuração de componente no formato JSON do Application Elastic Load Balancing.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ActiveConnectionCount",
    }, {
      "alarmMetricName": "TargetResponseTime"
    }
```

```
  ],
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "CPUUtilization",
        }, {
          "alarmMetricName": "StatusCheckFailed"
        }
      ],
      "logs": [
        {
          "logGroupName": "my_log_group",
          "logPath": "C:\\LogFolder\\*",
          "logType": "APPLICATION",
        }
      ],
      "windowsEvents": [
        {
          "logGroupName": "my_log_group_2",
          "eventName": "Application",
          "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ]
        }
      ]
    }, {
      "subComponentType": "AWS::EC2::Volume",
      "alarmMetrics": [
        {
          "alarmMetricName": "VolumeQueueLength",
        }, {
          "alarmMetricName": "BurstBalance"
        }
      ]
    }
  ],

  "alarms": [
    {
      "alarmName": "my_alb_alarm",
      "severity": "LOW"
    }
  ]
```

```
}
```

Funções do AWS Lambda

O exemplo a seguir mostra uma configuração de componente no formato JSON para a Função AWS Lambda.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "Errors",
      "monitor": true
    },
    {
      "alarmMetricName": "Throttles",
      "monitor": true
    },
    {
      "alarmMetricName": "IteratorAge",
      "monitor": true
    },
    {
      "alarmMetricName": "Duration",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "DEFAULT",
      "monitor": true
    }
  ]
}
```

Grupo de regras do AWS Network Firewall

O exemplo a seguir mostra uma configuração de componente no formato JSON para o grupo de regras AWS Network Firewall.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "FirewallRuleGroupQueryVolume",
```

```
      "monitor": true
    }
  ]
}
```

associação do grupos de regras AWS Network Firewall

O exemplo a seguir mostra uma configuração de componente no formato JSON para a associação do grupo de regras AWS Network Firewall.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "FirewallRuleGroupQueryVolume",
      "monitor": true
    }
  ]
}
```

AWS Step Functions

O exemplo a seguir mostra uma configuração de componentes no formato JSON para AWS Step Functions.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ExecutionsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "LambdaFunctionsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "ProvisionedRefillRate",
      "monitor": true
    }
  ],
  "logs": [
    {
     "logGroupName": "/aws/states/HelloWorld-Logs",
      "logType": "STEP_FUNCTION",
```

```
      "monitor": true,
    }
  ]
}
```

Instâncias do Amazon EC2 agrupadas pelo cliente

O exemplo a seguir mostra uma configuração de componente no formato JSON para instâncias do Amazon EC2 agrupadas pelo cliente.

```
{
    "subComponents": [
      {
        "subComponentType": "AWS::EC2::Instance",
        "alarmMetrics": [
          {
            "alarmMetricName": "CPUUtilization",
          },
          {
            "alarmMetricName": "StatusCheckFailed"
          }
        ],
        "logs": [
          {
            "logGroupName": "my_log_group",
            "logPath": "C:\\LogFolder\\*",
            "logType": "APPLICATION",
          }
        ],
        "processes": [
            {
                "processName": "my_process",
                "alarmMetrics": [
                    {
                        "alarmMetricName": "procstat cpu_usage",
                        "monitor": true
                    }, {
                        "alarmMetricName": "procstat memory_rss",
                        "monitor": true
                    }
                ]
            }
        ],
```

```
        "windowsEvents": [
          {
            "logGroupName": "my_log_group_2",
            "eventName": "Application",
            "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ]
          }
        ]
      }, {
        "subComponentType": "AWS::EC2::Volume",
        "alarmMetrics": [
          {
            "alarmMetricName": "VolumeQueueLength",
          }, {
            "alarmMetricName": "BurstBalance"
          }
        ]
      }
    ],
    "alarms": [
      {
        "alarmName": "my_alarm",
        "severity": "MEDIUM"
      }
    ]
  }
```

Elastic Load Balancing

O exemplo a seguir mostra uma configuração de componente no formato JSON para o Elastic Load Balancing.

```
{
    "alarmMetrics": [
      {
        "alarmMetricName": "EstimatedALBActiveConnectionCount"
      }, {
        "alarmMetricName": "HTTPCode_Backend_5XX"
      }
    ],
    "subComponents": [
      {
        "subComponentType": "AWS::EC2::Instance",
        "alarmMetrics": [
```

```
          {
            "alarmMetricName": "CPUUtilization"
          }, {
            "alarmMetricName": "StatusCheckFailed"
          }
        ],
        "logs": [
          {
            "logGroupName": "my_log_group",
            "logPath": "C:\\LogFolder\\*",
            "logType": "APPLICATION"
          }
        ],
        "processes": [
          {
            "processName": "my_process",
            "alarmMetrics": [
                {
                  "alarmMetricName": "procstat cpu_usage",
                  "monitor": true
                }, {
                  "alarmMetricName": "procstat memory_rss",
                  "monitor": true
                }
             ]
          }
        ],
        "windowsEvents": [
          {
            "logGroupName": "my_log_group_2",
            "eventName": "Application",
            "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ],
            "monitor": true
          }
        ]
      }, {
        "subComponentType": "AWS::EC2::Volume",
        "alarmMetrics": [
          {
            "alarmMetricName": "VolumeQueueLength"
          }, {
            "alarmMetricName": "BurstBalance"
          }
        ]
```

```
      }
    ],

    "alarms": [
      {
        "alarmName": "my_elb_alarm",
        "severity": "HIGH"
      }
    ]
  }
```

Java

O exemplo a seguir mostra uma configuração de componentes no formato JSON para Java.

```
{
  "alarmMetrics": [ {
    "alarmMetricName": "java_lang_threading_threadcount",
    "monitor": true
  },
  {
    "alarmMetricName": "java_lang_memory_heapmemoryusage_used",
    "monitor": true
  },
  {
    "alarmMetricName": "java_lang_memory_heapmemoryusage_committed",
    "monitor": true
  }],
  "logs": [ ],
  "JMXPrometheusExporter": {
      "hostPort": "8686",
      "prometheusPort": "9404"
  }
}
```

> **Note**
>
> O Application Insights não é compatível com a configuração de autenticação para o Prometheus JMX Exporter. Para obter informações sobre como configurar a autenticação, consulte a Exemplo de configuração do Prometheus JMX Exporter.

Kubernetes no Amazon EC2

O exemplo a seguir mostra uma configuração de componente no formato JSON para Kubernetes no Amazon EC2.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"cluster_failed_node_count",
          "monitor":true
       },
       {
          "alarmMetricName":"node_cpu_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName":"node_cpu_utilization",
          "monitor":true
       },
       {
          "alarmMetricName":"node_filesystem_utilization",
          "monitor":true
       },
       {
          "alarmMetricName":"node_memory_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName":"node_memory_utilization",
          "monitor":true
       },
       {
          "alarmMetricName":"node_network_total_bytes",
          "monitor":true
       },
       {
          "alarmMetricName":"pod_cpu_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName":"pod_cpu_utilization",
          "monitor":true
       },
       {
```

```
            "alarmMetricName":"pod_cpu_utilization_over_pod_limit",
            "monitor":true
         },
         {
            "alarmMetricName":"pod_memory_reserved_capacity",
            "monitor":true
         },
         {
            "alarmMetricName":"pod_memory_utilization",
            "monitor":true
         },
         {
            "alarmMetricName":"pod_memory_utilization_over_pod_limit",
            "monitor":true
         },
         {
            "alarmMetricName":"pod_network_rx_bytes",
            "monitor":true
         },
         {
            "alarmMetricName":"pod_network_tx_bytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName":"/aws/containerinsights/kubernetes/application",
            "logType":"APPLICATION",
            "monitor":true,
            "encoding":"utf-8"
         }
      ],
      "subComponents":[
         {
            "subComponentType":"AWS::EC2::Instance",
            "alarmMetrics":[
               {
                  "alarmMetricName":"CPUUtilization",
                  "monitor":true
               },
               {
                  "alarmMetricName":"StatusCheckFailed",
                  "monitor":true
               },
```

```
            {
               "alarmMetricName":"disk_used_percent",
               "monitor":true
            },
            {
               "alarmMetricName":"mem_used_percent",
               "monitor":true
            }
         ],
         "logs":[
            {
               "logGroupName":"APPLICATION-KubernetesClusterOnEC2-IAD",
               "logPath":"",
               "logType":"APPLICATION",
               "monitor":true,
               "encoding":"utf-8"
            }
         ],
         "processes" : [
           {
               "processName" : "my_process",
               "alarmMetrics" : [
                   {
                       "alarmMetricName" : "procstat cpu_usage",
                       "monitor" : true
                   }, {
                       "alarmMetricName" : "procstat memory_rss",
                       "monitor" : true
                   }
               ]
           }
       ]
      },
      {
         "subComponentType":"AWS::EC2::Volume",
         "alarmMetrics":[
            {
               "alarmMetricName":"VolumeReadBytes",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeWriteBytes",
               "monitor":true
            },
```

```
            {
               "alarmMetricName":"VolumeReadOps",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeWriteOps",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeQueueLength",
               "monitor":true
            },
            {
               "alarmMetricName":"BurstBalance",
               "monitor":true
            }
         ]
      }
   ]
}
```

RDS MariaDB e o RDS MySQL

O exemplo a seguir mostra uma configuração de componentes no formato JSON para RDS MariaDB e RDS MySQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "MYSQL",
      "monitor": true,
    },
    {
      "logType": "MYSQL_SLOW_QUERY",
      "monitor": false
    }
  ]
```

```
}
```

RDS Oracle

O exemplo a seguir mostra uma configuração de componente no formato JSON para o RDS Oracle.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "ORACLE_ALERT",
      "monitor": true,
    },
    {
      "logType": "ORACLE_LISTENER",
      "monitor": false
    }
  ]
}
```

RDS PostgreSQL

O exemplo a seguir mostra uma configuração de componente no formato JSON para RDS PostgreSQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "POSTGRESQL",
      "monitor": true
    }
  ]
```

```
}
```

SAP ASE no Amazon EC2

O exemplo a seguir mostra uma configuração de componente no formato JSON para o SAP ASE no Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "asedb_database_availability",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_trunc_log_on_chkpt_enabled",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_db_backup_age_in_days",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_transaction_log_backup_age_in_hours",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_suspected_database",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_log_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_locked_login",
          "monitor": true
```

```
        },
        {
          "alarmMetricName": "asedb_data_cache_hit_ratio",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_ASE_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2.log",
          "logType": "SAP_ASE_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_BACKUP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2_BS.log",
          "logType": "SAP_ASE_BACKUP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
  ],
  "sapAsePrometheusExporter": {
    "sapAseSid": "ASE",
    "sapAsePort": "4901",
    "sapAseSecretName": "ASE_DB_CREDS",
    "prometheusPort": "9399",
    "agreeToEnableASEMonitoring": true
}
```

Alta disponibilidade do SAP ASE no Amazon EC2

O exemplo a seguir mostra uma configuração de componente no formato JSON para a alta disponibilidade do SAP ASE no Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "asedb_database_availability",
          "monitor": true
```

```
        },
        {
          "alarmMetricName": "asedb_trunc_log_on_chkpt_enabled",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_db_backup_age_in_days",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_transaction_log_backup_age_in_hours",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_suspected_database",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_state",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_mode",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_latency_in_minutes",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_ASE_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2.log",
          "logType": "SAP_ASE_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_BACKUP_SERVER_LOGS-my-resource-group",
```

```
          "logPath": "/sybase/SY2/ASE-*/install/SY2_BS.log",
          "logType": "SAP_ASE_BACKUP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_REP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/DM/repservername/repservername.log",
          "logType": "SAP_ASE_REP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_RMA_AGENT_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/DM/RMA-*/instances/AgentContainer/logs/",
          "logType": "SAP_ASE_RMA_AGENT_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_FAULT_MANAGER_LOGS-my-resource-group",
          "logPath": "/opt/sap/FaultManager/dev_sybdbfm",
          "logType": "SAP_ASE_FAULT_MANAGER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
  ],
  "sapAsePrometheusExporter": {
    "sapAseSid": "ASE",
    "sapAsePort": "4901",
    "sapAseSecretName": "ASE_DB_CREDS",
    "prometheusPort": "9399",
    "agreeToEnableASEMonitoring": true
}
```

SAP HANA no Amazon EC2

O exemplo a seguir mostra uma configuração de componente no formato JSON para SAP HANA no Amazon EC2.

```
{
  "subComponents": [
    {
```

```
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "hanadb_server_startup_time_variations_seconds",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_5_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_4_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_out_of_memory_events_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_max_trigger_read_ratio_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_table_allocation_limit_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_cpu_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_plan_cache_hit_ratio_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_last_data_backup_age_days",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_HANA_TRACE-my-resourge-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.trc",
          "logType": "SAP_HANA_TRACE",
```

```
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_HANA_LOGS-my-resource-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.log",
          "logType": "SAP_HANA_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "hanaPrometheusExporter": {
    "hanaSid": "HDB",
    "hanaPort": "30013",
    "hanaSecretName": "HANA_DB_CREDS",
    "prometheusPort": "9668"
  }
}
```

SAP HANA High Availability no Amazon EC2

O exemplo a seguir mostra uma configuração de componente no formato JSON para SAP HANA High Availability no Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "hanadb_server_startup_time_variations_seconds",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_5_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_4_alerts_count",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "hanadb_out_of_memory_events_count",
          "monitor": true
        },
        {
          "alarmMetricName": "ha_cluster_pacemaker_stonith_enabled",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_HANA_TRACE-my-resourge-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.trc",
          "logType": "SAP_HANA_TRACE",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_HANA_HIGH_AVAILABILITY-my-resource-group",
          "logPath": "/var/log/pacemaker/pacemaker.log",
          "logType": "SAP_HANA_HIGH_AVAILABILITY",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "hanaPrometheusExporter": {
    "hanaSid": "HDB",
    "hanaPort": "30013",
    "hanaSecretName": "HANA_DB_CREDS",
    "prometheusPort": "9668"
  },
  "haClusterPrometheusExporter": {
    "prometheusPort": "9664"
  }
}
```

SAP NetWeaver no Amazon EC2

O exemplo a seguir mostra uma configuração de componente no formato JSON para o SAP NetWeaver no Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "CPUUtilization",
          "monitor": true
        },
        {
          "alarmMetricName": "StatusCheckFailed",
          "monitor": true
        },
        {
          "alarmMetricName": "disk_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "mem_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialog",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialogRFC",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_DBRequestTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_LongRunners",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_AbortedJobs",
```

```
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_BasisSystem",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Database",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Security",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_System",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_QueueTime",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Availability",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_start_service_processes",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_dispatcher_queue_now",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_dispatcher_queue_max",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_enqueue_server_locks_max",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_enqueue_server_locks_now",
```

```
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_replication_state",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_NETWEAVER_DEV_TRACE_LOGS-NetWeaver-ML4",
          "logPath": "/usr/sap/ML4/*/work/dev_w*",
          "logType": "SAP_NETWEAVER_DEV_TRACE_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "netWeaverPrometheusExporter": {
    "sapSid": "ML4",
    "instanceNumbers": [
      "00",
      "11"
    ],
    "prometheusPort": "9680"
  }
}
```

Alta disponibilidade do SAP NetWeaver no Amazon EC2

O exemplo a seguir mostra uma configuração de componente no formato JSON para alta disponibilidade do SAP NetWeaver no Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
```

```
            "alarmMetricName": "ha_cluster_corosync_ring_errors",
            "monitor": true
          },
          {
            "alarmMetricName": "ha_cluster_pacemaker_fail_count",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_HA_check_failover_config_state",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_HA_get_failover_config_HAActive",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_AbortedJobs",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Availability",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_BasisSystem",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_DBRequestTime",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Database",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_FrontendResponseTime",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_LongRunners",
            "monitor": true
          },
          {
```

```
          "alarmMetricName": "sap_alerts_QueueTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialog",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialogRFC",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Security",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Shortdumps",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_SqlError",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_System",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_replication_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_start_service_processes",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_NETWEAVER_DEV_TRACE_LOGS-NetWeaver-PR1",
          "logPath": "/usr/sap/<SID>/D*/work/dev_w*",
```

```
          "logType": "SAP_NETWEAVER_DEV_TRACE_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "haClusterPrometheusExporter": {
    "prometheusPort": "9664"
  },
  "netWeaverPrometheusExporter": {
    "sapSid": "PR1",
    "instanceNumbers": [
      "11",
      "12"
    ],
    "prometheusPort": "9680"
  }
}
```

Grupos de disponibilidade Always On do SQL

O exemplo a seguir mostra uma configuração de componente no formato JSON para o SQL Always On Availability Group.

```
{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Buffer cache hit ratio",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Page life expectancy",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics Processes blocked",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics User Connections",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Locks Number of Deadlocks/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:SQL Statistics Batch Requests/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica File Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log remaining for undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Send Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Mirrored Write Transaction/sec",
```

```
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Recovery Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redo Bytes Remaining",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redone Bytes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Total Log requiring undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Transaction Delay",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP-<RESOURCE_GROUP_NAME>",
      "logPath" : "C:\\Program Files\\Microsoft SQL Server\\MSSQL**.MSSQLSERVER\\MSSQL\
\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
```

```
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}
```

Instância de cluster de failover do SQL

O exemplo a seguir mostra uma configuração de componente no formato JSON para instância de cluster de failover do SQL.

```
{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
```

```
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Normal Messages Queue Length/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Urgent Message Queue Length/se",
      "monitor" : true
    }, {
      "alarmMetricName" : "Reconnect Count",
      "monitor" : true
    }, {
      "alarmMetricName" : "Unacknowledged Message Queue Length/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Messages Outstanding",
      "monitor" : true
    }, {
      "alarmMetricName" : "Messages Sent/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Database Update Messages/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Update Messages/sec",
      "monitor" : true
    }, {
```

```
      "alarmMetricName" : "Flushes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Crypto Checkpoints Saved/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Crypto Checkpoints Restored/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Registry Checkpoints Restored/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Registry Checkpoints Saved/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Cluster API Calls/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Resource API Calls/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Cluster Handles/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Resource Handles/sec",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL"],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
```

```
      "logGroupName" : "SQL_SERVER_FAILOVER_CLUSTER_INSTANCE-<RESOURCE_GROUP_NAME>",
      "logPath" : "\\\\amznfsxjmzbykwn.mydomain.aws\\SQLDB\\MSSQL**.MSSQLSERVER\\MSSQL\
\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}
```

Criar e configurar monitoramento do CloudWatch Application Insights usando modelos do CloudFormation

É possível adicionar o monitoramento do Application Insights, incluindo as principais métricas e telemetria, à aplicação, ao banco de dados e ao servidor Web, diretamente a partir de modelos do AWS CloudFormation.

Esta seção fornece exemplos de modelos do AWS CloudFormation nos formatos JSON e YAML para ajudar você a criar e configurar o monitoramento do Application Insights.

Para exibir o recurso e a referência de propriedade do Application Insights no Manual do usuário do AWS CloudFormation, consulte a referência de tipo de recurso do Application Insights.

Exemplos de modelo

- Crie uma aplicação do Application Insights para toda a pilha AWS CloudFormation
- Criar uma aplicação do Application Insights com configurações detalhadas
- Criar uma aplicação do Application Insights com configuração do componente modo CUSTOM
- Criar uma aplicação do Application Insights com configuração do componente modo DEFAULT
- Criar uma aplicação do Application Insights com configuração do componente modo DEFAULT_WITH_OVERWRITE

Crie uma aplicação do Application Insights para toda a pilha AWS CloudFormation

Para aplicar o modelo a seguir, é necessário criar recursos da AWS um ou mais grupos de recursos a partir dos quais criar aplicações do Application Insights para monitorar esses recursos. Para obter mais informações, consulte Conceitos básicos do AWS Resource Groups.

As duas primeiras partes do modelo a seguir especificam um recurso e um grupo de recursos. A última parte do modelo cria uma aplicação Application Insights para o grupo de recursos, mas não configura a aplicação ou o monitoramento de aplicação. Para obter mais informações, consulte os detalhes do comando CreateApplication na Referência de API do Amazon CloudWatch Application Insights.

Modelo em formato JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Test Resource Group stack",
    "Resources": {
        "EC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId" : "ami-abcd1234efgh5678i",
                "SecurityGroupIds" : ["sg-abcd1234"]
            }
        },
```

```
        ...
        "ResourceGroup": {
            "Type": "AWS::ResourceGroups::Group",
            "Properties": {
                "Name": "my_resource_group"
            }
        },
        "AppInsightsApp": {
            "Type": "AWS::ApplicationInsights::Application",
            "Properties": {
                "ResourceGroupName": "my_resource_group"
            },
            "DependsOn" : "ResourceGroup"
        }
    }
}
```

Modelo em formato YAML

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: Test Resource Group stack
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-abcd1234efgh5678i
      SecurityGroupIds:
      - sg-abcd1234
  ...
  ResourceGroup:
    Type: AWS::ResourceGroups::Group
    Properties:
      Name: my_resource_group
  AppInsightsApp:
    Type: AWS::ApplicationInsights::Application
    Properties:
      ResourceGroupName: my_resource_group
    DependsOn: ResourceGroup
```

A seção de modelo a seguir aplica a configuração de monitoramento padrão à aplicação do Application Insights. Para obter mais informações, consulte os detalhes do comando CreateApplication na Referência de API do Amazon CloudWatch Application Insights.

Quando `AutoConfigurationEnabled` esta definido como `true`, todos os componentes da aplicação são configurados com as configurações de monitoramento recomendadas para o nível `DEFAULT` da aplicação. Para obter mais informações sobre essas configurações e níveis, consulte DescribeComponentConfigurationRecommendation e UpdateComponentConfiguration na Referência da API do Amazon CloudWatch Application Insights.

Modelo em formato JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Test Application Insights Application stack",
    "Resources": {
        "AppInsightsApp": {
            "Type": "AWS::ApplicationInsights::Application",
            "Properties": {
                "ResourceGroupName": "my_resource_group",
                "AutoConfigurationEnabled": true
            }
        }
    }
}
```

Modelo em formato YAML

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: Test Application Insights Application stack
Resources:
  AppInsightsApp:
    Type: AWS::ApplicationInsights::Application
    Properties:
      ResourceGroupName: my_resource_group
      AutoConfigurationEnabled: true
```

Criar uma aplicação do Application Insights com configurações detalhadas

O modelo a seguir executa estas ações:

- Cria uma aplicação do Application Insights com a notificação do CloudWatch Events e o OpsCenter habilitados. Para obter mais informações, consulte os detalhes do comando CreateApplication na Referência de API do Amazon CloudWatch Application Insights.

- Marca a aplicação com duas etiquetas, uma das quais não tem valores de etiqueta. Para obter mais informações, consulte TagResource na Referência de API do Amazon CloudWatch Application Insights.
- Cria dois componentes personalizados do grupo de instâncias. Para obter mais informações, consulte CreateComponent na Referência de API do Amazon CloudWatch Application Insights.
- Cria dois conjuntos de padrões de log. Para obter mais informações, consulte CreateLogPattern na Referência de API do Amazon CloudWatch Application Insights.
- Define `AutoConfigurationEnabled` como `true`, que configura todos os componentes da aplicação com as configurações de monitoramento recomendadas para o nível `DEFAULT`. Para obter mais informações, consulte DescribeComponentConfigurationRecommendation na Referência de API do Amazon CloudWatch Application Insights.

Modelo em formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "CWEMonitorEnabled": true,
        "OpsCenterEnabled": true,
        "OpsItemSNSTopicArn": "arn:aws:sns:us-east-1:123456789012:my_topic",
        "AutoConfigurationEnabled": true,
        "Tags": [
            {
                "Key": "key1",
                "Value": "value1"
            },
            {
                "Key": "key2",
                "Value": ""
            }
        ],
        "CustomComponents": [
            {
                "ComponentName": "test_component_1",
                "ResourceList": [
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i"
                ]
            },
            {
```

```
                "ComponentName": "test_component_2",
                "ResourceList": [
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i",
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i"
                ]
            }
        ],
        "LogPatternSets": [
            {
                "PatternSetName": "pattern_set_1",
                "LogPatterns": [
                    {
                        "PatternName": "deadlock_pattern",
                        "Pattern": ".*\\sDeadlocked\\sSchedulers(([^\\w].*)|($))",
                        "Rank": 1
                    }
                ]
            },
            {
                "PatternSetName": "pattern_set_2",
                "LogPatterns": [
                    {
                        "PatternName": "error_pattern",
                        "Pattern": ".*[\\s\\[]ERROR[\\s\\]].*",
                        "Rank": 1
                    },
                    {
                        "PatternName": "warning_pattern",
                        "Pattern": ".*[\\s\\[]WARN(ING)?[\\s\\]].*",
                        "Rank": 10
                    }
                ]
            }
        ]
    }
}
```

Modelo em formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
```

```
  CWEMonitorEnabled: true
  OpsCenterEnabled: true
  OpsItemSNSTopicArn: arn:aws:sns:us-east-1:123456789012:my_topic
  AutoConfigurationEnabled: true
  Tags:
  - Key: key1
    Value: value1
  - Key: key2
    Value: ''
  CustomComponents:
  - ComponentName: test_component_1
    ResourceList:
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
  - ComponentName: test_component_2
    ResourceList:
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
  LogPatternSets:
  - PatternSetName: pattern_set_1
    LogPatterns:
    - PatternName: deadlock_pattern
      Pattern: ".*\\sDeadlocked\\sSchedulers(([^\\w].*)|($))"
      Rank: 1
  - PatternSetName: pattern_set_2
    LogPatterns:
    - PatternName: error_pattern
      Pattern: ".*[\\s\\[]ERROR[\\s\\]].*"
      Rank: 1
    - PatternName: warning_pattern
      Pattern: ".*[\\s\\[]WARN(ING)?[\\s\\]].*"
      Rank: 10
```

Criar uma aplicação do Application Insights com configuração do componente modo **CUSTOM**

O modelo a seguir executa estas ações:

- Cria uma aplicação do Application Insights. Para obter mais informações, consulte CreateApplication na Referência de API do Amazon CloudWatch Application Insights.
- O componente `my_component` define `ComponentConfigurationMode` como `CUSTOM`, o que faz com que esse componente seja configurado com a configuração especificada em `CustomComponentConfiguration`. Para obter mais informações, consulte

UpdateComponentConfiguration na Referência de API do Amazon CloudWatch Application Insights.

Modelo em formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group,
        "ComponentMonitoringSettings": [
            {
                "ComponentARN": "my_component",
                "Tier": "SQL_SERVER",
                "ComponentConfigurationMode": "CUSTOM",
                "CustomComponentConfiguration": {
                    "ConfigurationDetails": {
                        "AlarmMetrics": [
                            {
                                "AlarmMetricName": "StatusCheckFailed"
                            },
                            ...
                        ],
                        "Logs": [
                            {
                                "LogGroupName": "my_log_group_1",
                                "LogPath": "C:\\LogFolder_1\\*",
                                "LogType": "DOT_NET_CORE",
                                "Encoding": "utf-8",
                                "PatternSet": "my_pattern_set_1"
                            },
                            ...
                        ],
                        "WindowsEvents": [
                            {
                                "LogGroupName": "my_windows_event_log_group_1",
                                "EventName": "Application",
                                "EventLevels": [
                                    "ERROR",
                                    "WARNING",
                                    ...
                                ],
                                "Encoding": "utf-8",
                                "PatternSet": "my_pattern_set_2"
```

```
                        },
                        ...
                    ],
                    "Alarms": [
                        {
                            "AlarmName": "my_alarm_name",
                            "Severity": "HIGH"
                        },
                        ...
                    ]
                },
                "SubComponentTypeConfigurations": [
                    {
                        "SubComponentType": "EC2_INSTANCE",
                        "SubComponentConfigurationDetails": {
                            "AlarmMetrics": [
                                {
                                    "AlarmMetricName": "DiskReadOps"
                                },
                                ...
                            ],
                            "Logs": [
                                {
                                    "LogGroupName": "my_log_group_2",
                                    "LogPath": "C:\\LogFolder_2\\*",
                                    "LogType": "IIS",
                                    "Encoding": "utf-8",
                                    "PatternSet": "my_pattern_set_3"
                                },
                                ...
                            ],
                            "processes" : [
                                {
                                    "processName" : "my_process",
                                    "alarmMetrics" : [
                                {
                                    "alarmMetricName" : "procstat cpu_usage",
                                    "monitor" : true
                                }, {
                                    "alarmMetricName" : "procstat memory_rss",
                                    "monitor" : true
                                }
                            ]
                        }
```

```
                        ],
                                "WindowsEvents": [
                                    {
                                        "LogGroupName": "my_windows_event_log_group_2",
                                        "EventName": "Application",
                                        "EventLevels": [
                                            "ERROR",
                                            "WARNING",
                                            ...
                                        ],
                                        "Encoding": "utf-8",
                                        "PatternSet": "my_pattern_set_4"
                                    },
                                    ...
                                ]
                            }
                        }
                    ]
                }
            }
        ]
    }
}
```

Modelo em formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentARN: my_component
    Tier: SQL_SERVER
    ComponentConfigurationMode: CUSTOM
    CustomComponentConfiguration:
      ConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: StatusCheckFailed
        ...
        Logs:
        - LogGroupName: my_log_group_1
          LogPath: C:\LogFolder_1\*
          LogType: DOT_NET_CORE
```

```
          Encoding: utf-8
          PatternSet: my_pattern_set_1
        ...
        WindowsEvents:
        - LogGroupName: my_windows_event_log_group_1
          EventName: Application
          EventLevels:
          - ERROR
          - WARNING
          ...
          Encoding: utf-8
          PatternSet: my_pattern_set_2
        ...
        Alarms:
        - AlarmName: my_alarm_name
          Severity: HIGH
        ...
      SubComponentTypeConfigurations:
      - SubComponentType: EC2_INSTANCE
        SubComponentConfigurationDetails:
          AlarmMetrics:
          - AlarmMetricName: DiskReadOps
          ...
          Logs:
          - LogGroupName: my_log_group_2
            LogPath: C:\LogFolder_2\*
            LogType: IIS
            Encoding: utf-8
            PatternSet: my_pattern_set_3
          ...
          Processes:
          - ProcessName: my_process
            AlarmMetrics:
            - AlarmMetricName: procstat cpu_usage
              ...
            ...
          WindowsEvents:
          - LogGroupName: my_windows_event_log_group_2
            EventName: Application
            EventLevels:
            - ERROR
            - WARNING
            ...
            Encoding: utf-8
```

```
            PatternSet: my_pattern_set_4
          ...
```

Criar uma aplicação do Application Insights com configuração do componente modo **DEFAULT**

O modelo a seguir executa estas ações:

- Cria uma aplicação do Application Insights. Para obter mais informações, consulte CreateApplication na Referência de API do Amazon CloudWatch Application Insights.
- O componente `my_component` define `ComponentConfigurationMode` como `DEFAULT` e `Tier` como `SQL_SERVER`, o que faz com que esse componente seja configurado com as definições de configuração recomendadas pelo Application Insights para o nível `SQL_Server`. Para obter mais informações, consulte DescribeComponentConfiguration e UpdateComponentConfiguration na Referência de API do Amazon CloudWatch Application Insights.

Modelo em formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "ComponentMonitoringSettings": [
            {
                "ComponentARN": "my_component",
                "Tier": "SQL_SERVER",
                "ComponentConfigurationMode": "DEFAULT"
            }
        ]
    }
}
```

Modelo em formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentARN: my_component
```

```
    Tier: SQL_SERVER
    ComponentConfigurationMode: DEFAULT
```

Criar uma aplicação do Application Insights com configuração do componente modo **`DEFAULT_WITH_OVERWRITE`**

O modelo a seguir executa estas ações:

- Cria uma aplicação do Application Insights. Para obter mais informações, consulte CreateApplication na Referência de API do Amazon CloudWatch Application Insights.
- O componente `my_component` define `ComponentConfigurationMode` como `DEFAULT_WITH_OVERWRITE` e `tier` como `DOT_NET_CORE`, o que faz com que esse componente seja configurado com as definições de configuração recomendadas pelo Application Insights para o nível `DOT_NET_CORE`. As definições de configuração substituídas são especificadas no `DefaultOverwriteComponentConfiguration`:
  - No nível do componente, configurações de `AlarmMetrics` são substituídas.
  - No nível do subcomponente, para os subcomponentes de tipo `EC2_Instance`, as configurações de `Logs` são substituídas.

  Para obter mais informações, consulte UpdateComponentConfiguration na Referência de API do Amazon CloudWatch Application Insights.

Modelo em formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "ComponentMonitoringSettings": [
            {
                "ComponentName": "my_component",
                "Tier": "DOT_NET_CORE",
                "ComponentConfigurationMode": "DEFAULT_WITH_OVERWRITE",
                "DefaultOverwriteComponentConfiguration": {
                    "ConfigurationDetails": {
                        "AlarmMetrics": [
                            {
                                "AlarmMetricName": "StatusCheckFailed"
                            }
```

```
                    ]
                },
                "SubComponentTypeConfigurations": [
                    {
                        "SubComponentType": "EC2_INSTANCE",
                        "SubComponentConfigurationDetails": {
                            "Logs": [
                                {
                                    "LogGroupName": "my_log_group",
                                    "LogPath": "C:\\LogFolder\\*",
                                    "LogType": "IIS",
                                    "Encoding": "utf-8",
                                    "PatternSet": "my_pattern_set"
                                }
                            ]
                        }
                    }
                ]
            }
        }
    ]
}
}
```

Modelo em formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentName: my_component
    Tier: DOT_NET_CORE
    ComponentConfigurationMode: DEFAULT_WITH_OVERWRITE
    DefaultOverwriteComponentConfiguration:
      ConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: StatusCheckFailed
      SubComponentTypeConfigurations:
      - SubComponentType: EC2_INSTANCE
        SubComponentConfigurationDetails:
          Logs:
          - LogGroupName: my_log_group
```

```
            LogPath: C:\LogFolder\*
            LogType: IIS
            Encoding: utf-8
            PatternSet: my_pattern_set
```

Tutorial: como configurar o monitoramento para o SAP ASE

Este tutorial demonstra como configurar o CloudWatch Application Insights para definir o monitoramento dos seus bancos de dados do SAP ASE. Você pode usar os painéis automáticos do CloudWatch Application Insights para visualizar os detalhes do problema, acelerar a solução de problemas e facilitar o tempo médio de resolução (MTTR) para seus bancos de dados do SAP ASE.

Application Insights para tópicos do SAP ASE

- Ambientes compatíveis
- Sistemas operacionais compatíveis
- Atributos
- Pré-requisitos
- Configurar o monitoramento em seu banco de dados do SAP ASE
- Gerenciar o monitoramento do seu banco de dados do SAP ASE
- Configurar o limite de alarmes
- Visualizar e solucionar problemas do SAP ASE detectados pelo Application Insights
- Solução de problemas do Application Insights para SAP ASE

Ambientes compatíveis

O CloudWatch Application Insights é compatível com a implantação de recursos da AWS para os sistemas e os padrões a seguir. Você fornece e instala o software de banco de dados do SAP ASE e o software de aplicação do SAP compatível.

- Um ou mais bancos de dados do SAP ASE em uma única instância do Amazon EC2: SAP ASE em uma arquitetura de aumento vertical de escala de nó único.
- Configuração de alta disponibilidade do banco de dados do SAP ASE entre zonas de disponibilidade: SAP ASE com alta disponibilidade configurada em duas zonas de disponibilidade usando o clustering SUSE/RHEL.

> **Note**
>
> O CloudWatch Application Insights é compatível apenas com ambientes ASE HA com um único ID de sistema SAP (SID). Se vários SIDs do ASE HA estiverem conectados, o monitoramento será configurado apenas para o primeiro SID detectado.

Sistemas operacionais compatíveis

O CloudWatch Application Insights para SAP ASE é compatível com a arquitetura x86-64 nos seguintes sistemas operacionais:

- SuSE Linux 12 SP4
- SuSE Linux 12 SP5
- SUSE Linux 15
- SuSE Linux 15 SP1
- SuSE Linux 15 SP2
- SuSE Linux 15 SP3
- SuSE Linux 15 SP4
- SuSE Linux 15 SP1 para SAP
- SuSE Linux 15 SP2 para SAP
- SuSE Linux 15 SP3 para SAP
- SuSE Linux 15 SP4 para SAP
- SuSE Linux 12 SP4 para SAP
- SuSE Linux 12 SP5 para SAP
- RedHat Linux 7.6
- RedHat Linux 7.7
- RedHat Linux 7.9
- RedHat Linux 8.1
- RedHat Linux 8.4
- RedHat Linux 8.6

Atributos

O CloudWatch Application Insights para SAP ASE oferece os seguintes recursos:

- Detecção automática da workload do SAP ASE
- Criação automática de alarmes do SAP ASE com base em um limite estático
- Criação automática de alarmes do SAP ASE com base na detecção de anomalias
- Reconhecimento automático de padrões de log do SAP ASE
- Painel de controle de integridade do SAP ASE
- Painel de problemas para o SAP ASE

Pré-requisitos

Você deve executar os seguintes pré-requisitos para configurar um banco de dados do SAP ASE com o CloudWatch Application Insights:

- Parâmetros de configuração do SAP ASE: os seguintes parâmetros de configuração devem ser ativados em seu banco de dados ASE: `"enable monitoring"`, `"sql text pipe max messages"`, `"sql text pipe active"`. Isso permite que o CloudWatch Application Insights forneça recursos completos de monitoramento para o seu banco de dados. Se essas configurações não estiverem ativadas no seu banco de dados do ASE, o Application Insights as ativará automaticamente para coletar as métricas necessárias para permitir o monitoramento.
- Usuário do banco de dados do SAP ASE: o usuário do banco de dados fornecido durante a integração do Application Insights deve ter permissão para acessar o seguinte:
  - Tabelas de sistema no banco de dados mestre e nos bancos de dados de usuários (locatários)
  - Tabelas de monitoramento
- SAPHostCtrl: instalar e configurar o SAPHostCtrl em sua instância do Amazon EC2.
- Agente do Amazon CloudWatch: verificar que um agente preexistente do CloudWatch não esteja em execução na sua instância do Amazon EC2. Se você tiver o agente do CloudWatch instalado, certifique-se de remover do seu arquivo de configuração existente a configuração dos recursos que você está usando no CloudWatch Application Insights, para evitar conflito de mesclagem. Para ter mais informações, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

- Habilitação do AWS Systems Manager: instale o agente do SSM em suas instâncias e habilite as instâncias habilitadas para SSM. Para obter mais informações sobre como instalar o SSM Agent, consulte Trabalhar com o SSM Agent no Guia do usuário do AWS Systems Manager.
- Perfis de instância do Amazon EC2: você deve anexar os perfis de instância do Amazon EC2 a seguir para configurar seu banco de dados.
  - Você deve anexar o perfil do `AmazonSSMManagedInstanceCore` para habilitar o Systems Manager. Para obter mais informações, consulte exemplos de políticas baseadas em identidade do AWS Systems Manager.
  - Deve-se anexar o `CloudWatchAgentServerPolicy` para permitir que métricas e logs de instância sejam emitidos por meio do CloudWatch. Para obter mais informações, consulte Criar perfis e usuários do IAM para uso com o agente do Amazon CloudWatch.
  - Você deve anexar a seguinte política em linha do IAM à função de instância do Amazon EC2 para leitura da senha armazenada no AWS Secrets Manager. Para obter mais informações sobre políticas em linha, consulte Políticas em linha no Guia do usuário do AWS Identity and Access Management.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
        }
    ]
}
```

- AWS Resource Groups: você deve criar um grupo de recursos que inclua todos os recursos da AWS associados usados pela sua pilha de aplicações para integrar suas aplicações ao CloudWatch Application Insights. Isso inclui instâncias do Amazon EC2 e volumes do Amazon EBS que executam seu banco de dados do SAP ASE. Se houver vários bancos de dados por conta, recomendamos que você crie um grupo de recursos que inclua os recursos da AWS de cada sistema de banco de dados do SAP ASE.
- Permissões do IAM: para usuários não administradores:

- é necessário criar uma política do AWS Identity and Access Management (IAM) que permita ao Application Insights criar uma função vinculada ao serviço e anexá-la à sua identidade do usuário. Para obter as etapas para anexar a política, consulte Política do IAM.
- O usuário deve ter a permissão para criar um segredo no AWS Secrets Manager para armazenar as credenciais do usuário do banco de dados. Para obter mais informações, consulte Exemplo: permissão para criar segredos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
    }
  ]
}
```

- Perfil vinculado ao serviço: o Application Insights usa os perfis vinculadas ao serviço do AWS Identity and Access Management (IAM). Ao criar a primeira aplicação com o Application Insights, é criada uma função vinculada ao serviço no console do Application Insights. Para ter mais informações, consulte Usar funções vinculadas ao serviço do CloudWatch Application Insights.

Configurar o monitoramento em seu banco de dados do SAP ASE

Use as etapas a seguir para configurar o monitoramento do seu banco de dados do SAP ASE

1. Abra o console do CloudWatch.
2. No painel de navegação esquerdo, em Insights, escolha Application Insights.
3. A página Application Insights exibe a lista de aplicações que são monitoradas pelo Application Insights e o status de monitoramento de cada aplicação. No canto superior direito, escolha Add an application (Adicionar uma aplicação).
4. Na página Especificar detalhes da aplicação, na lista suspensa em Grupo de recursos, selecione o grupo de recursos da AWS que contém os recursos do banco de dados do SAP ASE. Se ainda não tiver criado um grupo de recursos para a aplicação, você poderá criar um escolhendo Create new resource group (Criar grupo de recursos) na lista suspensa Resource group (Grupo

de recursos). Para obter mais informações sobre grupos de recursos, consulte o Guia do usuário dos AWS Resource Groups.

5. Em Monitor CloudWatch Events (Monitorar CloudWatch Events), marque a caixa de seleção para integrar o monitoramento do Application Insights com o CloudWatch Events para obter insights do Amazon EBS, Amazon EC2, AWS CodeDeploy, Amazon ECS, APIs e notificações do AWS Health, Amazon RDS, Amazon S3 e AWS Step Functions.
6. Em Integrar com o OpsCenter do AWS Systems Manager, marque a caixa de seleção ao lado de Gerar OpsItems do OpsCenter do AWS Systems Manager para ações corretivas para visualizar e receber notificações quando forem detectados problemas nas aplicações selecionadas. Para rastrear as operações executadas para resolver itens de trabalho operacionais (OpsItems) relacionados aos recursos da AWS, forneça um ARN do tópico do SNS.
7. É possível, opcionalmente, inserir etiquetas para ajudar a identificar e organizar seus recursos. O CloudWatch Application Insights é compatível com os grupos de recursos baseados em tags e em filas do AWS CloudFormation, exceto os grupos do Application Auto Scaling). Para obter mais informações, consulte Tag Editor (Editor de etiquetas) no Guia do usuário do AWS Resource Groups.
8. Escolha Next (Próximo) para continuar a configurar o monitoramento.
9. Na página Revisar componentes detectados, são listados os componentes monitorados e suas workloads automaticamente detectadas pelo CloudWatch Application Insights.

> **Note**
>
> Os componentes que contêm uma workload detectada de alta disponibilidade do SAP ASE são compatíveis com apenas uma workload em um componente. Os componentes que contêm uma workload de nó único do SAP ASE detectada são compatíveis com várias workloads, mas você não pode adicionar ou remover workloads. Todas as workloads detectadas automaticamente serão monitoradas.

10. Escolha Próximo.
11. Na página Especificar detalhes do componente, insira o nome de usuário e a senha dos bancos de dados do SAP ASE.
12. Revise a configuração de monitoramento de aplicações e escolha Submit (Enviar).
13. A página de detalhes da aplicação se abrirá, e nela serápossível visualizar o Resumo da aplicação, a lista de Componentes e workloads monitoradas e de Componentes e workloads

não monitoradas. Ao selecionar o botão de opção ao lado de um componente ou workload, também será possível visualizar o Histórico de configuração, os Padrões de log e quaisquer Tags criadas. Quando você envia sua configuração, sua conta implementa todas as métricas e alarmes para o sistema do SAP ASE, o que pode levar até duas horas.

Gerenciar o monitoramento do seu banco de dados do SAP ASE

Você pode gerenciar credenciais de usuário, métricas e caminhos de log para o banco de dados do SAP ASE executando as seguintes etapas:

1. Abra o console do CloudWatch.
2. No painel de navegação esquerdo, em Insights, escolha Application Insights.
3. A página Application Insights exibe a lista de aplicações que são monitoradas pelo Application Insights e o status de monitoramento de cada aplicação.
4. Em Componentes monitorados, selecione o botão ao lado do nome do componente. Em seguida, escolha Manage monitoring (Gerenciar o monitoramento).
5. Em EC2 instance group logs (Logs de grupos de instâncias do EC2), é possível atualizar o caminho de log existente, o conjunto de padrões de log e o nome do grupo de logs. Além disso, você pode adicionar até outros três Application logs (Logs de aplicações).
6. Em Métricas, você pode escolher as métricas do SAP ASE de acordo com suas necessidades. Os nomes de métricas do SAP ASE são prefixados com `asedb`. Você pode adicionar até 60 métricas por componente.
7. Em Configuração do ASE, insira o nome de usuário e a senha do banco de dados do SAP ASE. Esse é o nome de usuário e a senha que o agente do Amazon CloudWatch usa para se conectar ao banco de dados do SAP ASE.
8. Em Custom alarms (Alarmes personalizados), é possível pode adicionar alarmes extras para serem monitorados pelo CloudWatch Application Insights.
9. Revise a configuração de monitoramento de aplicações e escolha Submit (Enviar). Quando você envia sua configuração, sua conta atualiza todas as métricas e alarmes do sistema SAP HANA, o que pode levar até duas horas.

Configurar o limite de alarmes

O CloudWatch Application Insights cria automaticamente uma métrica do Amazon CloudWatch para o alarme observar, bem como o limite dessa métrica. O alarme passará para o estado `ALARM`

quando a métrica ultrapassar o limite de um número especificado de períodos de avaliação. Observe que essas configurações não são retidas pelo Application Insights.

Para editar um alarme de uma única métrica, execute as seguintes etapas:

1. Abra o console do CloudWatch.
2. No painel de navegação à esquerda, escolha Alarms (Alarmes), All alarms (Todos os alarmes).
3. Selecione o botão ao lado do alarme criado automaticamente pelo CloudWatch Application Insights. Em seguida, escolha Actions (Ações) e selecione Edit (Editar) no menu suspenso.
4. Edite os seguintes parâmetros em Metric (Métrica).
    a. Em Statistic (Estatística), escolha uma das estatísticas ou um dos percentis predefinidos ou especifique um percentil personalizado. Por exemplo, `p95.45`.
    b. Em Period (Período), escolha os períodos de avaliação do alarme. Ao avaliar o alarme, todos cada período é agregado em um único ponto de dados.
5. Edite os seguintes parâmetros em Conditions (Condições).
    a. Escolha se a métrica deve ser maior que, menor que ou igual ao limite.
    b. Especifique o valor do limite.
6. Em Additional Configuration (Configuração adicional) edite os parâmetros a seguir.
    a. Em Datapoints to alarm (Datapoints para alarme), especifique o número de pontos de dados, ou períodos de avaliação, que devem estar no estado `ALARM` para iniciar o alarme. Quando os dois valores correspondem, é criado um alarme que entra no estado `ALARM` se o número designado de períodos consecutivos for excedido. Para criar um alarme `m` de n, especifique um número menor para o primeiro valor do que para o segundo valor. Para obter mais informações sobre como a avaliação de alarmes, consulte Avaliação de um alarme.
    b. Para o Missing data treatment (Tratamento de dados ausentes), escolha como deseja que o alarme se comporte quando alguns pontos de dados estiverem ausentes. Para obter mais informações, consulte Configurar como alarmes do CloudWatch tratam dados ausentes.
    c. Se o alarme usar um percentil como estatística monitorada, uma caixa Percentiles with low samples (Percentis com amostras baixas) será exibida. Escolha se deseja avaliar ou ignorar casos com taxas de amostra baixas. Se você escolher ignore (maintain the alarm state) (ignorar (manter o estado do alarme)), o estado do alarme atual será sempre mantido quando o tamanho da amostra for muito baixo. Para obter mais informações sobre

percentis com amostras baixas, consulte Alarmes do CloudWatch baseados em percentual e exemplos de poucos dados.

7. Escolha Próximo.
8. Em Notification (Notificação), selecione um tópico do SNS para notificar quando o alarme estiver no estado `ALARM`, `OK` ou `INSUFFICIENT_DATA`.
9. Escolha Create alarm (Criar alarme).

Visualizar e solucionar problemas do SAP ASE detectados pelo Application Insights

Esta seção ajuda você a resolver problemas comuns de solução de problemas que ocorrem quando você configura o monitoramento do SAP ASE no Application Insights.

Erros do servidor de backup do SAP ASE

Você pode identificar a mensagem de erro ao verificar o painel criado dinamicamente. O painel mostra a mensagem de erro relatada no servidor de backup do SAP ASE. Para obter mais detalhes sobre os logs do servidor de backup do SAP ASE, consulte SAP Documentation Backup Server Error Logging.

Transações de longa duração do SAP ASE

Identifique a transação de longa duração e confirme se ela pode ser interrompida ou se o tempo de execução é intencional. Para obter mais detalhes, consulte 2180410 - How to display transaction log records for long running transactions? - SAP ASE.

Conexões de usuário do SAP ASE

Verifique se o banco de dados do SAP ASE está dimensionado adequadamente para a workload que você pretende executar no banco de dados. Para obter mais detalhes, consulte Configuring User Connections na documentação do SAP.

Espaço em disco do SAP ASE

Você pode identificar a camada do banco de dados que está causando o problema verificando o painel criado dinamicamente. O painel mostra as métricas relacionadas e os trechos de arquivos de log. É importante entender a causa do crescimento do disco e, quando aplicável, aumentar o tamanho do disco físico, o espaço em disco alocado ou ambos. Para obter mais detalhes, consulte a Documentação do SAP sobre redimensionamento de disco na documentação do SAP.

Solução de problemas do Application Insights para SAP ASE

Esta seção fornece etapas para ajudar você a resolver erros comuns retornados pelo painel do Application Insights.

Erro	Erro retornado	Causa-raiz	Resolução
Não é possível adicionar mais de 60 métricas de monitoramento.	`Component cannot have more than 60 monitored metric`	O limite atual de métricas é de 60 métricas monitoradas por componente.	Remova métricas desnecessárias para aderir ao limite.
Nenhuma métrica ou alarme do SAP aparece após o processo de integração	O comando `run` no `AWS-ConfigureAWSPackage` falhou no AWS Systems Manager. A saída mostra o erro: `CT-LIBRARY error:ct_connect(): protocol specific layer: external error: The attempt to connect to the server failed`	O nome de usuário e a senha podem estar incorretos.	Verifique se o nome de usuário e a senha são válidos e, em seguida, execute novamente o processo de integração.

Tutorial: Configurar o monitoramento para SAP HANA

Este tutorial demonstra como configurar o CloudWatch Application Insights para estabelecer o monitoramento para seus bancos de dados SAP HANA. Você pode usar os painéis automáticos do CloudWatch Application Insights para visualizar detalhes do problema, acelerar a solução de problemas e facilitar o tempo médio de resolução (MTTR) para seus bancos de dados SAP HANA.

Tópicos do Application Insights para SAP HANA

- Ambientes compatíveis

Ambientes compatíveis

O CloudWatch Application Insights oferece suporte à implantação de recursos da AWS para os sistemas e padrões a seguir. Você fornece e instala o software de banco de dados SAP HANA e o software de aplicação SAP compatível.

- Banco de dados SAP HANA em uma única instância do Amazon EC2: SAP HANA em uma arquitetura de aumento na escala vertical em nó único, com até 24 TB de memória.
- Banco de dados SAP HANA em várias instâncias do Amazon EC2: SAP HANA em uma arquitetura de redução da escala em vários nós.
- Configuração de alta disponibilidade do banco de dados SAP HANA Cross-AZ: SAP HANA com alta disponibilidade configurada em duas zonas de disponibilidade usando clustering SUSE/RHEL.

> **Note**
>
> O CloudWatch Application Insights oferece suporte apenas a ambientes SID HANA únicos. Se houver vários SID HANA conectados, o monitoramento será configurado apenas para o primeiro SID detectado.

Sistemas operacionais compatíveis

O CloudWatch Application Insights para SAP HANA é compatível com a arquitetura x86-64 nos seguintes sistemas operacionais:

- SuSE Linux 12 SP4 para SAP

- SuSE Linux 12 SP5 para SAP
- SUSE Linux 15
- SuSE Linux 15 SP1
- SuSE Linux 15 SP2
- SuSE Linux 15 para SAP
- SuSE Linux 15 SP1 para SAP
- SuSE Linux 15 SP2 para SAP
- SuSE Linux 15 SP3 para SAP
- SuSE Linux 15 SP4 para SAP
- SuSE Linux 15 SP5 para SAP
- RedHat Linux 8.6 para SAP com alta disponibilidade e serviços de atualização
- RedHat Linux 8.5 para SAP com alta disponibilidade e serviços de atualização
- RedHat Linux 8.4 para SAP com alta disponibilidade e serviços de atualização
- RedHat Linux 8.3 para SAP com alta disponibilidade e serviços de atualização
- RedHat Linux 8.2 para SAP com alta disponibilidade e serviços de atualização
- RedHat Linux 8.1 para SAP com alta disponibilidade e serviços de atualização
- RedHat Linux 7.9 para SAP com alta disponibilidade e serviços de atualização

Atributos

O CloudWatch Application Insights para SAP HANA fornece os seguintes recursos:

- Detecção automática de workload SAP HANA
- Criação automática de alarmes SAP HANA com base no limite estático
- Criação automática de alarmes SAP HANA com base na detecção de anomalias
- Reconhecimento automático de padrões de log SAP HANA
- Painel de integridade para o SAP HANA
- Painel de problemas para o SAP HANA

Pré-requisitos

É necessário cumprir os seguintes pré-requisitos para configurar um banco de dados SAP HANA com o CloudWatch Application Insights:

- SAP HANA: instalar um banco de dados SAP HANA 2.0 SPS05 que esteja em execução e seja acessível em uma instância do Amazon EC2.
- Usuário do banco de dados SAP HANA: um usuário de banco de dados com perfis de monitoramento deve ser criado no banco de dados SYSTEM e em todos os locatários.

  Exemplo

  Os comandos SQL a seguir criam um usuário com funções de monitoramento.

```
su - <sid>adm
hdbsql -u SYSTEM -p <SYSTEMDB password> -d SYSTEMDB
CREATE USER CW_HANADB_EXPORTER_USER PASSWORD <Monitoring user password> NO
 FORCE_FIRST_PASSWORD_CHANGE;
CREATE ROLE CW_HANADB_EXPORTER_ROLE;
GRANT MONITORING TO CW_HANADB_EXPORTER_ROLE;
GRANT CW_HANADB_EXPORTER_ROLE TO CW_HANADB_EXPORTER_USER;
```

- Python 3.8: instalar a versão Python 3.8 ou versões posteriores em seu sistema operacional. Use a versão mais recente do Python. Se o Python3 não for detectado em seu sistema operacional, a versão Python 3.6 será instalada.

  Para obter mais informações, consulte installation example.

  > **Note**
  >
  > A instalação manual da versão Python 3.8 ou de versões posteriores é obrigatória para os sistemas operacionais SuSE Linux 15 SP4, RedHat Linux 8.6 e versões posteriores.

- Pip3: instalar o programa instalador, pip3, em seu sistema operacional. Se o pip3 não for detectado em seu sistema operacional, ele será instalado.
- hdbclient: o CloudWatch Application Insights usa o driver Python para efetuar a conexão com o banco de dados SAP HANA. Se o cliente não tiver realizado a instalação usando python3, certifique-se de ter o arquivo hdbclient em tar na versão `2.10 or later` em `/hana/shared/SID/hdbclient/`.
- Agente do Amazon CloudWatch: verificar que um agente preexistente do CloudWatch não esteja em execução na sua instância do Amazon EC2. Se você tiver o agente do CloudWatch instalado, certifique-se de remover do seu arquivo de configuração existente a configuração dos recursos que você está usando no CloudWatch Application Insights, para evitar conflito de mesclagem.

Para ter mais informações, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

- Habilitação do AWS Systems Manager: instalar o SSM Agent em suas instâncias, que deverão estar habilitadas para o SSM. Para obter informações sobre como instalar o SSM Agent, consulte Trabalhar com o SSM Agent no Guia do usuário do AWS Systems Manager.
- Perfis de instância do Amazon EC2: você deve anexar os perfis de instância do Amazon EC2 a seguir para configurar seu banco de dados.
  - Você deve anexar o perfil do `AmazonSSMManagedInstanceCore` para habilitar o Systems Manager. Para obter mais informações, consulte exemplos de políticas baseadas em identidade do AWS Systems Manager.
  - Deve-se anexar o `CloudWatchAgentServerPolicy` para permitir que métricas e logs de instância sejam emitidos por meio do CloudWatch. Para obter mais informações, consulte Criar perfis e usuários do IAM para uso com o agente do CloudWatch.
  - Você deve anexar a seguinte política em linha do IAM à função de instância do Amazon EC2 para leitura da senha armazenada no AWS Secrets Manager. Para obter mais informações sobre políticas em linha, consulte Políticas em linha no Guia do usuário do AWS Identity and Access Management.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
        }
    ]
}
```

- Grupos de recursos da AWS: você deve criar um grupo de recursos que inclua todos os recursos associados da AWS usados pela pilha de aplicações para integrar suas aplicações ao CloudWatch Application Insights. Isso inclui instâncias do Amazon EC2 e volumes do Amazon EBS que executam seu banco de dados SAP HANA. Se houver vários bancos de dados por conta, recomendamos criar um grupo de recursos que inclua os recursos da AWS para cada sistema de banco de dados SAP HANA.

- Permissões do IAM: para usuários não administradores:
  - é necessário criar uma política do AWS Identity and Access Management (IAM) que permita ao Application Insights criar uma função vinculada ao serviço e anexá-la à sua identidade do usuário. Para obter as etapas para anexar a política, consulte Política do IAM.
  - O usuário deve ter a permissão para criar um segredo no AWS Secrets Manager para armazenar as credenciais do usuário do banco de dados. Para obter mais informações, consulte Exemplo: permissão para criar segredos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
    }
  ]
}
```

- Perfil vinculado ao serviço: o Application Insights usa os perfis vinculadas ao serviço do AWS Identity and Access Management (IAM). Ao criar a primeira aplicação com o Application Insights, é criada uma função vinculada ao serviço no console do Application Insights. Para ter mais informações, consulte Usar funções vinculadas ao serviço do CloudWatch Application Insights.

Configurar seu banco de dados SAP HANA para monitoramento

Use as etapas a seguir para configurar o monitoramento para seu banco de dados SAP HANA

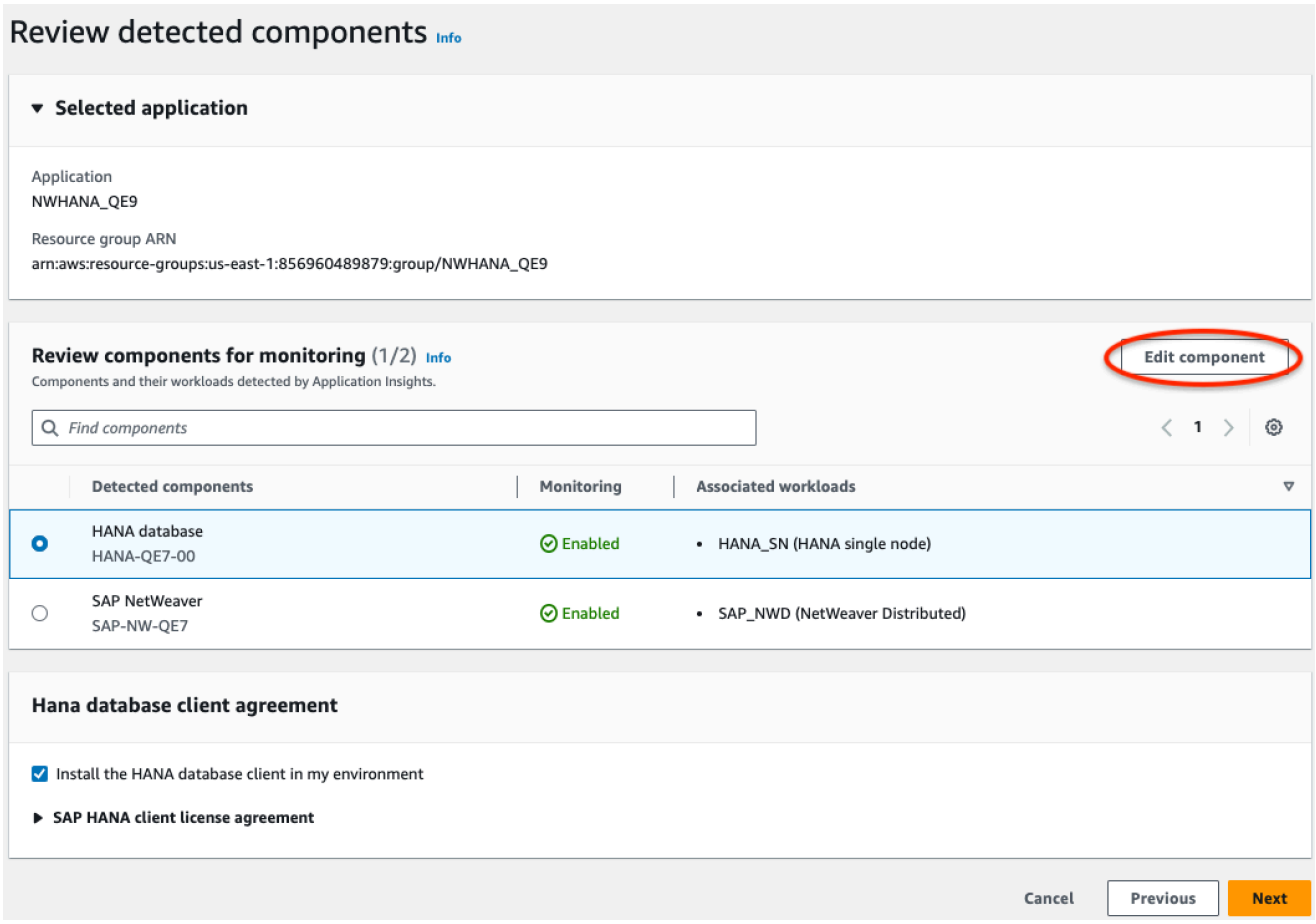
1. Abra o console do CloudWatch.
2. No painel de navegação esquerdo, em Insights, escolha Application Insights.
3. A página Application Insights exibe a lista de aplicações que são monitoradas pelo Application Insights e o status de monitoramento de cada aplicação. No canto superior direito, escolha Add an application (Adicionar uma aplicação).
4. Na página Specify application details (Especificar os detalhes da aplicação), na lista suspensa em Resource group (Grupo de recursos), selecione o grupo de recursos da AWS que contém seus recursos de banco de dados SAP HANA. Se ainda não tiver criado um grupo de recursos

para a aplicação, você poderá criar um escolhendo Create new resource group (Criar grupo de recursos) na lista suspensa Resource group (Grupo de recursos). Para obter mais informações sobre grupos de recursos, consulte o Guia do usuário dos AWS Resource Groups.

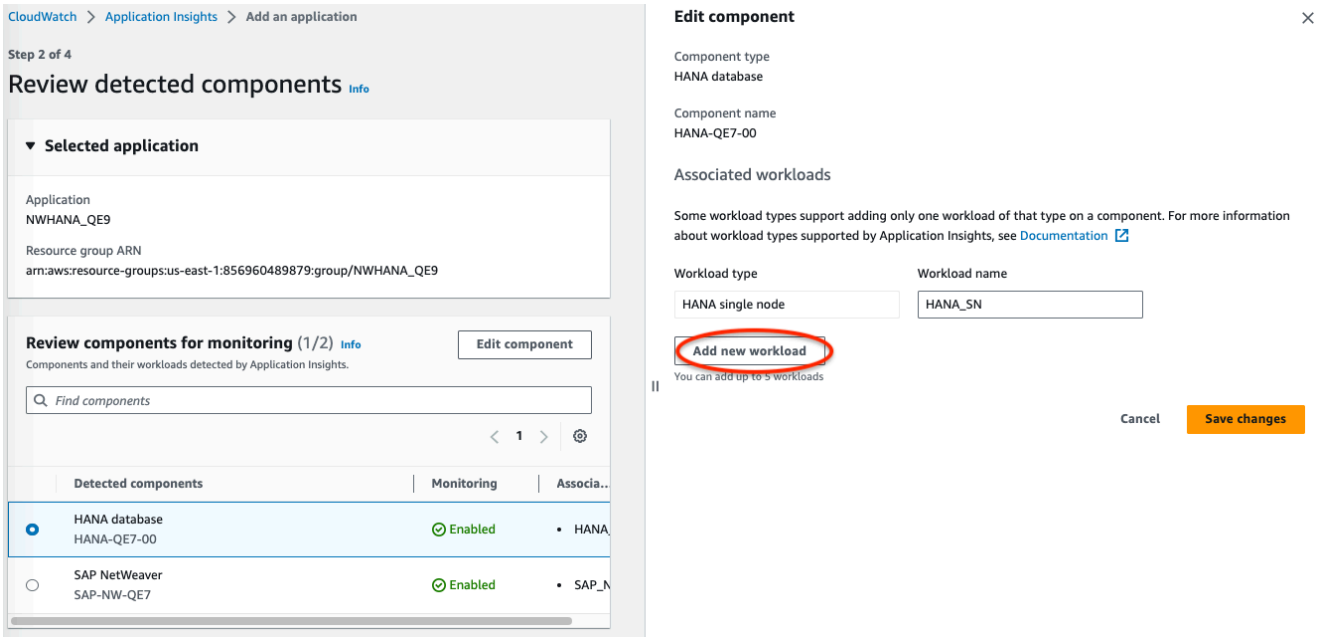
5. Em Monitor CloudWatch Events (Monitorar CloudWatch Events), marque a caixa de seleção para integrar o monitoramento do Application Insights com o CloudWatch Events para obter insights do Amazon EBS, Amazon EC2, AWS CodeDeploy, Amazon ECS, APIs e notificações do AWS Health, Amazon RDS, Amazon S3 e AWS Step Functions.
6. Em Integrar com o OpsCenter do AWS Systems Manager, marque a caixa de seleção ao lado de Gerar OpsItems do OpsCenter do AWS Systems Manager para ações corretivas para visualizar e receber notificações quando forem detectados problemas nas aplicações selecionadas. Para rastrear as operações executadas para resolver itens de trabalho operacionais (OpsItems) relacionados aos recursos da AWS, forneça um ARN do tópico do SNS.
7. É possível, opcionalmente, inserir etiquetas para ajudar a identificar e organizar seus recursos. O CloudWatch Application Insights é compatível com os grupos de recursos baseados em tags e em filas do AWS CloudFormation, exceto os grupos do Application Auto Scaling). Para obter mais informações, consulte Tag Editor (Editor de etiquetas) no Guia do usuário do AWS Resource Groups.
8. Escolha Next (Próximo) para continuar a configurar o monitoramento.
9. Na página Revisar componentes detectados, são listados os componentes monitorados e suas workloads automaticamente detectadas pelo CloudWatch Application Insights.

    a. Para adicionar workloads a um componente que contenha uma workload de nó único detectada do SAP HANA, selecione o componente e escolha Editar componente.

> **Note**
>
> Os componentes que contêm uma workload detectada de vários nós do SAP HANA ou de alta disponibilidade do HANA oferecem suporte a apenas uma workload em um componente.



b. Para adicionar uma nova workload, escolha Adicionar nova workload.



c. Quando terminar de editar workloads, escolha Salvar alterações.

10. Escolha Próximo.
11. Na página Especificar detalhes do componente, insira o seu nome e senha de usuário.
12. Revise a configuração de monitoramento de aplicações e escolha Submit (Enviar).
13. A página de detalhes da aplicação se abrirá, e nela serápossível visualizar o Resumo da aplicação, a lista de Componentes e workloads monitoradas e de Componentes e workloads não monitoradas. Ao selecionar o botão de opção ao lado de um componente ou workload, também será possível visualizar o Histórico de configuração, os Padrões de log e quaisquer Tags criadas. Quando você envia sua configuração, sua conta implanta todas as métricas e alarmes para o sistema SAP HANA, o que pode levar até duas horas.

Gerenciar o monitoramento de seu banco de dados SAP HANA

Você pode gerenciar credenciais de usuário, métricas e caminhos de log para o banco de dados SAP HANA executando as seguintes etapas:

1. Abra o console do CloudWatch.
2. No painel de navegação esquerdo, em Insights, escolha Application Insights.
3. A página Application Insights exibe a lista de aplicações que são monitoradas pelo Application Insights e o status de monitoramento de cada aplicação.
4. Em Componentes monitorados, selecione o botão ao lado do nome do componente. Em seguida, escolha Manage monitoring (Gerenciar o monitoramento).
5. Em EC2 instance group logs (Logs de grupos de instâncias do EC2), é possível atualizar o caminho de log existente, o conjunto de padrões de log e o nome do grupo de logs. Além disso, você pode adicionar até outros três Application logs (Logs de aplicações).
6. Em Metrics (Métricas), você pode escolher as métricas SAP HANA de acordo com suas necessidades. Os nomes das métricas SAP HANA são prefixados com `hanadb`. Você pode adicionar até 40 métricas por componente.
7. Em HANA configuration (Configuração do HANA), insira a senha e o nome de usuário para o banco de dados SAP HANA. Este é o nome de usuário e a senha que o agente do Amazon CloudWatch usa para efetuar a conexão com o banco de dados SAP HANA.
8. Em Custom alarms (Alarmes personalizados), é possível pode adicionar alarmes extras para serem monitorados pelo CloudWatch Application Insights.

9. Revise a configuração de monitoramento de aplicações e escolha Submit (Enviar). Quando você envia sua configuração, sua conta atualiza todas as métricas e alarmes do sistema SAP HANA, o que pode levar até duas horas.

Visualizar e solucionar problemas detectados pelo CloudWatch Application Insights

As seções a seguir fornecem etapas para ajudar a resolver cenários comuns de solução de problemas que ocorrem quando você configura o monitoramento para SAP HANA no Application Insights.

Tópicos de solução de problemas

- Banco de dados SAP HANA atinge o limite de alocação de memória
- Evento de disco cheio
- O backup SAP HANA parou de ser executado

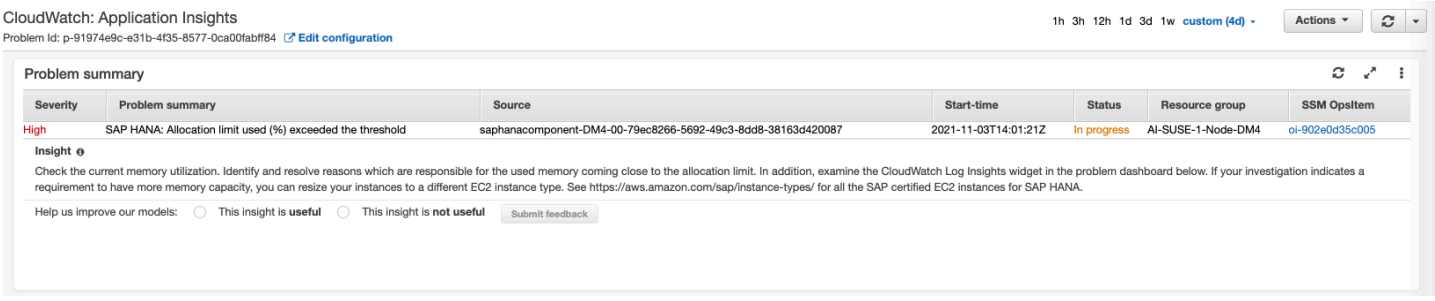
Banco de dados SAP HANA atinge o limite de alocação de memória

Descrição

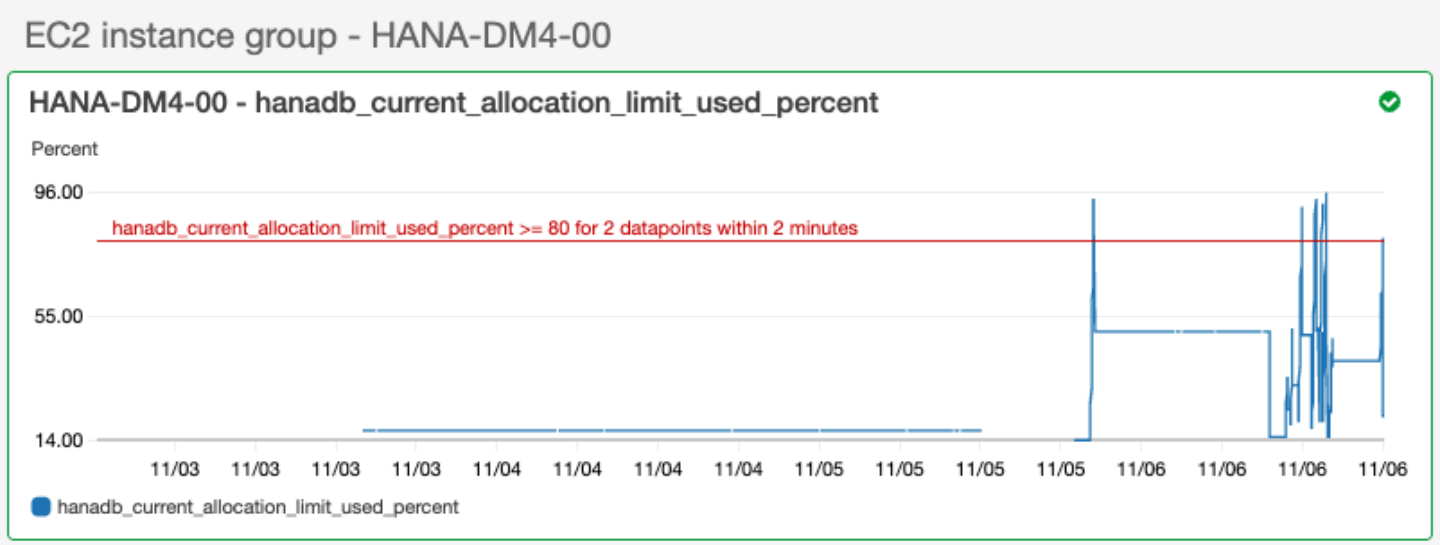
Sua aplicação SAP que é apoiado por um banco de dados SAP HANA apresenta um funcionamento defeituoso devido à alta pressão de memória, levando à degradação da performance da aplicação.

Resolução

É possível identificar a camada da aplicação que está causando o problema conferindo o painel criado dinamicamente que mostra as métricas relacionadas e os trechos do arquivo de log. No exemplo a seguir, o problema pode ocorrer devido a uma grande carga de dados no sistema SAP HANA.



A alocação de memória usada excede o limite de 80% do limite total de alocação de memória.



O grupo de logs mostra o esquema `BNR-DATA` e a tabela `IMDBMASTER_30003` ficou sem memória. Além disso, o grupo de logs mostra a hora exata do problema, o limite de localização global atual, a memória compartilhada, o tamanho do código e o tamanho da alocação de reserva OOM.

Log Group: SAP_HANA_TRACE-AI-SUSE-1-Node-DM4, Log Type: SAP_HANA_TRACE, AWS::SAPHANA.OutOfMemory

#	@timestamp	@message
1	2021-11-06T13:31:23.317Z	GLOBAL_ALLOCATION_LIMIT (GAL) = 55.78gb (59901001728b), SHARED_MEMORY = 567.77mb (595357696b), CODE_SIZE = 2.94gb (3162550272b), OOM_RESERVATION_ALLOCATOR_SIZE = 96.14mb (100810752b)
2	2021-11-06T13:31:23.316Z	[2867]{311260}[22/963854] 2021-11-06 13:00:44.999570 e OOM_Notification Statement.cc(04580) : oom exception occurred at 'imdbmaster:30003': conn_id=311260, stmt_id=1336853818011966, stmt_hash=17e1ccc2b5f460604ceae8c98690fd01, sql=CAL…
3	2021-11-06T13:31:23.316Z	[3033]{311513}[22/967162] 2021-11-06 13:31:17.163640 e Memory mmReportMemoryProblems.cpp(01805) : OUT OF MEMORY occurred.
4	2021-11-06T13:31:23.316Z	Current callstack: 1: 0x00007f824538dd35 in MemoryManager::PoolAllocator::notifyOOMImpl(unsigned long, unsigned long, bool, ltt::allocation_failure_type, bool)+0x1b1 at mmPoolAllocator.cpp:2284 (libhdbbasis.so) 2: 0x00007f824524a7ad …
5	2021-11-06T13:31:23.316Z	[2822]{-1}[-1/-1] 2021-11-06 13:31:17.175597 e Memory mmReportMemoryProblems.cpp(01805) : OUT OF MEMORY occurred.
6	2021-11-06T13:31:23.316Z	Current callstack: 1: 0x00007f824538dd35 in MemoryManager::PoolAllocator::notifyOOMImpl(unsigned long, unsigned long, bool, ltt::allocation_failure_type, bool)+0x1b1 at mmPoolAllocator.cpp:2284 (libhdbbasis.so) 2: 0x00007f824524a7ad …
7	2021-11-06T13:31:23.316Z	GLOBAL_ALLOCATION_LIMIT (GAL) = 55.78gb (59901001728b), SHARED_MEMORY = 567.77mb (595357696b), CODE_SIZE = 2.94gb (3162550272b), OOM_RESERVATION_ALLOCATOR_SIZE = 96.14mb (100810752b)
8	2021-11-06T13:31:17.318Z	GLOBAL_ALLOCATION_LIMIT (GAL) = 55.78gb (59901001728b), SHARED_MEMORY = 567.77mb (595357696b), CODE_SIZE = 2.94gb (3162550272b), OOM_RESERVATION_ALLOCATOR_SIZE = 96.14mb (100810752b)
9	2021-11-06T13:31:17.317Z	GLOBAL_ALLOCATION_LIMIT (GAL) = 55.78gb (59901001728b), SHARED_MEMORY = 567.77mb (595357696b), CODE_SIZE = 2.94gb (3162550272b), OOM_RESERVATION_ALLOCATOR_SIZE = 96.14mb (100810752b)
10	2021-11-06T13:31:17.317Z	[3033]{311513}[22/967162] 2021-11-06 13:31:17.100223 w Memory mmPoolAllocator.cpp(01212) : Out of memory for Pool/PersistenceManager/PersistentSpace/DefaultLPA/DataPage, size 167772168, alignment=40968, flags 0x0, reason GLOBAL_ALLOC…
11	2021-11-06T13:31:17.317Z	GLOBAL_ALLOCATION_LIMIT (GAL) = 55.78gb (59901001728b), SHARED_MEMORY = 567.77mb (595357696b), CODE_SIZE = 2.94gb (3162550272b), OOM_RESERVATION_ALLOCATOR_SIZE = 96.14mb (100810752b)
12	2021-11-06T13:31:17.317Z	[3033]{311513}[22/967162] 2021-11-06 13:31:17.163640 e Memory mmReportMemoryProblems.cpp(01805) : OUT OF MEMORY occurred.
13	2021-11-06T13:31:17.317Z	Current callstack: 1: 0x00007f824538dd35 in MemoryManager::PoolAllocator::notifyOOMImpl(unsigned long, unsigned long, bool, ltt::allocation_failure_type, bool)+0x1b1 at mmPoolAllocator.cpp:2284 (libhdbbasis.so) 2: 0x00007f824524a7ad …
14	2021-11-06T13:31:17.317Z	[2822]{-1}[-1/-1] 2021-11-06 13:31:17.170707 w Memory mmPoolAllocator.cpp(01212) : Out of memory for Pool/malloc/libhdbbasement.so, size 422808, alignment=88, flags 0x0, reason GLOBAL_ALLOCATION_LIMIT
15	2021-11-06T13:31:17.317Z	[2822]{-1}[-1/-1] 2021-11-06 13:31:17.175597 e Memory mmReportMemoryProblems.cpp(01805) : OUT OF MEMORY occurred.
16	2021-11-06T13:31:17.317Z	Current callstack: 1: 0x00007f824538dd35 in MemoryManager::PoolAllocator::notifyOOMImpl(unsigned long, unsigned long, bool, ltt::allocation_failure_type, bool)+0x1b1 at mmPoolAllocator.cpp:2284 (libhdbbasis.so) 2: 0x00007f824524a7ad …
17	2021-11-06T13:31:17.317Z	GLOBAL_ALLOCATION_LIMIT (GAL) = 55.78gb (59901001728b), SHARED_MEMORY = 567.77mb (595357696b), CODE_SIZE = 2.94gb (3162550272b), OOM_RESERVATION_ALLOCATOR_SIZE = 96.14mb (100810752b)
18	2021-11-06T13:31:16.317Z	GLOBAL_ALLOCATION_LIMIT (GAL) = 55.78gb (59901001728b), SHARED_MEMORY = 567.77mb (595357696b), CODE_SIZE = 2.94gb (3162550272b), OOM_RESERVATION_ALLOCATOR_SIZE = 96.14mb (100810752b)
19	2021-11-06T13:31:16.317Z	GLOBAL_ALLOCATION_LIMIT (GAL) = 55.78gb (59901001728b), SHARED_MEMORY = 567.77mb (595357696b), CODE_SIZE = 2.94gb (3162550272b), OOM_RESERVATION_ALLOCATOR_SIZE = 96.14mb (100810752b)

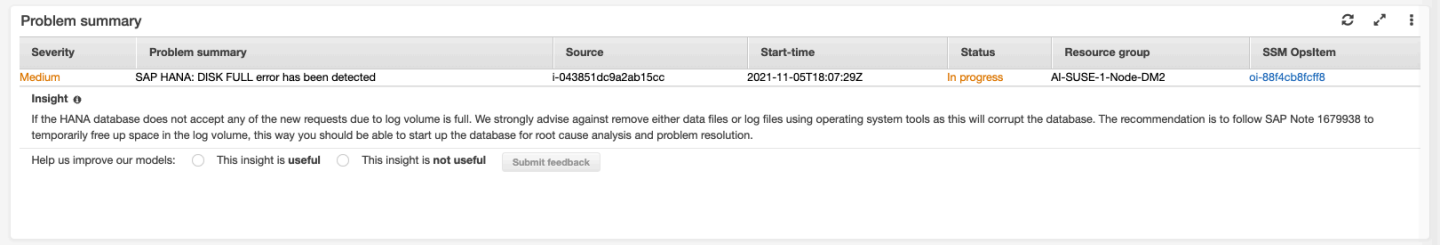
Evento de disco cheio

Descrição

Sua aplicação SAP que é apoiada por um banco de dados SAP HANA deixa de responder, o que leva a uma incapacidade de acessar o banco de dados.

Resolução

É possível identificar a camada do banco de dados que está causando o problema conferindo o painel criado dinamicamente que mostra as métricas relacionadas e os trechos do arquivo de log. No exemplo a seguir, o problema pode ter sido causado por falha do administrador ao habilitar o backup automático de log, o que fez com que o diretório sap/hana/log fosse preenchido.



O widget do grupo de log no painel de problemas exibe o evento `DISKFULL`.

O backup SAP HANA parou de ser executado

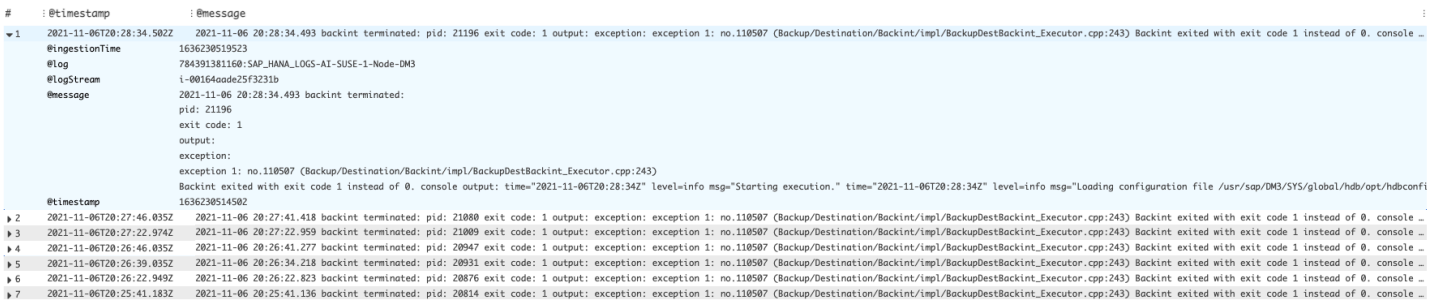
Descrição

Sua aplicação SAP com suporte de um banco de dados SAP HANA parou de funcionar.

Resolução

É possível identificar a camada do banco de dados que está causando o problema conferindo o painel criado dinamicamente que mostra as métricas relacionadas e os trechos do arquivo de log.

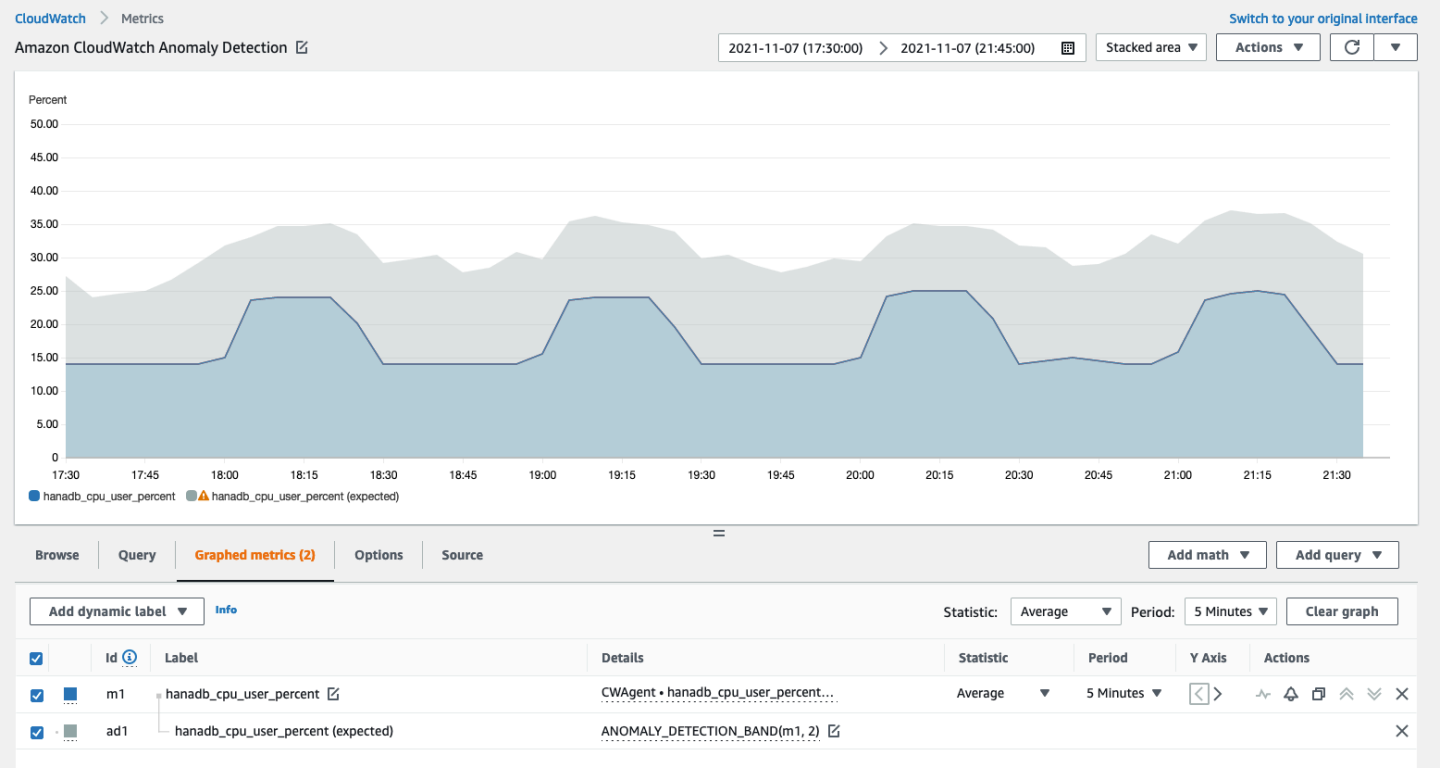
O widget do grupo de log no painel de problemas exibe o evento `ACCESS DENIED`. Isso inclui informações adicionais, como o bucket do S3, a pasta do bucket do S3 e a região do bucket do S3.



Detecção de anomalias para o SAP HANA

Para métricas específicas do SAP HANA, como o número de contagens de threads, o CloudWatch aplica algoritmos estatísticos e de machine learning para definir o limite. Esses algoritmos analisam continuamente métricas do banco de dados SAP HANA, determinam linhas de base normais e apontam anomalias com intervenção mínima do usuário. Os algoritmos geram um modelo de detecção de anomalia, o que gera um intervalo de valores esperados que representam o comportamento normal da métrica.

Os algoritmos de detecção de anomalias consideram a sazonalidade e as mudanças de tendência das métricas. As mudanças de sazonalidade podem ser por hora, dia ou semana, conforme mostrado nos exemplos de uso da CPU do SAP HANA.



Depois de criar um modelo, a detecção de anomalias do CloudWatch avaliará continuamente o modelo e fará ajustes para garantir que ele seja o mais preciso possível. Isso inclui treinar novamente o modelo para ajustes caso os valores da métrica evoluam ao longo do tempo ou sofram mudanças repentinas. Também inclui preditores para melhorar os modelos de métricas sazonais, pontiagudas ou esparsas.

Solução de problemas do Application Insights para o SAP HANA

Esta seção fornece etapas para ajudar você a resolver erros comuns retornados pelo painel do Application Insights.

Não foi possível adicionar mais de 60 métricas monitoradas

A saída mostra o erro apresentado a seguir.

```
Component cannot have more than 60 monitored metrics
```

Principal causa: no momento, o limite de métricas é de 60 métricas monitoradas por componente.

Resolução: para permanecer abaixo do limite, remova as métricas que não são necessárias.

Nenhuma métrica SAP é exibida após o processo de integração

Use as informações apresentadas a seguir para descobrir o motivo pelo qual as métricas SAP não são exibidas no painel após o processo de integração. A primeira etapa é solucionar problemas relacionados às métricas SAP que não estão sendo exibidas usando o AWS Management Console ou os logs do exportador de uma instância do Amazon EC2. Em seguida, faça uma análise da saída do erro para encontrar uma solução.

Como solucionar problemas relacionados às métricas SAP que não são exibidas após a integração

É possível usar o AWS Management Console ou os logs do exportador de uma instância do Amazon EC2 para solucionar problemas.

AWS Management Console

Como solucionar problemas relacionados às métricas SAP que não são exibidas após a integração usando o console

1. Abra o console AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação à esquerda, escolha Gerenciador de estados.
3. Em Associações, verifique o status do documento `AWSEC2-ApplicationInsightsCloudwatchAgentInstallAndConfigure`. Se o status for `Failed`, em ID da execução, selecione o ID com falha e visualize a saída.
4. Em Associações, verifique o status do documento `AWS-ConfigureAWSPackage`. Se o status for `Failed`, em ID da execução, selecione o ID com falha e visualize a saída.

Exporter logs from Amazon EC2 instance

Como solucionar problemas relacionados às métricas SAP que não são exibidas após a integração usando os logs do exportador

1. Conecte-se à instância do Amazon EC2 na qual seu banco de dados SAP HANA está em execução.
2. Encontre a convenção de nomenclatura mais adequada para `WORKLOAD_SHORT_NAME` usando o comando apresentado a seguir. Você usará esse nome abreviado nas duas etapas a seguir.

```
sudo systemctl | grep exporter
```

> **Note**
>
> O Application Insights adiciona um sufixo, `WORKLOAD_SHORT_NAME`, ao nome do serviço, dependendo da workload que está em execução. Os nomes abreviados para nó único, múltiplos nós e implantações de alta disponibilidade do SAP HANA são `HANA_SN`, `HANA_MN` e `HANA_HA`.

3. Para verificar se há erros nos logs de serviço do gerenciador do exportador, execute o comando apresentado a seguir, substituindo `WORKLOAD_SHORT_NAME` pelo nome abreviado encontrado na Step 2.

```
sudo journalctl -e --unit=prometheus-
hanadb_exporter_manager_WORKLOAD_SHORT_NAME.service
```

4. Se os logs de serviço do gerenciador do exportador não mostrarem um erro, verifique se há erros nos logs do serviço do exportador ao executar o comando apresentado a seguir.

```
sudo journalctl -e --unit=prometheus-hanadb_exporter_WORKLOAD_SHORT_NAME.service
```

Como resolver as principais causas conhecidas do problema de métricas SAP que não são exibidas após a integração

Os exemplos apresentados a seguir descrevem como resolver as principais causas conhecidas de métricas SAP que não são exibidas após a integração.

- A saída mostra o erro apresentado a seguir.

```
Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-
cloudwatch-agent.d/default ...
Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/
amazon-cloudwatch-agent.d/ssm_AmazonCloudWatch-ApplicationInsights-
SSMParameterForTESTCWEC2INSTANCEi0d88867f1f3e36285.tmp ...
2023/11/30 22:25:17 Failed to merge multiple json config files.
2023/11/30 22:25:17 Failed to merge multiple json config files.
2023/11/30 22:25:17 Under path : /metrics/append_dimensions | Error : Different
 values are specified for append_dimensions
2023/11/30 22:25:17 Under path : /metrics/metrics_collected/disk | Error : Different
 values are specified for disk
2023/11/30 22:25:17 Under path : /metrics/metrics_collected/mem | Error : Different
 values are specified for mem
2023/11/30 22:25:17 Configuration validation first phase failed. Agent version: 1.0.
 Verify the JSON input is only using features supported by this version.
```

  Resolução: o Application Insights está tentando configurar métricas semelhantes que foram configuradas previamente como parte do arquivo de configuração existente do agente do CloudWatch. Remova os arquivos existentes em `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/` ou remova as métricas que estão causando o conflito para o arquivo de configuração existente do agente do CloudWatch.

- A saída mostra o erro apresentado a seguir.

```
Unable to find a host with system database, for more info rerun using -v
```

  Resolução: o nome de usuário, a senha ou a porta do banco de dados podem estar incorretos. Verifique se o nome de usuário, a senha e a porta são válidos e, em seguida, execute novamente o processo de integração.

- A saída mostra o erro apresentado a seguir.

```
This hdbcli installer is not compatible with your Python interpreter
```

  Resolução: atualize o instalador pip3 e o formato wheel, conforme mostrado no exemplo a seguir, para a versão Python 3.6.

```
python3.6 -m pip install --upgrade pip setuptools wheel
```

- A saída mostra o erro apresentado a seguir.

```
Unable to install hdbcli using pip3. Please try to install it
```

  Resolução: certifique-se de ter seguido os pré-requisitos do `hdbclient` ou instale o `hdbclient` manualmente no pip3.

- A saída mostra o erro apresentado a seguir.

```
Package 'boto3' requires a different Python: 3.6.15 not in '>= 3.7'
```

  Resolução: a versão Python 3.8 ou versões posteriores são necessárias para usar essa versão do sistema operacional. Verifique os pré-requisitos da versão Python 3.8 e instale-a.

- A saída mostra um dos erros de instalação apresentados a seguir.

```
Can not execute `setup.py` since setuptools is not available in the build environment
```

  ou

```
[SSL: CERTIFICATE_VERIFY_FAILED]
```

  Resolução: instale o Python usando comandos do SUSE Linux, conforme mostrado no exemplo a seguir. O exemplo apresentado a seguir realiza a instalação da versão mais recente do Python 3.8.

```
wget https://www.python.org/ftp/python/3.8.<LATEST_RELEASE>/
Python-3.8.<LATEST_RELEASE>.tgz
tar xf Python-3.*
cd Python-3.*/
sudo zypper install make gcc-c++ gcc automake autoconf libtool
sudo zypper install zlib-devel
sudo zypper install libopenssl-devel libffi-devel
./configure --with-ensurepip=install
sudo make
sudo make install
sudo su
python3.8 -m pip install --upgrade pip setuptools wheel
```

Tutorial: Configurar monitoramento para o SAP NetWeaver

Este tutorial demonstra como configurar o Amazon CloudWatch Application Insights para estabelecer o monitoramento para o SAP NetWeaver. Você pode usar os painéis automáticos do CloudWatch Application Insights para visualizar detalhes dos problemas, acelerar a solução dos problemas e reduzir o tempo médio de resolução (MTTR) para os servidores de aplicações do SAP NetWeaver.

Tópicos do CloudWatch Application Insights para o SAP NetWeaver

Ambientes compatíveis

O CloudWatch Application Insights é compatível com a implantação de recursos da AWS para os sistemas e os padrões a seguir.

- Implantação do sistema SAP NetWeaver Standard.
- Implantações distribuídas do SAP NetWeaver em várias instâncias do Amazon EC2.
- Configuração da alta disponibilidade do SAP NetWeaver Cross-AZ: SAP NetWeaver com alta disponibilidade configurado em duas zonas de disponibilidade usando clustering SUSE/RHEL.

Sistemas operacionais compatíveis

O CloudWatch Application Insights para SAP NetWeaver é compatível com os seguintes sistemas operacionais:

- Oracle Linux 8
- Red Hat Enterprise Linux 7.6

- Red Hat Enterprise Linux 7.7
- Red Hat Enterprise Linux 7.9
- Red Hat Enterprise Linux 8.1
- Red Hat Enterprise Linux 8,2
- Red Hat Enterprise Linux 8.4
- Red Hat Enterprise Linux 8,6
- SUSE Linux Enterprise Server 15 for SAP
- SUSE Linux Enterprise Server 15 SP1 for SAP
- SUSE Linux Enterprise Server 15 SP2 for SAP
- SUSE Linux Enterprise Server 15 SP3 for SAP
- SUSE Linux Enterprise Server 15 SP4 for SAP
- SUSE Linux Enterprise Server 12 SP4 for SAP
- SUSE Linux Enterprise Server 12 SP5 for SAP
- SUSE Linux Enterprise Server 15 exceto padrões de alta disponibilidade
- SUSE Linux Enterprise Server 15 SP1 exceto padrões de alta disponibilidade
- SUSE Linux Enterprise Server 15 SP2 exceto padrões de alta disponibilidade
- SUSE Linux Enterprise Server 15 SP3 exceto padrões de alta disponibilidade
- SUSE Linux Enterprise Server 15 SP4 exceto padrões de alta disponibilidade
- SUSE Linux Enterprise Server 12 SP4 exceto padrões de alta disponibilidade
- SUSE Linux Enterprise Server 12 SP5 exceto padrões de alta disponibilidade

Atributos

O CloudWatch Application Insights para SAP NetWeaver 7.0x a 7.5x (incluindo a plataforma ABAP) fornece os seguintes recursos:

- Detecção automática de workload do SAP NetWeaver
- Criação automática de alarmes do SAP NetWeaver baseados em limites estáticos
- Reconhecimento automático de padrões de log do SAP NetWeaver
- Painel de integridade para o SAP NetWeaver

- Painel de problemas para o SAP NetWeaver

Pré-requisitos

É necessário atender aos seguintes pré-requisitos para configurar o SAP NetWeaver com o CloudWatch Application Insights:

- Habilitação do AWS Systems Manager: instale o SSM Agent nas instâncias do Amazon EC2 e habilite-as para o SSM. Para obter informações sobre como instalar o SSM Agent, consulte Configurando o AWS Systems Manager, no Guia do usuário do AWS Systems Manager.
- Perfis de instâncias do Amazon EC2: você deve anexar os perfis de instâncias do Amazon EC2 a seguir para configurar o banco de dados do SAP NetWeaver.
  - Você deve anexar o perfil do `AmazonSSMManagedInstanceCore` para habilitar o Systems Manager. Para obter mais informações, consulte exemplos de políticas baseadas em identidade do AWS Systems Manager.
  - Você deve anexar a política do `CloudWatchAgentServerPolicy` para permitir que métricas e logs de instâncias sejam emitidos por meio do CloudWatch. Para obter mais informações, consulte Criar perfis e usuários do IAM para uso com o agente do CloudWatch.
- Grupos de recursos da AWS: você deve criar um grupo de recursos que inclua todos os recursos associados da AWS usados pela pilha de aplicações para integrar suas aplicações ao CloudWatch Application Insights. Isso inclui instâncias do Amazon EC2, volumes do EFS e volumes do Amazon EBS que executam os servidores de aplicações do SAP NetWeaver. Se houver vários sistemas SAP NetWeaver por conta, recomendamos criar um grupo de recursos que inclua os recursos da AWS para cada sistema SAP NetWeaver. Para obter mais informações sobre a criação de grupos de recursos, consulte o Guia do usuário de grupos de recursos e etiquetas da AWS.
- Permissões do IAM: para usuários que não têm acesso administrativo, você deve criar uma política do AWS Identity and Access Management (IAM) que permita ao Application Insights criar um perfil vinculado ao serviço e anexá-lo à identidade do usuário. Para obter informações sobre como criar uma política do IAM, consulte política do IAM.
- Perfil vinculado ao serviço: o Application Insights usa os perfis vinculadas ao serviço do AWS Identity and Access Management (IAM). Ao criar a primeira aplicação com o Application Insights, é criada uma função vinculada ao serviço no console do Application Insights. Para ter mais informações, consulte Usar funções vinculadas ao serviço do CloudWatch Application Insights.
- Agente do Amazon CloudWatch: o Application Insights instala e configura o agente do CloudWatch. Se você tiver o agente CloudWatch instalado, o Application Insights manterá sua

configuração. Para evitar um conflito de mesclagem, remova a configuração de recursos que você deseja usar no Application Insights do arquivo de configuração do agente do CloudWatch existente. Para ter mais informações, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

Configurar servidores de aplicações do SAP NetWeaver para monitoramento

Use as etapas a seguir para configurar o monitoramento dos servidores de aplicações do SAP NetWeaver.

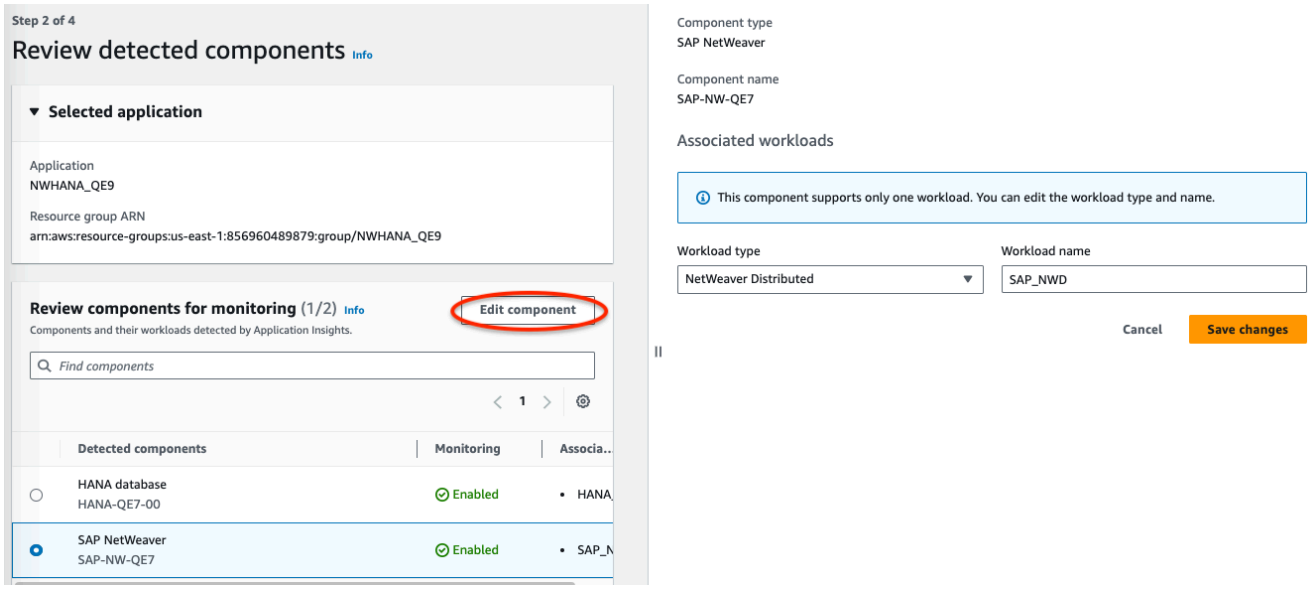
Para configurar o monitoramento

1. Abra o console do CloudWatch.
2. No painel de navegação esquerdo, em Insights, escolha Application Insights (Insights de aplicações).
3. A página Application Insights exibe a lista de aplicações que são monitoradas pelo Application Insights e o status de monitoramento de cada aplicação. No canto superior direito, escolha Add an application (Adicionar uma aplicação).
4. Na página Specify application details (Especificar os detalhes da aplicação), na lista suspensa em Resource group (Grupo de recursos), selecione o grupo de recursos da AWS que você criou e que contém seus recursos de banco de dados do SAP NetWeaver. Se ainda não criou um grupo de recursos para a aplicação, você poderá criar um escolhendo Create new resource group (Criar novo grupo de recursos) na lista suspensa Resource group (Grupo de recursos).
5. Em Automatic monitoring of new resources (Monitoramento automático de novos recursos), marque a caixa de seleção para permitir que o Application Insights monitore automaticamente os recursos que forem adicionados ao grupo de recursos da aplicação após sua integração.
6. Em Monitor EventBridge Events (Monitorar eventos do EventBridge), marque a caixa de seleção para integrar o monitoramento do Application Insights com o CloudWatch Events para obter insights do Amazon EBS, do Amazon EC2, do AWS CodeDeploy, do Amazon ECS, das APIs e notificações do AWS Health, do Amazon RDS, do Amazon S3 e do AWS Step Functions.
7. Em Integrar com o OpsCenter do AWS Systems Manager, marque a caixa de seleção ao lado de Gerar OpsItems do OpsCenter do AWS Systems Manager para ações corretivas para visualizar e receber notificações quando forem detectados problemas nas aplicações selecionadas. Para rastrear as operações realizadas para resolver itens de trabalho operacionais denominados OpsItems, que são relacionados aos recursos da AWS, forneça um ARN do tópico do SNS.

8. É possível, opcionalmente, inserir etiquetas para ajudar a identificar e organizar seus recursos. O CloudWatch Application Insights é compatível com os grupos de recursos baseados em tags e em filas do AWS CloudFormation, exceto os grupos do Application Auto Scaling). Para obter mais informações, consulte Tag Editor (Editor de etiquetas) no Guia do usuário do AWS Resource Groups.
9. Para revisar os componentes detectados, escolha Avançar.
10. Na página Revisar componentes detectados, são listados os componentes monitorados e suas workloads automaticamente detectadas pelo CloudWatch Application Insights.

    - Para editar o tipo e o nome da workload, escolha Editar componente.

> **Note**
>
> Os componentes que contêm uma workload do NetWeaver Distributed ou NetWeaver High Availability detectada oferecem suporte a apenas uma workload em um componente.



11. Escolha Próximo.
12. Na página Specify Details (Especificar detalhes), escolha Next (Avançar).
13. Revise a configuração de monitoramento de aplicações e escolha Enviar.
14. A página de detalhes da aplicação se abrirá, e nela será possível visualizar o Resumo da aplicação, o Painel, os Components e as Workloads. Você também poderá visualizar o

Configuration history (Histórico de configuração), os Log patterns (Padrões de log) e as Tags (Etiquetas) criadas. Depois de enviar sua aplicação, o CloudWatch Application Insights implanta todas as métricas e alarmes para o sistema SAP NetWeaver, o que pode levar até uma hora.

Gerenciar o monitoramento dos servidores de aplicações do SAP NetWeaver

Use as etapas a seguir para gerenciar o monitoramento dos servidores de aplicações do SAP NetWeaver.

Para gerenciar o monitoramento

1. Abra o console do CloudWatch.
2. No painel de navegação esquerdo, em Insights, escolha Application Insights (Insights de aplicações).
3. Escolha a guia List view (Visualização em lista).
4. A página Application Insights exibe a lista de aplicações que são monitoradas pelo Application Insights e o status de monitoramento de cada aplicação.
5. Selecione a aplicação.
6. Escolha a guia Componets (Componentes).
7. Em Componentes monitorados, selecione o botão ao lado do nome do componente. Depois, selecione Manage monitoring (Gerenciar monitoramento).
8. Em Instance logs (Logs de instâncias), é possível atualizar o caminho de log existente, o conjunto de padrões de log e o nome do grupo de logs. Além disso, você pode adicionar até outros três Application logs (Logs de aplicações).
9. Em Metrics (Métricas), você pode escolher as métricas do SAP NetWeaver de acordo com suas necessidades. Os nomes das métricas do SAP NetWeaver são prefixados com `sap`. Você pode adicionar até 40 métricas por componente.
10. Em Custom alarms (Alarmes personalizados), é possível pode adicionar alarmes extras para serem monitorados pelo CloudWatch Application Insights.
11. Revise a configuração de monitoramento da aplicação e escolha Save (Salvar). Quando você envia a configuração, sua conta atualiza todas as métricas e alarmes do sistema SAP NetWeaver, o que pode levar até duas horas.

Visualizar e solucionar problemas do SAP NetWeaver detectados pelo CloudWatch Application Insights

As seções a seguir fornecem as etapas para ajudar você a resolver cenários comuns de solução de problemas que ocorrem quando você configura o monitoramento para o SAP NetWeaver no Application Insights.

Tópicos de solução de problemas

- Problemas de conectividade com o banco de dados do SAP NetWeaver
- Problemas de disponibilidade de aplicações do SAP NetWeaver

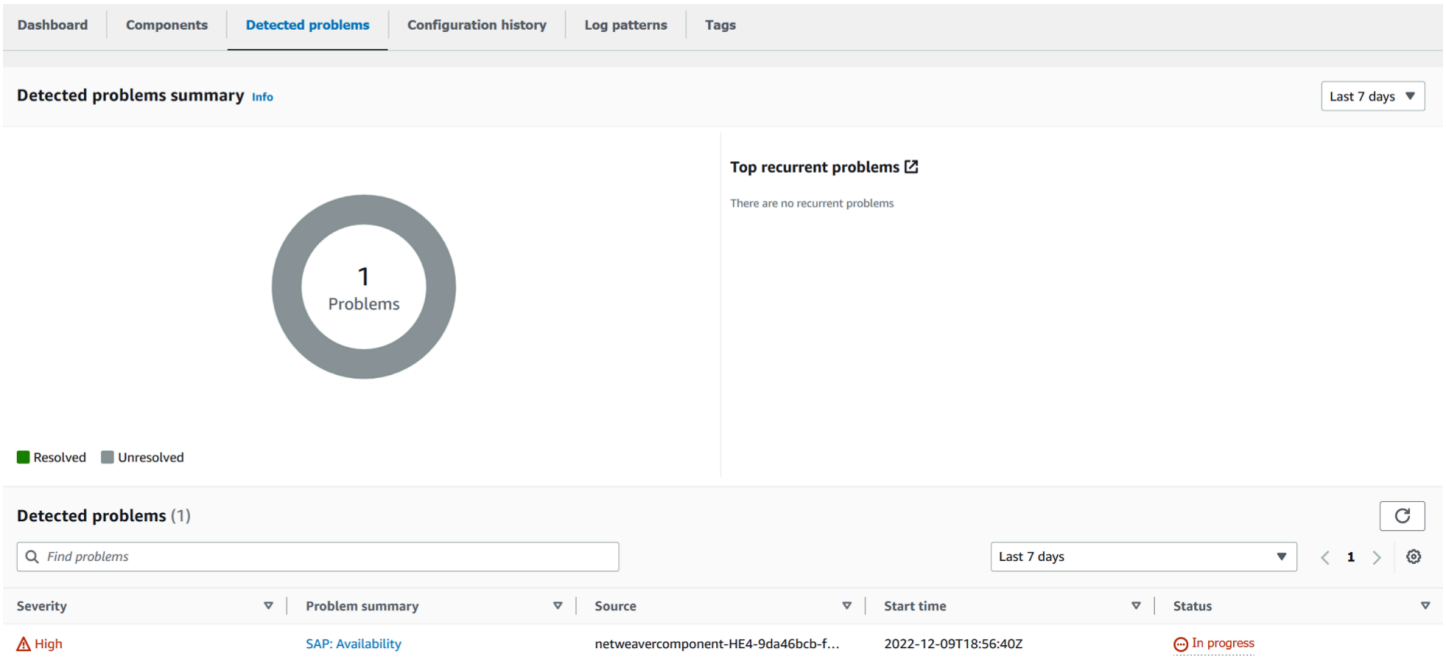
Problemas de conectividade com o banco de dados do SAP NetWeaver

Descrição

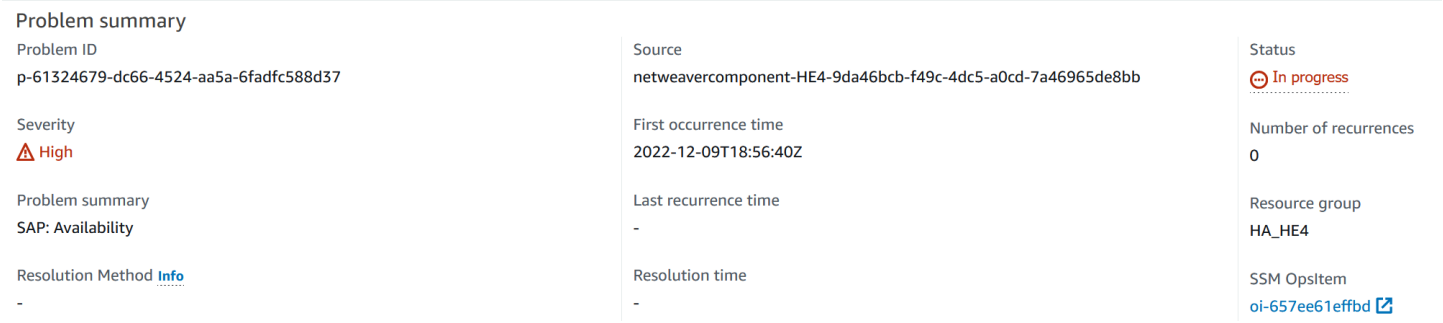
A aplicação do SAP NetWeaver enfrenta problemas de conectividade com o banco de dados.

Causa

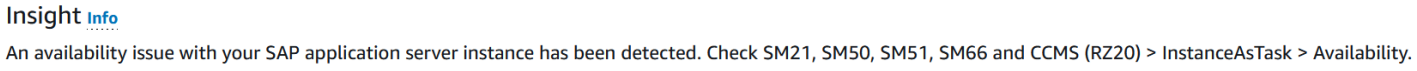
Você pode identificar o problema de conectividade acessando o console do CloudWatch Application Insights e verificando o painel de problemas do SAP NetWeaver Application Insights. Selecione o link em Problem summary (Resumo do problema) para ver o problema específico.



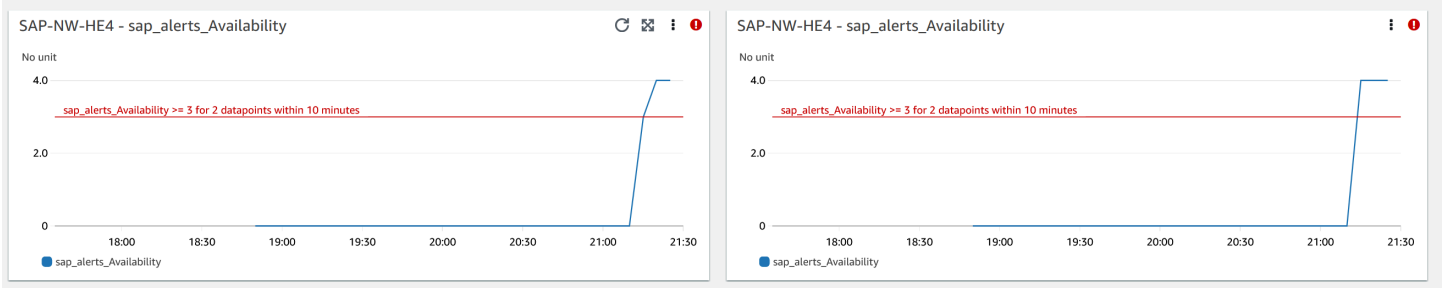
No exemplo a seguir, em Problem summary (Resumo do problema), SAP: Availability (SAP: disponibilidade) é o problema.



Imediatamente após o Problem summary (Resumo do problema), a seção Insight fornece mais contexto sobre o erro e onde você pode obter mais informações sobre suas causas.

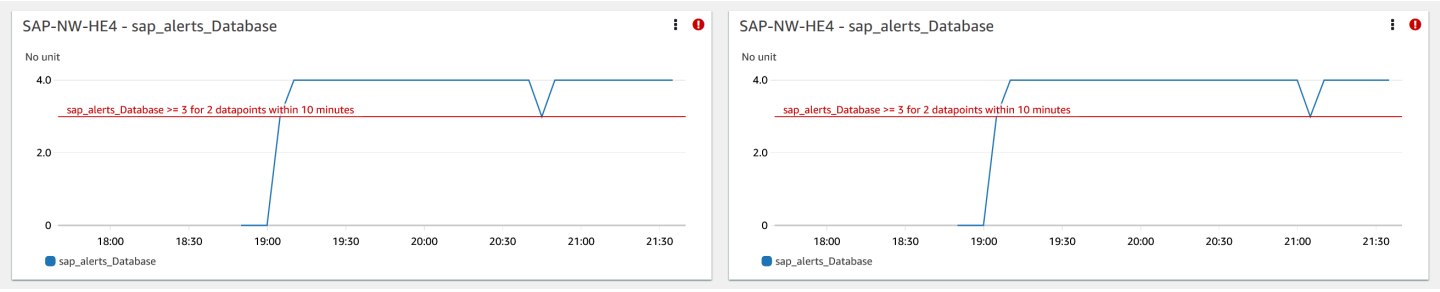


No mesmo painel de problemas, você pode visualizar os logs e as métricas relacionados que a detecção de problemas agrupou para ajudar você a isolar a causa do erro. A métrica `sap_alerts_Availability` monitora a disponibilidade do sistema SAP NetWeaver ao longo do tempo. Você pode usar o rastreamento histórico para correlacionar quando a métrica iniciou um estado de erro ou violou o limite de alarme. No exemplo a seguir, há um problema de disponibilidade com o sistema SAP NetWeaver. O exemplo mostra dois alarmes porque há duas instâncias do servidor de aplicações SAP e um alarme foi criado para cada instância.

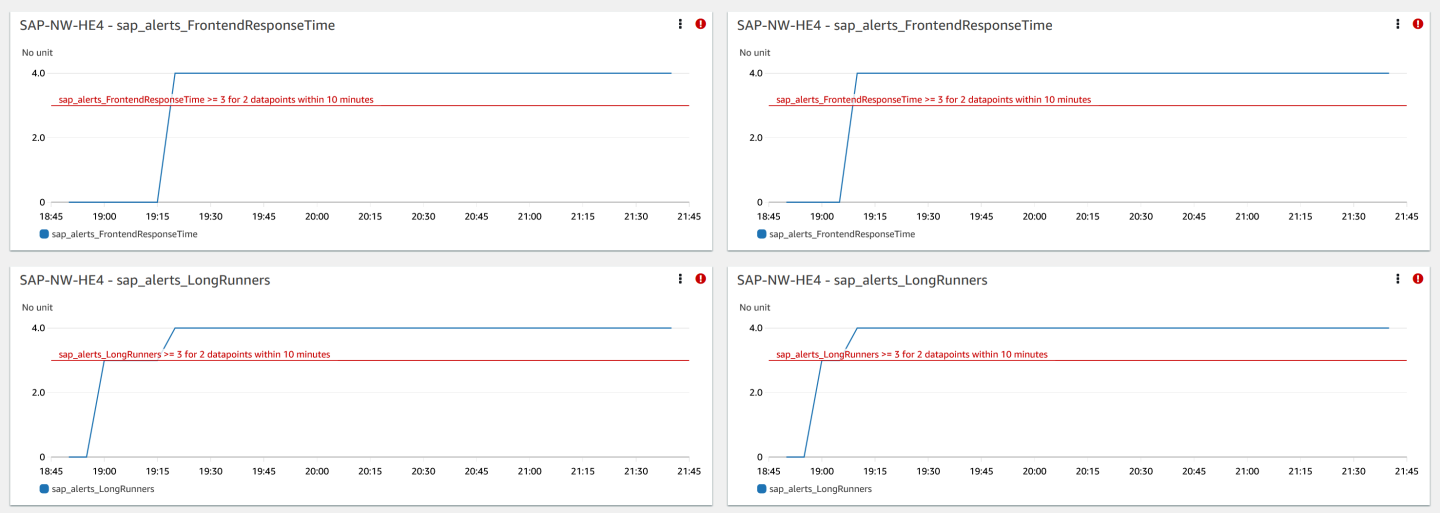


Para obter mais informações sobre cada alarme, passe o mouse sobre o nome da métrica `sap_alerts_Availability`.

No exemplo a seguir, a métrica `sap_alerts_Database` mostra que o nível de banco de dados tem um problema ou uma falha. Esse alarme indica que o SAP NetWeaver teve problemas para se conectar ou se comunicar com seu banco de dados.



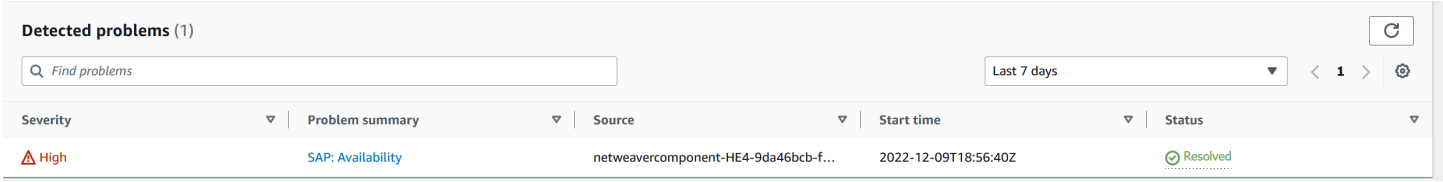
Como o banco de dados é um recurso essencial do SAP NetWeaver, você pode receber muitos alarmes relacionados quando o banco de dados sofre um problema ou uma falha. No exemplo a seguir, as métricas `sap_alerts_FrontendResponseTime` e `sap_alerts_LongRunners` são iniciadas porque o banco de dados não está disponível.



Resolução

O Application Insights monitora de hora em hora o problema detectado. Se não houver novas entradas de log relacionadas nos arquivos de log do SAP NetWeaver, as entradas de log mais antigas serão tratadas como resolvidas. Você deve corrigir todas as condições de erro relacionadas aos alarmes do CloudWatch. Depois que as condições de erro forem resolvidas, o alarme será resolvido quando os alarmes e os logs forem recuperados. Quando todos os erros e alarmes de log do CloudWatch forem resolvidos, o Application Insights interromperá a detecção de erros e o problema será resolvido automaticamente em uma hora. Recomendamos que você resolva todas as condições de erro e alarmes do log para ter os problemas mais recentes no painel de problemas.

No exemplo a seguir, o problema de disponibilidade do SAP está resolvido.



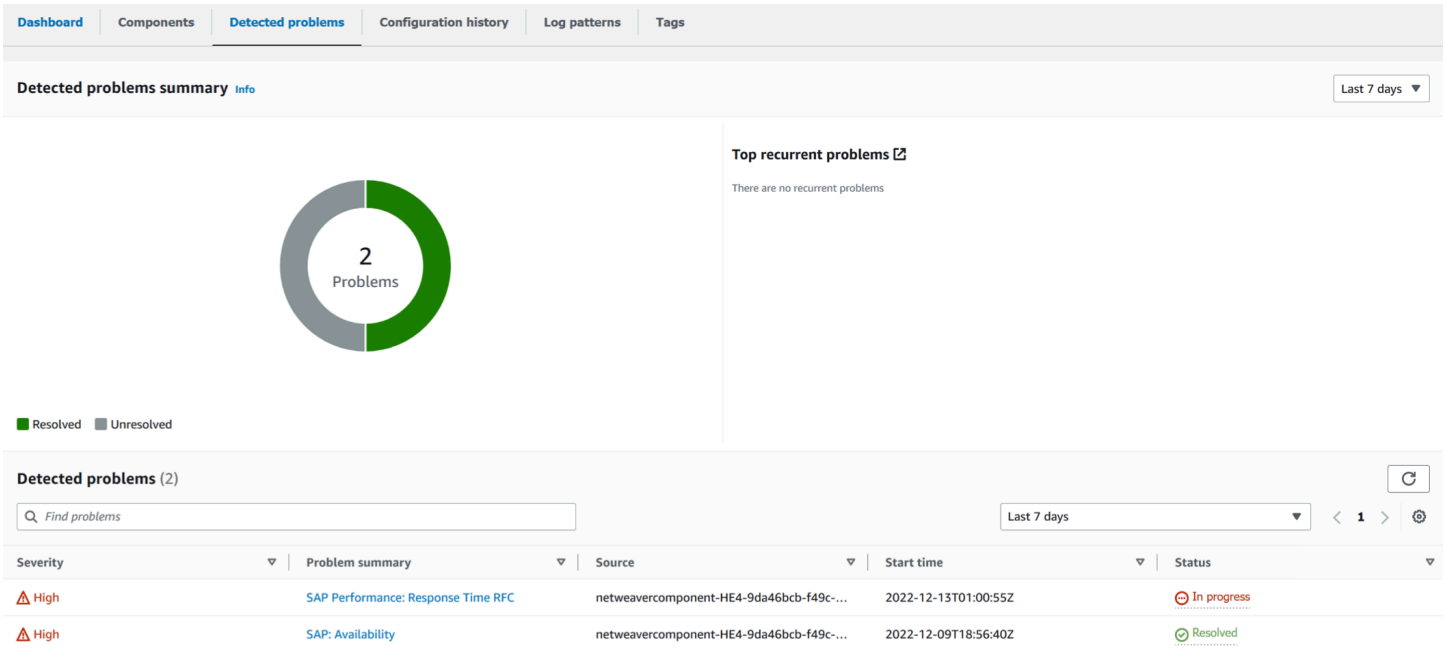
Problemas de disponibilidade de aplicações do SAP NetWeaver

Descrição

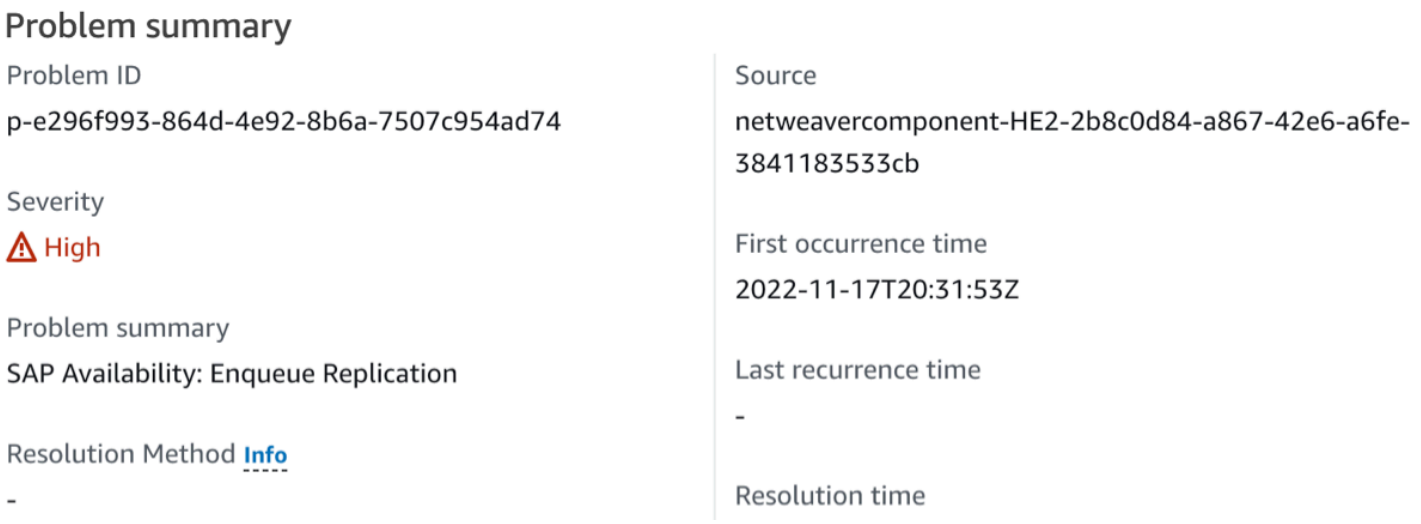
A replicação em fila de alta disponibilidade do SAP NetWeaver parou de funcionar.

Causa

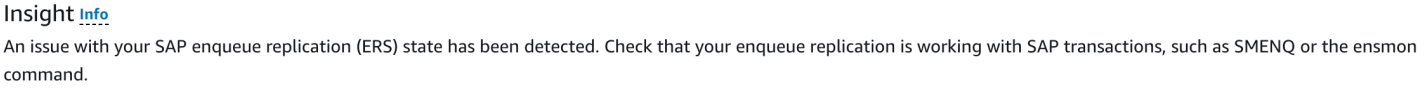
Você pode identificar o problema de conectividade acessando o console do CloudWatch Application Insights e verificando o painel de problemas do SAP NetWeaver Application Insights. Selecione o link em Problem summary (Resumo do problema) para ver o problema específico.



No exemplo a seguir, em Problem summary (Resumo do problema), a replicação em fila de alta disponibilidade é o problema.

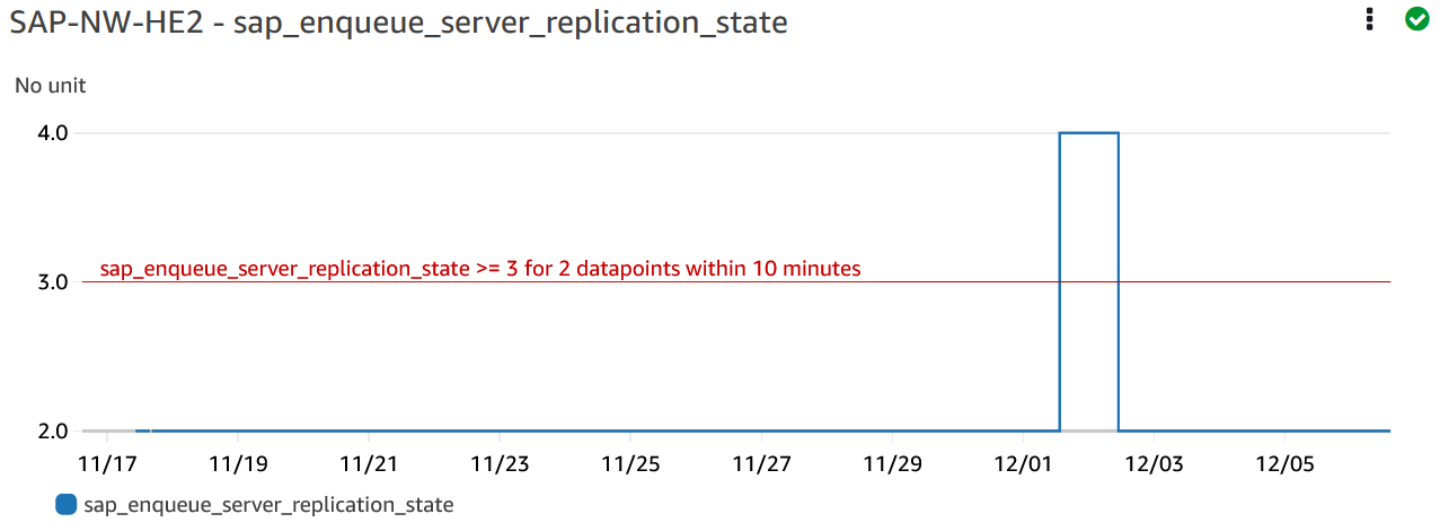


Imediatamente após o Problem summary (Resumo do problema), a seção Insight fornece mais contexto sobre o erro e onde você pode obter mais informações sobre suas causas.

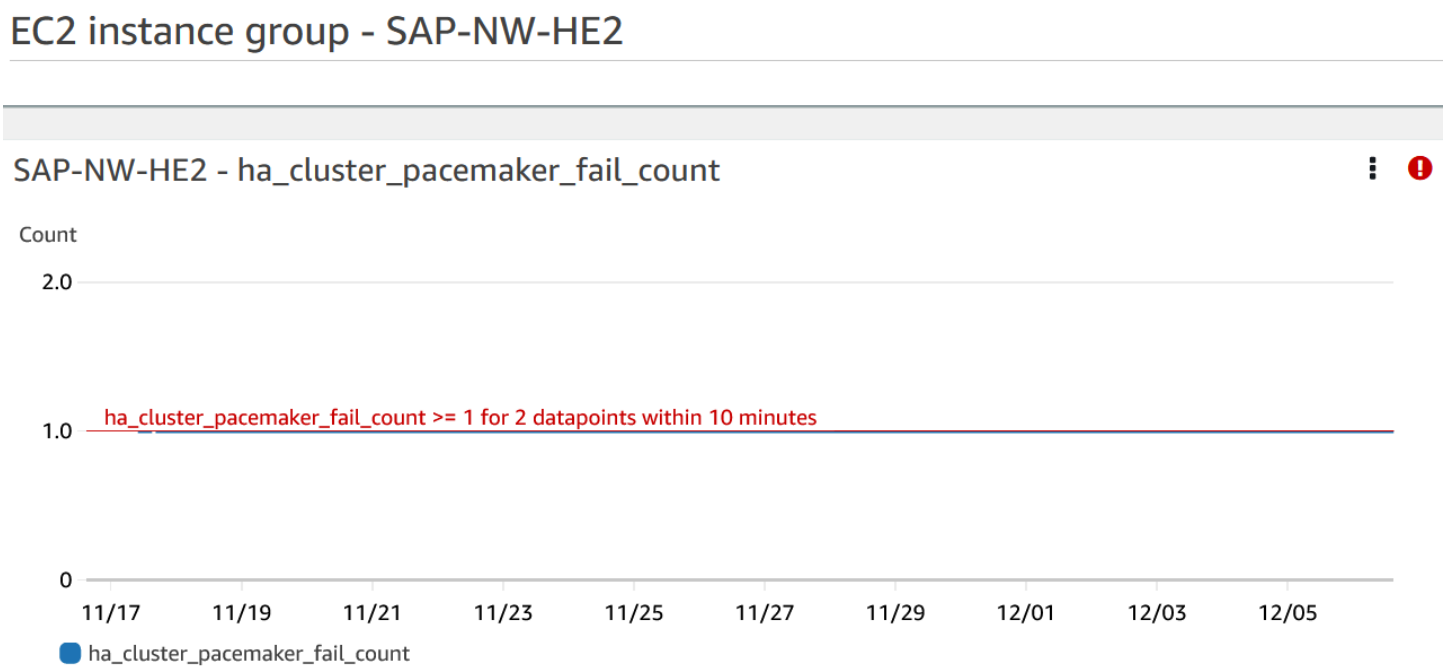


O exemplo a seguir mostra o painel de problemas onde você visualiza os logs e as métricas que são agrupados para ajudar você a isolar as causas de erro. A métrica `sap_enqueue_server_replication_state` rastreia o valor ao longo do tempo. Você pode usar

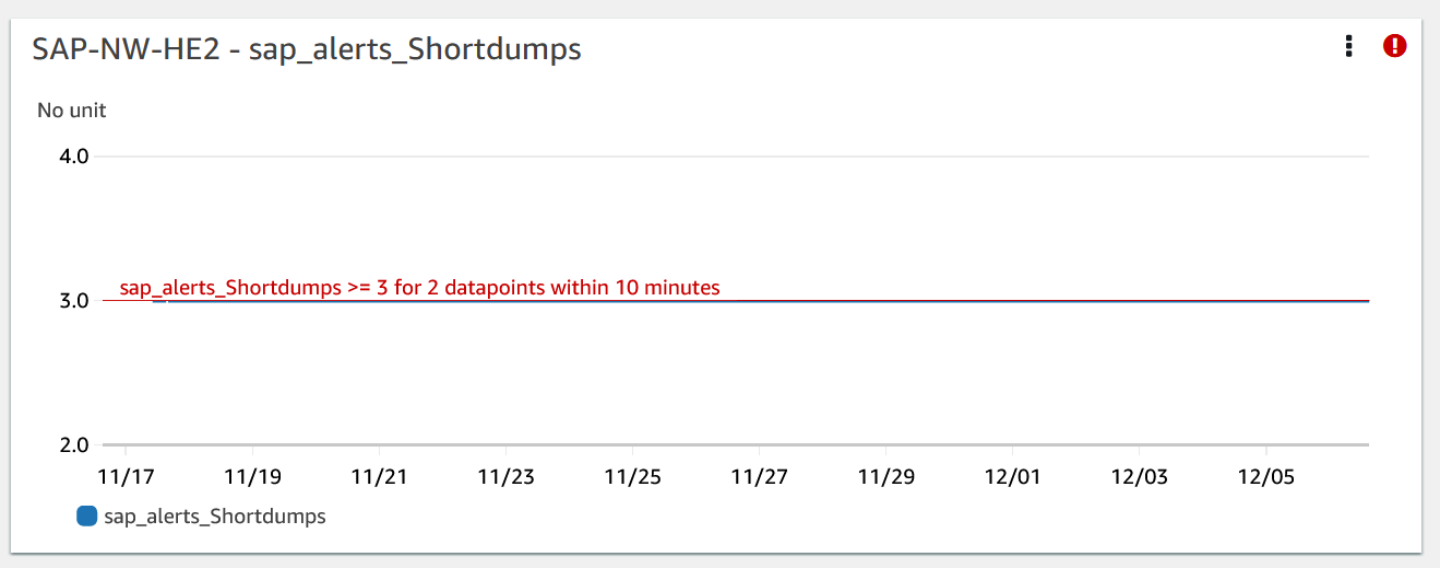
o rastreamento histórico para correlacionar quando a métrica iniciou um estado de erro ou violou o limite de alarme.



No exemplo a seguir, a métrica `ha_cluster_pacemaker_fail_count` mostra que o cluster de marcapassos de alta disponibilidade sofreu uma falha de recurso. Os recursos específicos de marcapasso que tiveram uma contagem de falhas maior ou igual a um são identificados no painel do componente.

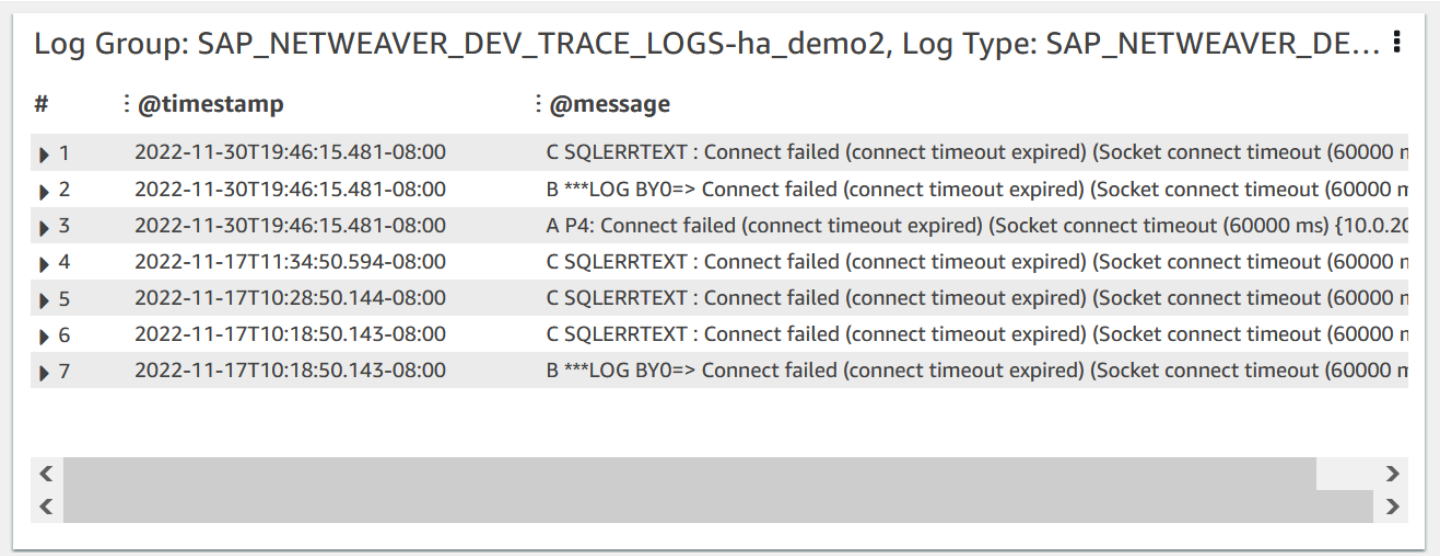


O exemplo a seguir mostra a métrica `sap_alerts_Shortdumps`, que indica que a performance da aplicação SAP caiu quando o problema foi detectado.

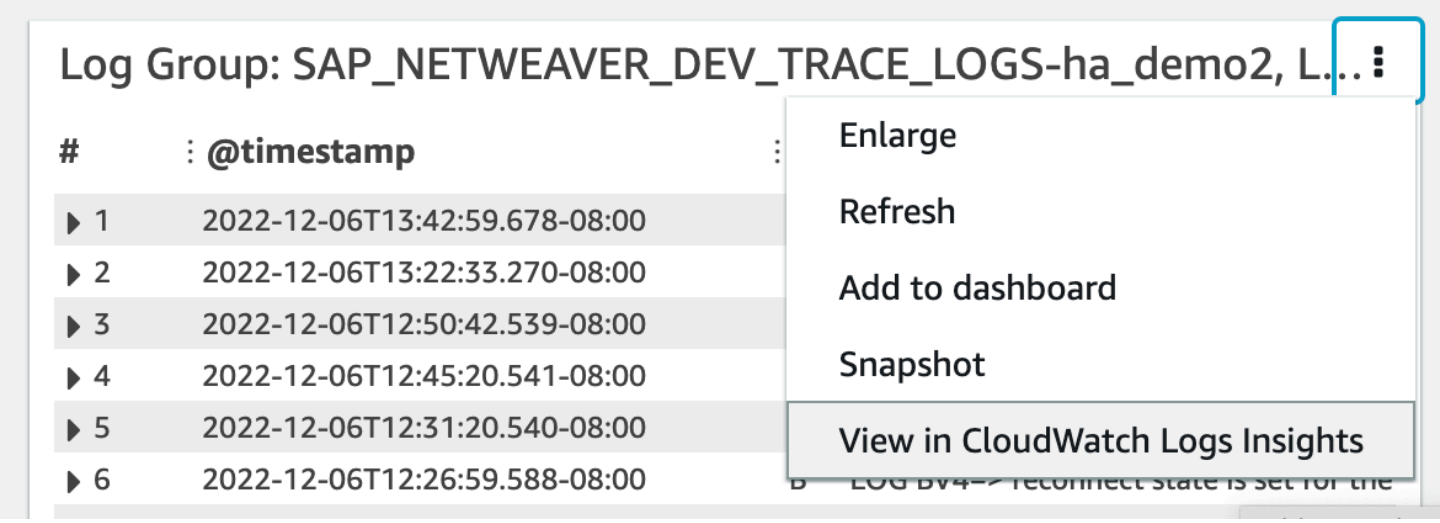


Logs

As entradas de log são úteis para se entender melhor os problemas que ocorreram no nível do SAP NetWeaver quando o problema foi detectado. O widget de grupo de logs no painel de problemas exibe a hora específica do problema.



Para ver informações detalhadas sobre os logs, selecione os três pontos verticais no canto superior direito e selecione View in CloudWatch Logs Insights (Exibir no CloudWatch Logs Insights).

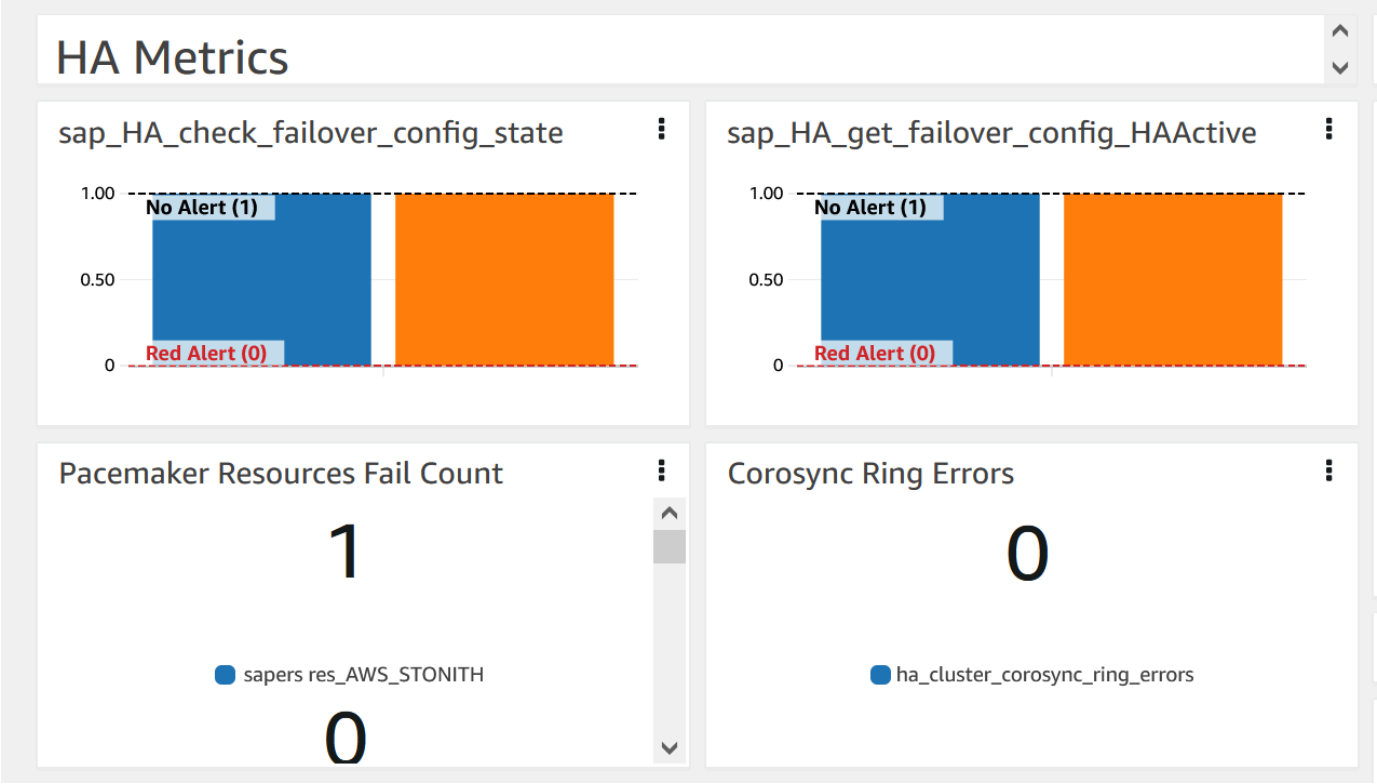


Use as etapas a seguir para obter mais informações sobre as métricas e os alarmes exibidos no painel de problemas.

Para obter mais informações sobre métricas e alarmes

1. Abra o console do CloudWatch.
2. No painel de navegação esquerdo, em Insights, selecione Application Insights (Insights sobre aplicações). Depois, escolha a guia List view (Visualização em lista) e selecione a aplicação.
3. Selecione a guia Components (Componentes). Depois, selecione o componente do SAP NetWeaver sobre o qual você deseja obter mais informações.

   O exemplo a seguir mostra a seção HA Metrics (Métricas de HA) com a métrica `ha_cluster_pacemaker_fail_count` que foi exibida no painel de problemas.

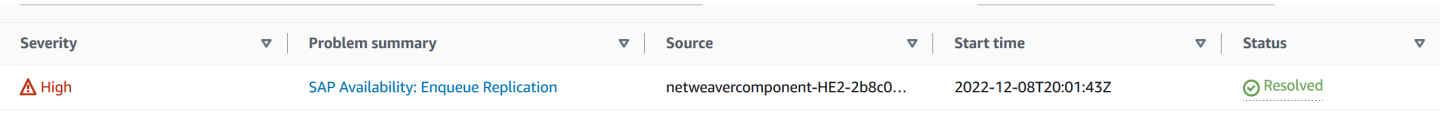


Resolução

O Application Insights monitora de hora em hora o problema detectado. Se não houver novas entradas de log relacionadas nos arquivos de log do SAP NetWeaver, as entradas de log mais antigas serão tratadas como resolvidas. Você deve corrigir todas as condições de erro relacionadas a esse problema.

Para o alarme `sap_alerts_Shortdumps`, você deve resolver o alerta no sistema SAP NetWeaver usando o código da transação `RZ20 # R3Abap # Shortdumps` para navegar até o alerta do CCMS. Para obter mais informações sobre os alertas do CCMS, consulte o site da SAP. Resolva todos os alertas do CCMS na árvore de shortdumps. Depois que todos os alertas forem resolvidos no sistema SAP NetWeaver, o CloudWatch não relatará mais a métrica em estado de alarme.

Quando todos os erros e alarmes de log do CloudWatch forem resolvidos, o Application Insights interromperá a detecção de erros e o problema será resolvido automaticamente em uma hora. Recomendamos que você resolva todas as condições de erro e alarmes do log para ter os problemas mais recentes no painel de problemas. No exemplo a seguir, em Problem summary (Resumo do problema), o problema de replicação em fila de alta disponibilidade foi resolvido.



Solução de problemas do Application Insights para o SAP NetWeaver

Esta seção fornece etapas para ajudar você a resolver erros comuns retornados pelo painel do Application Insights.

Não é possível adicionar mais de 60 métricas de monitoramento

Erro retornado: `Component cannot have more than 60 monitored metrics.`

Causa raiz: `The current metric limit is 60 monitor metrics per component.`

Resolução: remova métricas que não são necessárias para aderir ao limite.

As métricas do SAP não são exibidas no painel após o processo de integração

Causa raiz: o painel de componentes usa um período de métrica de cinco minutos para agregar os pontos de dados.

Resolução: todas as métricas devem aparecer no painel após cinco minutos.

As métricas e os alarmes do SAP não são no painel

Use as etapas a seguir para identificar por que as métricas e os alarmes do SAP não são exibidos no painel após o processo de integração.

Para identificar o problema com as métricas e os alarmes

1. Abra o console do CloudWatch.
2. No painel de navegação esquerdo, em Insights, selecione Application Insights (Insights sobre aplicações). Depois, escolha a guia List view (Visualização em lista) e selecione a aplicação.
3. Escolha a guia Configuration (Configuração).
4. Se você encontrar pontos de dados com métricas ausentes, verifique se há erros relacionados a `prometheus-sap_host_exporter`.
5. Se você não encontrar nenhum erro na etapa anterior, conecte-se à sua instância do Linux. Para implantações de alta disponibilidade, conecte-se à instância primária do cluster do Amazon EC2.

6. Na instância, verifique se o exportador está sendo executado usando o comando a seguir. A porta padrão é a `9680`. Se você estiver usando uma porta diferente, substitua `9680` pela porta que você está usando.

```
curl localhost:9680/metrics
```

   Se nenhum dado for retornado, houve falha na inicialização do exportador.

7. Para encontrar a convenção de nomenclatura correta a ser usada para `WORKLOAD_SHORT_NAME` nas próximas duas etapas, execute o comando a seguir.

> **Note**
>
> O Application Insights adiciona um sufixo, `WORKLOAD_SHORT_NAME`, ao nome do serviço, dependendo da workload em execução. Os nomes abreviados das implantações distribuídas, padrão e de alta disponibilidade do NetWeaver são `SAP_NWD`, `SAP_NWS` e `SAP_NWH`.

```
sudo systemctl | grep exporter
```

8. Para verificar se há erros nos logs do serviço exportador, execute o seguinte comando:

```
sudo journalctl -e --unit=prometheus-sap_host_exporter_WORKLOAD_SHORT_NAME.service
```

9. Para verificar erros nos logs de serviço do gerenciador de exportação, execute o seguinte comando:

```
sudo journalctl -e --unit=prometheus-
sap_host_exporter_manager_WORKLOAD_SHORT_NAME.service
```

> **Note**
>
> Esse serviço deve estar em funcionamento todo o tempo.

   Se esse comando não retornar nenhum erro, siga para a próxima etapa.

10. Para iniciar o exportador manualmente, execute o comando a seguir. Depois, verifique a saída do exportador.

    ```
    sudo /opt/aws/sap_host_exporter/sap_host_exporter
    ```

    Você pode sair do processo de exportação depois que verificar se houve erros.

Causa raiz: há várias causas possíveis para esse problema. Uma causa comum é o exportador não conseguir se conectar a uma das instâncias do servidor de aplicações.

Resolução

Use as etapas a seguir para conectar o exportador às instâncias do servidor de aplicações. Você verificará que a instância da aplicação SAP está em execução e usará o SAPControl para se conectar à instância.

Para conectar o exportador às instâncias do servidor de aplicações

1. Em sua instância do Amazon EC2, execute o comando a seguir para verificar se a aplicação SAP está sendo executada.

   ```
   sapcontrol -nr <App_InstNo> -function GetProcessList
   ```

2. Você deve estabelecer uma conexão funcional do SAPControl. Se a conexão do SAPControl não funcionar, encontre a causa raiz do problema na instância relevante da aplicação SAP.
3. Para iniciar manualmente o exportador depois de corrigir o problema de conexão do SAP Control, execute o seguinte comando:

   ```
   sudo systemctl start prometheus-sap_host_exporter.service
   ```

4. Se você não conseguir resolver o problema de conexão do SAP Control, execute o procedimento a seguir como uma correção temporária.

   a. Abra o console de AWS Systems Manager.

   b. No painel de navegação esquerdo, escolha Automation (Automação).

   c. Em Associations (Associações), procure a associação do sistema SAP NetWeaver.

```
Association Name: Equal: AWS-ApplicationInsights-
SSMSAPHostExporterAssociationForCUSTOMSAPNW<SID>-1
```

d. Selecione o Association id (ID da associação).

e. Escolha a guia Parameters (Parâmetros) e remova o número do servidor de aplicações de AdditionalArguments.

f. Escolha Apply association now (Aplicar associação agora).

> **Note**
>
> Essa é uma solução temporária. Se forem feitas atualizações nas configurações de monitoramento do componente, a instância será adicionada novamente.

Visualizar e solucionar problemas detectados pelo Amazon CloudWatch Application Insights

Os tópicos desta seção fornecem informações detalhadas sobre os problemas e insights detectados que são exibidos pelo Application Insights. Eles também fornecem sugestões de resoluções para problemas detectados com sua conta ou sua configuração.

Tópicos de solução de problemas

- Visão geral do console do CloudWatch
- Página de resumo do problema do Application Insights
- Falhas de conflito de mesclagem do agente do CloudWatch
- Os alarmes não são criados
- Feedback
- Erros de configuração

Visão geral do console do CloudWatch

Uma visão geral dos problemas que impactam suas aplicações monitoradas podem ser encontradas no painel do CloudWatch Application Insights na página de visão geral do console do CloudWatch. Para ter mais informações, consulte Comece a usar o Amazon CloudWatch Application Insights.

O painel de visão geral do CloudWatch Application Insights exibe o seguinte:

- A gravidade dos problemas detectados: Alta/Média/Baixa
- Um breve resumo do problema
- A fonte do problema
- A hora em que o problema começou
- O status de resolução do problema
- O grupo de recursos afetado

Para visualizar os detalhes de um problema específico, em Problem Summary (Resumo do problema), selecione a descrição do problema. Um painel detalhado exibe informações sobre o problema e as anomalias de métrica relacionadas, além de trechos dos erros de log. É possível fornecer feedback sobre a relevância do insight selecionando se ele é útil.

Se um recurso novo que não esteja configurado for detectado, a descrição do resumo do problema direciona você ao assistente Edit configuration (Editar configuração) para configurar o novo recurso. Se for necessário, você poderá visualizar ou editar a configuração do Grupo de recursos selecionando View/edit configuration (Visualizar/editar configuração) no canto superior direito do painel detalhado.

Para retornar à visão geral, selecione Back to overview (Voltar à visão geral), que está ao lado do cabeçalho do painel detalhado do CloudWatch Application Insights.

Página de resumo do problema do Application Insights

Página de resumo do problema do Application Insights

O CloudWatch Application Insights fornece as seguintes sobre os problemas detectados na página de resumo do problema:

- Um breve resumo do problema
- A data e a hora de início do problema
- A gravidade do problema: High/Medium/Low (Alta/média/baixa)
- O status do problema detectado: In-progress/Resolved (Em andamento/resolvido)
- Insights: insights gerados automaticamente sobre o problema detectado e a possível causa
- Feedback sobre os insights: o feedback que você forneceu sobre a utilidade dos insights gerados pelo CloudWatch Application Insights

- Observações relacionadas: uma visão detalhada das anomalias da métrica e dos trechos do erro de logs relevantes relacionados ao problema em vários componentes da aplicação

Falhas de conflito de mesclagem do agente do CloudWatch

O CloudWatch Application Insights instala e configura o agente do CloudWatch em instâncias do cliente. Isso inclui a criação de um arquivo de configuração do agente do CloudWatch com configurações para métricas ou logs. Um conflito de mesclagem poderá ocorrer se a instância do cliente já tiver um arquivo de configuração do agente do CloudWatch com configurações diferentes definidas para as mesmas métricas ou logs. Para resolver o conflito da mesclagem, use as etapas a seguir:

1. Identifique os arquivos de configuração do agente do CloudWatch em seu sistema. Para obter mais informações sobre localizações de arquivos, consulte Arquivos e locais do atendente do CloudWatch.
2. Remova as configuração de recursos que você deseja usar no Application Insights do arquivo de configuração do agente do CloudWatch existente. Se você deseja usar somente configurações do Application Insights, exclua os arquivos de configuração do agente do CloudWatch existentes.

Os alarmes não são criados

Para algumas métricas, o Application Insights prevê o limite de alarme com base nos pontos de dados anteriores da métrica. Para habilitar essa previsão, os critérios a seguir devem ser atendidos.

- Pontos de dados recentes: deve haver no mínimo 100 pontos de dados das últimas 24 horas. Os pontos de dados não precisam ser contínuos e podem estar espalhados por todo o período de 24 horas.
- Dados históricos: deve haver um mínimo de 100 pontos de dados abrangendo o período de 15 dias antes da data atual até 1 dia antes da data atual. Os pontos de dados não precisam ser contínuos e podem estar espalhados por todo o período de 15 dias.

> **Note**
>
> Para algumas métricas, o Application Insights atrasa a criação de alarmes até que as condições anteriores sejam atendidas. Nesse caso, você obtém um evento do histórico

de configuração informando que a métrica não tem pontos de dados suficientes para estabelecer o limite do alarme.

Feedback

Feedback

É possível fornecer feedback sobre os insights gerados automaticamente sobre problemas detectados designando-os como úteis ou não úteis. Seu feedback sobre os insights com o diagnóstico da aplicação (anomalias da métrica e exceções de log) são usados para melhorar a futura detecção de problemas semelhantes.

Erros de configuração

O CloudWatch Application Insights usa sua configuração para criar telemetrias de monitoramento para os componentes. Quando o Application Insights detecta um problema com sua conta ou configuração, são fornecidas informações no campo Remarks (Observações) sobre como resolver o problema de configuração da sua aplicação.

A tabela a seguir mostra resoluções sugeridas para observações específicas.

Observações	Resolução sugerida	Notas adicionais
A cota para o CloudFormation já foi atingida.	O Application Insights cria uma pilha do CloudFormation para que cada aplicação para gerencie a instalação e a configuração do atendente do CloudWatch para todos os componentes da aplicação . Por padrão, cada conta da AWS pode conter 2.000 pilhas. Consulte Limites do AWS CloudFormation. Para resolver isso, aumente o limite para pilhas do CloudForm ation.	n/a

Observações	Resolução sugerida	Notas adicionais
Não há função de instância do SSM nas instâncias a seguir.	Para que o Application Insights consiga instalar e configurar o atendente do CloudWatch em instâncias da aplicação, as políticas AmazonSSMManagedInstanceCore e CloudWatchAgentServerPolicy devem estar anexadas à função da instância.	O Application Insights chama a API DescribeInstanceInformation do SSM para obter a lista de instâncias com permissão do SSM. Depois que a função é anexada à instância, leva um tempo para que o SSM inclua a instância no resultado DescribeInstanceInformation. Até que o SSM inclua a instância no resultado, o erro NO_SSM_INSTANCE_ROLE permanece presente para a aplicação.
Novos componentes podem precisar de configuração.	O Application Insights detecta que há novos componentes no Grupo de recursos da aplicação. Para resolver isso, configure os novos componentes adequadamente.	n/a

Logs e métricas compatíveis com o Amazon CloudWatch Application Insights

As listas a seguir mostram os logs e as métricas compatíveis com o Amazon CloudWatch Application Insights.

O CloudWatch Application Insights é compatível com os seguintes logs:

- Logs do Microsoft Internet Information Services (IIS)
- Log de erro do SQL Server no EC2
- Logs da aplicação .NET personalizados, como Log4Net

- Logs de eventos do Windows, incluindo registros do Windows (sistema, aplicação e segurança) e registros de aplicações e serviços
- Amazon CloudWatch Logs for AWS Lambda
- Log de erros e log lento para RDS MySQL, Aurora MySQL e MySQL no EC2
- Log do PostgreSQL para o PostgreSQL RDS e o PostgreSQL no EC2
- Amazon CloudWatch Logs for AWS Step Functions
- Logs de execução e logs de acesso (JSON, CSV e XML, mas não CLF) para estágios de API REST do API Gateway
- Logs do Prometheus JMX Exporter (EMF)
- Logs de alerta e logs de listener para Oracle no Amazon RDS e Oracle no Amazon EC2
- Roteamento de logs de contêiner do Amazon ECS para o CloudWatch usando o driver de log `awslogs`.
- Roteamento de logs de contêineres do Amazon ECS para o CloudWatch usando o roteamento de log de contêiner do FireLens.
- Roteamento de logs de contêiner do Amazon EKS ou Kubernetes em execução no Amazon EC2 para o CloudWatch usando o processador de log Fluent Bit ou Fluentd com o Container Insights.
- Logs de erros e rastreamento do SAP HANA
- Logs de HA Pacemaker
- Logs do servidor do SAP ASE
- Logs do servidor de backup do SAP ASE
- Logs do servidor de replicação do SAP ASE
- Logs do agente RMA do SAP ASE
- Logs do gerenciador de falhas do SAP ASE
- Logs de rastreamentos para desenvolvedor do SAP NetWeaver
- Métricas de processo para processos do Windows usando o plug-in proctstat para o atendente do CloudWatch
- Logs de consulta de DNS pública para a zona hospedada
- Logs de consulta de DNS do Amazon Route 53 Resolver

O CloudWatch Application Insights é compatível com as seguintes classes de log:

- Padrão: o Amazon CloudWatch Application Insights requer que os grupos de logs sejam configurados com a classe de log padrão do CloudWatch Logs para habilitar o monitoramento.

O CloudWatch Application Insights é compatível com as métricas para os seguintes componentes da aplicação:

- Amazon Elastic Compute Cloud (EC2)
  - Métricas integradas do CloudWatch
  - Métricas do atendente do CloudWatch (Windows Server)
  - Métricas de processo do atendente do CloudWatch (servidor Windows)
  - Métricas do atendente do CloudWatch (servidor Linux)
- Elastic Block Store (EBS)
- Amazon Elastic File System (Amazon EFS)
- Elastic Load Balancer (ELB)
- Aplicativo ELB
- Grupos do Amazon EC2 Auto Scaling
- Amazon Simple Queue Server (SQS)
- Amazon Relational Database Service (RDS)
  - Instâncias de bancos de dados do RDS
  - Clusters de banco de dados do RDS
- Função do AWS Lambda
- Tabela do Amazon DynamoDB
- Bucket do Amazon S3
- AWS Step Functions
  - Nível da execução
  - Atividade
  - Função do Lambda
  - Integração de serviços
  - API Step Functions
- Etapas da API REST do API Gateway
- SAP HANA
- SAP ASE

- Alta disponibilidade do SAP ASE no Amazon EC2
- SAP NetWeaver
- Cluster de HA
- Java
- Amazon Elastic Container Service (Amazon ECS)
  - Métricas integradas do CloudWatch
  - Métricas do Container Insights
  - Métricas do Container Insights Prometheus
- Kubernetes na AWS
  - Métricas do Container Insights
  - Métricas do Container Insights Prometheus
- Amazon FSx
- Amazon VPC
- Gateways de NAT da Amazon VPC
- Verificação de integridade do Amazon Route 53
- Zona hospedada do Amazon Route 53
- Endpoint do Amazon Route 53 Resolver
- Grupo de regras do AWS Network Firewall
- associação do grupos de regras AWS Network Firewall
- Métricas com requisitos de pontos de dados
  - AWS/ApplicationELB
  - AWS/AutoScaling
  - AWS/EC2
  - Elastic Block Store (EBS)
  - AWS/ELB
  - AWS/RDS
  - AWS/Lambda
  - AWS/SQS
  - AWS/CWAgent
  - AWS/DynamoDB

  - AWS/S3
  - AWS/States
  - AWS/ApiGateway
  - AWS/SNS
- Métricas recomendadas
- Métricas de contador de performance

Amazon Elastic Compute Cloud (EC2)

O CloudWatch Application Insights é compatível com as seguintes métricas:

Metrics

- Métricas integradas do CloudWatch
- Métricas do atendente do CloudWatch (Windows Server)
- Métricas de processo do atendente do CloudWatch (servidor Windows)
- Métricas do atendente do CloudWatch (servidor Linux)

Métricas integradas do CloudWatch

CPUCreditBalance

CPUCreditUsage

CPUSurplusCreditBalance

CPUSurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBSByteBalance%

EBSIOBalance%

EBSReadBytes

EBSReadOps

EBSWriteBytes

EBSWriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Instance

StatusCheckFailed_System

Métricas do atendente do CloudWatch (Windows Server)

Nº de exceções do CLR .NET de exceções ocorridas

Nº de exceções do CLR .NET de exceções ocorridas por segundo

Nº de exceções do CLR .NET de filtros por segundo

Nº de exceções do CLR .NET de conclusões por segundo

Exceções do CLR .NET lançadas para capturar profundidade por segundo

Nº de interoperabilidade do CLR .NET de CCWs

Nº de interoperabilidade do CLR .NET de stubs

Nº de interoperabilidade do CLR .NET de exportações TLB/s

Nº de interoperabilidade do CLR .NET de importações TLB/s

Nº de interoperabilidade do CLR .NET de marshaling

% de tempo de Jit do CLR .NET no Jit

Falhas do Jit padrão do Jit do CLR .NET

% de tempo de carregamento do CLR .NET

Taxa de carregamento do CLR .NET de falhas de carga

Taxa de contenção/s LocksAndThreads do CLR .NET

Tamanho da fila/s LocksAndThreads do CLR .NET

Nº total de bytes confirmados de memória do CLR .NET

% de tempo de memória do CLR .NET em GC

Tempo médio da fila HttpWebRequest de rede 4.0.0.0 do CLR .NET

HttpWebRequests canceladas/s de rede 4.0.0.0 do CLR .NET

HttpWebRequests com falha/s de rede 4.0.0.0 do CLR .NET

HttpWebRequests colocadas na fila por segundo de rede 4.0.0.0 do CLR .NET

Total de falhas de ping do processo do operador APP_POOL_WAS

A aplicação ASP.NET é reiniciada

% do tempo de processador gerenciado de aplicações ASP.NET (estimado)

Total de erros de aplicações ASP.NET por segundo

Erros de aplicações ASP.NET não processados durante a execução por segundo

Solicitações de aplicações ASP.NET na fila de aplicações

Solicitações de aplicações ASP.NET por segundo

Tempo de espera de solicitação do ASP.NET

Solicitações do ASP.NET em fila

Filas de solicitação de serviço HTTP CurrentQueueSize

% de espaço livre de disco lógico

% de bytes confirmados em uso na memória

Mbytes de memória disponíveis

Páginas de memória/s

Total/s de bytes de interface de rede

% de uso de arquivo de paginação

% de tempo de disco de disco físico

Média de disco físico Comprimento da fila de discos

Média de disco físico S/leitura de disco

Média de disco físico S/gravação de disco

Bytes/s de leitura de disco do disco físico

Leituras/s de disco do disco físico

Bytes/s de gravação de disco do disco físico

Gravações/s de disco do disco físico

% de tempo ocioso do processador

% de tempo de interrupção do processador

% de tempo de processador do processador

% de tempo de usuário do processador

SQLServer: gravações/s encaminhadas dos métodos de acesso

SQLServer: verificações completas dos métodos de acesso/s

SQLServer: divisões/s da página de métodos de acesso

SQLServer: proporção de acertos do cache do buffer do gerenciador de buffer

SQLServer: expectativa de vida da página do gerenciador de buffer

SQLServer: processos de estatísticas gerais bloqueados

SQLServer: conexões de usuário de estatísticas gerais

SQLServer: trava o tempo médio de espera (ms)

SQLServer: bloqueia o tempo médio de espera (ms)

SQLServer: bloqueia os tempos limite de bloqueio/s

SQLServer: bloqueia a espera de bloqueio/s

SQLServer: bloqueia o número de bloqueios/s

SQLServer: gerenciador de memória concessão de memória pendente

SQLServer: solicitações em lote/s de estatísticas do SQL

SQLServer: compilações do SQL/s de estatísticas do SQL

SQLServer: novas compilações do SQL/s de estatísticas do SQL

Tamanho da fila do processador do sistema

Conexões TCPv4 estabelecidas

Conexões TCPv6 estabelecidas

Descargas de cache de arquivos W3SVC_W3WP

Erros de cache de arquivos W3SVC_W3WP

Solicitações W3SVC_W3WP/s

Descargas de cache URI W3SVC_W3WP

Erros de cache URI W3SVC_W3WP

Bytes de serviço web recebidos por segundo

Bytes de serviço web enviados por segundo

Tentativas de conexão de web service/s

Conexões atuais de serviços web

Solicitações para obter web service/s

Solicitações para publicar web service/s

Bytes recebidos/s

Tamanho da fila de mensagens normais/s

Tamanho da fila de mensagens urgentes/s

Contagem de reconexão

Tamanho da fila de mensagens não confirmadas/s

Mensagens pendentes

Mensagens enviadas/s

Mensagens de atualização de banco de dados/s

Atualizar mensagens/s

Liberações/s

Pontos de verificação de criptografia salvos/s

Pontos de verificação de criptografia restaurados/s

Pontos de verificação do registro restaurados/s

Pontos de verificação do registro salvos/s

Chamadas de API de cluster/s

Chamadas de API de recursos/s

Monitores de cluster/s

Manipulações de recursos/s

Métricas de processo do atendente do CloudWatch (servidor Windows)

As métricas de processo são coletadas usando o plug-in procstat do atendente do CloudWatch. Somente instâncias do Amazon EC2 que executam workloads do Windows oferecem suporte a métricas de processo.

procstat cpu_time_system

procstat cpu_time_user

procstat cpu_usage

procstat memory_rss

procstat memory_vms

procstat read_bytes

procstat write_bytes

.procstat read_count

procstat write_count

Métricas do atendente do CloudWatch (servidor Linux)

cpu_time_active

cpu_time_guest

cpu_time_guest_nice

cpu_time_idle

cpu_time_iowait

cpu_time_irq

cpu_time_nice

cpu_time_softirq

cpu_time_steal

cpu_time_system

cpu_time_user

cpu_usage_active

cpu_usage_guest

cpu_usage_guest_nice

cpu_usage_idle

cpu_usage_iowait

cpu_usage_irq

cpu_usage_nice

cpu_usage_softirq

cpu_usage_steal

cpu_usage_system

cpu_usage_user

disk_free

disk_inodes_free

disk_inodes_used

disk_used

disk_used_percent

diskio_io_time

diskio_iops_in_progress

diskio_read_bytes

diskio_read_time

diskio_reads

diskio_write_bytes

diskio_write_time

diskio_writes

mem_active

mem_available

mem_available_percent

mem_buffered

mem_cached

mem_free

mem_inactive

mem_used

mem_used_percent

net_bytes_recv

net_bytes_sent

net_drop_in

net_drop_out

net_err_in

net_err_out

net_packets_recv

net_packets_sent

netstat_tcp_close

netstat_tcp_close_wait

netstat_tcp_closing

netstat_tcp_established

netstat_tcp_fin_wait1

netstat_tcp_fin_wait2

netstat_tcp_last_ack

netstat_tcp_listen

netstat_tcp_none

netstat_tcp_syn_recv

netstat_tcp_syn_sent

netstat_tcp_time_wait

netstat_udp_socket

processes_blocked

processes_dead

processes_idle

processes_paging

processes_running

processes_sleeping

processes_stopped

processes_total

processes_total_threads

processes_wait

processes_zombies

swap_free

swap_used

swap_used_percent

Elastic Block Store (EBS)

O CloudWatch Application Insights é compatível com as seguintes métricas:

VolumeReadBytes

VolumeWriteBytes

VolumeReadOps

VolumeWriteOps

VolumeTotalReadTime

VolumeTotalWriteTime

VolumeIdleTime

VolumeQueueLength

VolumeThroughputPercentage

VolumeConsumedReadWriteOps

BurstBalance

Amazon Elastic File System (Amazon EFS)

O CloudWatch Application Insights é compatível com as seguintes métricas:

BurstCreditBalance

PercentIOLimit

PermittedThroughput

MeteredIOBytes

TotalIOBytes

DataWriteIOBytes

DataReadIOBytes

MetadataIOBytes

Conexões de clientes

TimeSinceLastSync

StorageBytes

Throughput

PercentageOfPermittedThroughputUtilization

ThroughputIOPS

PercentThroughputDataReadIOByte

PercentThroughputDataWriteIOBytes

PercentageOfIOPSDataReadIOBytes

PercentageOfIOPSDataWriteIOBytes

Tamanho médio de bytes de rádio da área de dados

Tamanho médio de escrita de dados por BytesSize

Elastic Load Balancer (ELB)

O CloudWatch Application Insights é compatível com as seguintes métricas:

EstimatedALBActiveConnectionCount

EstimatedALBConsumedLCUs

EstimatedALBNewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

RequestCount

UnHealthyHostCount

Aplicativo ELB

O CloudWatch Application Insights é compatível com as seguintes métricas:

EstimatedALBActiveConnectionCount

EstimatedALBConsumedLCUs

EstimatedALBNewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

Latência

RequestCount

SurgeQueueLength

UnHealthyHostCount

Grupos do Amazon EC2 Auto Scaling

O CloudWatch Application Insights é compatível com as seguintes métricas:

CPUCreditBalance

CPUCreditUsage

CPUSurplusCreditBalance

CPUSurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBSByteBalance%

EBSIOBalance%

EBSReadBytes

EBSReadOps

EBSWriteBytes

EBSWriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Instance

StatusCheckFailed_System

Amazon Simple Queue Server (SQS)

O CloudWatch Application Insights é compatível com as seguintes métricas:

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

Amazon Relational Database Service (RDS)

O CloudWatch Application Insights é compatível com as seguintes métricas:

Metrics

- Instâncias de bancos de dados do RDS
- Clusters de banco de dados do RDS

Instâncias de bancos de dados do RDS

BurstBalance

CPUCreditBalance

CPUUtilization

DatabaseConnections

DiskQueueDepth

FailedSQLServerAgentJobsCount

FreeStorageSpace

FreeableMemory

NetworkReceiveThroughput

NetworkTransmitThroughput

ReadIOPS

ReadLatency

ReadThroughput

WriteIOPS

WriteLatency

WriteThroughput

Clusters de banco de dados do RDS

ActiveTransactions

AuroraBinlogReplicaLag

AuroraReplicaLag

BackupRetentionPeriodStorageUsed

BinLogDiskUsage

BlockedTransactions

BufferCacheHitRatio

CPUUtilization

CommitLatency

CommitThroughput

DDLLatency

DDLThroughput

DMLLatency

DMLThroughput

DatabaseConnections

Deadlocks

DeleteLatency

DeleteThroughput

EngineUptime

FreeLocalStorage

FreeableMemory

InsertLatency

InsertThroughput

LoginFailures

NetworkReceiveThroughput

NetworkThroughput

NetworkTransmitThroughput

Consultas

ResultSetCacheHitRatio

SelectLatency

SelectThroughput

SnapshotStorageUsed

TotalBackupStorageBilled

UpdateLatency

UpdateThroughput

VolumeBytesUsed

VolumeReadIOPs

VolumeWriteIOPs

Função do AWS Lambda

O CloudWatch Application Insights é compatível com as seguintes métricas:

Erros

DeadLetterErrors

Duração

Controles de utilização

IteratorAge

ProvisionedConcurrencySpilloverInvocations

Tabela do Amazon DynamoDB

O CloudWatch Application Insights é compatível com as seguintes métricas:

SystemErrors

UserErrors

ConsumedReadCapacityUnits

ConsumedWriteCapacityUnits

ReadThrottleEvents

WriteThrottleEvents

TimeToLiveDeletedItemCount

ConditionalCheckFailedRequests

TransactionConflict

ReturnedRecordsCount

PendingReplicationCount

ReplicationLatency

Bucket do Amazon S3

O CloudWatch Application Insights é compatível com as seguintes métricas:

ReplicationLatency

BytesPendingReplication

OperationsPendingReplication

4xxErrors

5xxErrors

AllRequests

GetRequests

PutRequests

DeleteRequests

HeadRequests

PostRequests

SelectRequests

ListRequests

SelectScannedBytes

SelectReturnedBytes

FirstByteLatency

TotalRequestLatency

BytesDownloaded

BytesUploaded

AWS Step Functions

O CloudWatch Application Insights é compatível com as seguintes métricas:

Metrics

- Nível da execução

- Atividade
- Função do Lambda
- Integração de serviços
- API Step Functions

Nível da execução

ExecutionTime

ExecutionThrottled

ExecutionsFailed

ExecutionsTimedOut

ExecutionsAborted

ExecutionsSucceeded

ExecutionsStarted

Atividade

ActivityRunTime

ActivityScheduleTime

ActivityTime

ActivitiesFailed

ActivitiesHeartbeatTimedOut

ActivitiesTimedOut

ActivitiesScheduled

ActivitiesSucceeded

ActivitiesStarted

Função do Lambda

LambdaFunctionRunTime

LambdaFunctionScheduleTime

LambdaFunctionTime

LambdaFunctionsFailed

LambdaFunctionsTimedOut

LambdaFunctionsScheduled

LambdaFunctionsSucceeded

LambdaFunctionsStarted

Integração de serviços

ServiceIntegrationRunTime

ServiceIntegrationScheduleTime

ServiceIntegrationTime

ServiceIntegrationsFailed

ServiceIntegrationsTimedOut

ServiceIntegrationsScheduled

ServiceIntegrationsSucceeded

ServiceIntegrationsStarted

API Step Functions

ThrottledEvents

ProvisionedBucketSize

ProvisionedRefillRate

ConsumedCapacity

Etapas da API REST do API Gateway

O CloudWatch Application Insights é compatível com as seguintes métricas:

4XXError

5XXError

IntegrationLatency

Latência

CacheHitCount

CacheMissCount

SAP HANA

> **Note**
>
> O CloudWatch Application Insights oferece suporte apenas a ambientes SID HANA únicos. Se houver vários SID HANA conectados, o monitoramento será configurado apenas para o primeiro SID detectado.

O CloudWatch Application Insights é compatível com as seguintes métricas:

hanadb_every_service_started_status

hanadb_daemon_service_started_status

hanadb_preprocessor_service_started_status

hanadb_webdispatcher_service_started_status

hanadb_compileserver_service_started_status

hanadb_nameserver_service_started_status

hanadb_server_startup_time_variations_seconds

hanadb_level_5_alerts_count

hanadb_level_4_alerts_count

hanadb_out_of_memory_events_count

hanadb_max_trigger_read_ratio_percent

hanadb_max_trigger_write_ratio_percent

hanadb_log_switch_wait_ratio_percent

hanadb_log_switch_race_ratio_percent

hanadb_time_since_last_savepoint_seconds

hanadb_disk_usage_highlevel_percent

hanadb_max_converter_page_number_count

hanadb_long_running_savepoints_count

hanadb_failed_io_reads_count

hanadb_failed_io_writes_count

hanadb_disk_data_unused_percent

hanadb_current_allocation_limit_used_percent

hanadb_table_allocation_limit_used_percent

hanadb_host_total_physical_memory_mb

hanadb_host_physical_memory_used_mb

hanadb_host_physical_memory_free_mb

hanadb_swap_memory_free_mb

hanadb_swap_memory_used_mb

hanadb_host_allocation_limit_mb

hanadb_host_total_memory_used_mb

hanadb_host_total_peak_memory_used_mb

hanadb_host_total_allocation_limit_mb

hanadb_host_code_size_mb

hanadb_host_shared_memory_allocation_mb

hanadb_cpu_usage_percent

hanadb_cpu_user_percent

hanadb_cpu_system_percent

hanadb_cpu_waitio_percent

hanadb_cpu_busy_percent

hanadb_cpu_idle_percent

hanadb_long_delta_merge_count

hanadb_unsuccessful_delta_merge_count

hanadb_successful_delta_merge_count

hanadb_row_store_allocated_size_mb

hanadb_row_store_free_size_mb

hanadb_row_store_used_size_mb

hanadb_temporary_tables_count

hanadb_large_non_compressed_tables_count

hanadb_total_non_compressed_tables_count

hanadb_longest_running_job_seconds

hanadb_average_commit_time_milliseconds

hanadb_suspended_sql_statements_count

hanadb_plan_cache_hit_ratio_percent

hanadb_plan_cache_lookup_count

hanadb_plan_cache_hit_count

hanadb_plan_cache_total_execution_microseconds

hanadb_plan_cache_cursor_duration_microseconds

hanadb_plan_cache_preparation_microseconds

hanadb_plan_cache_evicted_count

hanadb_plan_cache_evicted_microseconds

hanadb_plan_cache_evicted_preparation_count

hanadb_plan_cache_evicted_execution_count

hanadb_plan_cache_evicted_preparation_microseconds

hanadb_plan_cache_evicted_cursor_duration_microseconds

hanadb_plan_cache_evicted_total_execution_microseconds

hanadb_plan_cache_evicted_plan_size_mb

hanadb_plan_cache_count

hanadb_plan_cache_preparation_count

hanadb_plan_cache_execution_count

hanadb_network_collision_rate

hanadb_network_receive_rate

hanadb_network_transmit_rate

hanadb_network_packet_receive_rate

hanadb_network_packet_transmit_rate

hanadb_network_transmit_error_rate

hanadb_network_receive_error_rate

hanadb_time_until_license_expires_days

hanadb_is_license_valid_status

hanadb_local_running_connections_count

hanadb_local_idle_connections_count

hanadb_remote_running_connections_count

hanadb_remote_idle_connections_count

hanadb_last_full_data_backup_age_days

hanadb_last_data_backup_age_days

hanadb_last_log_backup_age_hours

hanadb_failed_data_backup_past_7_days_count

hanadb_failed_log_backup_past_7_days_count

hanadb_oldest_backup_in_catalog_age_days

hanadb_backup_catalog_size_mb

hanadb_hsr_replication_status

hanadb_hsr_log_shipping_delay_seconds

hanadb_hsr_secondary_failover_count

hanadb_hsr_secondary_reconnect_count

hanadb_hsr_async_buffer_used_mb

hanadb_hsr_secondary_active_status

hanadb_handle_count

hanadb_ping_time_milliseconds

hanadb_connection_count

hanadb_internal_connection_count

hanadb_external_connection_count

hanadb_idle_connection_count

hanadb_transaction_count

hanadb_internal_transaction_count

hanadb_external_transaction_count

hanadb_user_transaction_count

hanadb_blocked_transaction_count

hanadb_statement_count

hanadb_active_commit_id_range_count

hanadb_mvcc_version_count

hanadb_pending_session_count

hanadb_record_lock_count

hanadb_read_count

hanadb_write_count

hanadb_merge_count

hanadb_unload_count

hanadb_active_thread_count

hanadb_waiting_thread_count

hanadb_total_thread_count

hanadb_active_sql_executor_count

hanadb_waiting_sql_executor_count

hanadb_total_sql_executor_count

hanadb_data_write_size_mb

hanadb_data_write_time_milliseconds

hanadb_log_write_size_mb

hanadb_log_write_time_milliseconds

hanadb_data_read_size_mb

hanadb_data_read_time_milliseconds

hanadb_log_read_size_mb

hanadb_log_read_time_milliseconds

hanadb_data_backup_write_size_mb

hanadb_data_backup_write_time_milliseconds

hanadb_log_backup_write_size_mb

hanadb_log_backup_write_time_milliseconds

hanadb_mutex_collision_count

hanadb_read_write_lock_collision_count

hanadb_admission_control_admit_count

hanadb_admission_control_reject_count

hanadb_admission_control_queue_size_mb

hanadb_admission_control_wait_time_milliseconds

SAP ASE

O CloudWatch Application Insights é compatível com as seguintes métricas:

asedb_database_availability

asedb_trunc_log_on_chkpt_enabled

asedb_last_db_backup_age_in_days

asedb_last_transaction_log_backup_age_in_hours

asedb_suspected_database

asedb_db_space_usage_percent

asedb_db_log_space_usage_percent

asedb_locked_login

asedb_has_mixed_log_and_data

asedb_runtime_for_open_transactions

asedb_data_cache_hit_ratio

asedb_data_cache_usage

asedb_sql_cache_hit_ratio

asedb_cache_usage

asedb_run_queue_length

asedb_number_of_rollbacks

asedb_number_of_commits

asedb_number_of_transactions

asedb_outstanding_disk_io

asedb_percent_io_busy

asedb_percent_system_busy

asedb_percent_locks_active

asedb_scheduled_jobs_failed_percent

asedb_user_connections_percent

asedb_query_logical_reads

asedb_query_physical_reads

asedb_query_cpu_time

asedb_query_memory_usage

Alta disponibilidade do SAP ASE no Amazon EC2

O CloudWatch Application Insights é compatível com as seguintes métricas:

asedb_ha_replication_state

asedb_ha_replication_mode

asedb_ha_replication_latency_in_minutes

SAP NetWeaver

O CloudWatch Application Insights é compatível com as seguintes métricas:

Métrica	Descrição
sap_alerts_ResponseTime	O alerta de tempo de resposta do SAP do CCMS (RZ20)>R3services>Dialog>ResponseTime.
sap_alerts_ResponseTimeDialog	O alerta do diálogo de tempo de resposta do SAP do CCMS (RZ20)>R3Services>Dialog>ResponseTimeDialog.
sap_alerts_ResponseTimeDialogRFC	O alerta de tempo de resposta do SAP do CCMS (RZ20)>R3Services> Dialog>ResponseTimeDialogRFC.
sap_alerts_DBRequestTime	O alerta de tempo de resposta do SAP do CCMS (RZ20)>R3Services>Dialog>DBRequestTime.
sap_alerts_FrontendResponseTime	O alerta de tempo de resposta do SAP do CCMS (RZ20)>R3Services > Dialog>FrontEndResponseTime.
sap_alerts_Database	O sistema SAP registrou erros relacionados ao banco de dados. Alerta do SM21 ou do CCMS (RZ20)>R3Syslog>Database.
sap_alerts_QueueTime	O alerta de tempo de fila do SAP do CCMS (RZ20)>R3Services>Dialog>QueueTime.
sap_alerts_AbortedJobs	Trabalhos em segundo plano com falha no sistema SAP. Alerta de (RZ20)>R3Services > Background>AbortedJobs.

Métrica	Descrição
sap_alerts_BasisSystem	O sistema SAP registrou erros no nível do sistema. Alerta do SM21 ou do CCMS (RZ20)>R3Syslog>BasisSystem.
sap_alerts_Security	O sistema SAP registrou mensagens relacionadas a segurança. Alerta do SM21 ou do CCMS ((RZ20)>R3Syslog>Security.
sap_alerts_System	O sistema SAP registrou mensagens relacionadas a segurança ou a auditoria. Alerta do SM21 ou do CCMS (RZ20)>Security>System.
sap_alerts_LongRunners	Existem programas de longa duração no sistema SAP. Alerta do CCMS (RZ20)>R3 Services>Dialog>LongRunners.
sap_alerts_SqlError	Existem logs de erros no nível do cliente do banco de dados SAP. Alerta do CCMS(RZ20)>DatabaseClient>AbapSql>SqlError.
sap_alerts_State	Alerta de estado do CCMS (RZ20)>OS Collector>State.
sap_alerts_Shortdumps	Alerta de shortdumps do ST22 e do CCMS (RZ20)>R3Abap>Shortdumps.
sap_alerts_Availability	Alerta de disponibilidade para instância do servidor de aplicações SAP de SM21, SM50, SM51, SM66 e CCMS (RZ20)>InstanceAsTask>Availability.
sap_dispatcher_queue_high	A função `GetQueueStatistic` do serviço da Web SAPControl fornece a contagem alta da fila do despachante.

Métrica	Descrição
sap_dispatcher_queue_max	A função `GetQueueStatistic` do serviço da Web SAPControl fornece a contagem máxima da fila do despachante.
sap_dispatcher_queue_now	A função `GetQueueStatistic` do serviço da Web SAPControl fornece a contagem no momento da fila do despachante.
sap_dispatcher_queue_reads	A função `GetQueueStatistic` do serviço da Web SAPControl fornece a contagem de leituras da fila do despachante.
sap_dispatcher_queue_writes	A função `GetQueueStatistic` do serviço da Web SAPControl fornece a contagem de gravações da fila do despachante.
sap_enqueue_server_arguments_high	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a contagem alta de argumentos em fila.
sap_enqueue_server_arguments_max	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a contagem máxima de argumentos em fila.
sap_enqueue_server_arguments_now	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a contagem no momento de argumentos em fila.
sap_enqueue_server_arguments_state	A função `EnqGetStatistic` do serviço da Web SAPControl fornece o estado dos argumentos em fila.
sap_enqueue_server_backup_requests	A função `EnqGetStatistic` do serviço da Web SAPControl fornece as solicitações de backup em fila.

Métrica	Descrição
sap_enqueue_server_cleanup_requests	A função `EnqGetStatistic` do serviço da Web SAPControl fornece as solicitações de limpeza em fila.
sap_enqueue_server_dequeue_all_requests	A função `EnqGetStatistic` do serviço da Web SAPControl fornece as solicitações de retirada de todos da fila.
sap_enqueue_server_dequeue_errors	A função `EnqGetStatistic` do serviço da Web SAPControl fornece os erros de retirada da fila.
sap_enqueue_server_dequeue_requests	A função `EnqGetStatistic` do serviço da Web SAPControl fornece todas as solicitações de retirada da fila.
sap_enqueue_server_enqueue_errors	A função `EnqGetStatistic` do serviço da Web SAPControl fornece os erros em fila.
sap_enqueue_server_enqueue_rejects	A função `EnqGetStatistic` do serviço da Web SAPControl fornece as rejeições em fila.
sap_enqueue_server_enqueue_requests	A função `EnqGetStatistic` do serviço da Web SAPControl fornece as solicitações em fila.
sap_enqueue_server_lock_time	A função `EnqGetStatistic` do serviço da Web SAPControl fornece o tempo de bloqueio em fila.
sap_enqueue_server_lock_wait_time	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a espera de bloqueio em fila.
sap_enqueue_server_locks_high	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a contagem alta de bloqueios em fila.

Métrica	Descrição
sap_enqueue_server_locks_max	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a contagem máxima de bloqueios em fila.
sap_enqueue_server_locks_now	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a contagem no momento de bloqueios em fila.
sap_enqueue_server_locks_state	A função `EnqGetStatistic` do serviço da Web SAPControl fornece o estado de bloqueio em fila.
sap_enqueue_server_owner_high	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a contagem alta de proprietários em fila.
sap_enqueue_server_owner_max	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a contagem máxima de proprietários em fila.
sap_enqueue_server_owner_now	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a contagem no momento de proprietários em fila.
sap_enqueue_server_owner_state	A função `EnqGetStatistic` do serviço da Web SAPControl fornece o estado do proprietá rio em fila.
sap_enqueue_server_replication_state	A função `EnqGetStatistic` do serviço da Web SAPControl fornece o estado de replicaçã o em fila.
sap_enqueue_server_reporting_requests	A função `EnqGetStatistic` do serviço da Web SAPControl fornece o status das solicitaç ões de relatórios.

Métrica	Descrição
sap_enqueue_server_server_time	A função `EnqGetStatistic` do serviço da Web SAPControl fornece a hora do servidor em fila.
sap_HA_check_failover_config_state	A função `HACheckFailoverConfig` do serviço da Web SAPControl fornece o status de alta disponibilidade do SAP.
sap_HA_get_failover_config_HAActive	A função `HAGetFailoverConfig` do serviço da Web SAPControl fornece o status e a configuração do cluster de alta disponibilidade do SAP.
sap_start_service_processes	A função `GetProcessList` do serviço da Web SAPControl fornece o status dos processos de disp+work, IGS, gwrd, icman, servidor de mensagens e servidor em fila.

Cluster de HA

O CloudWatch Application Insights é compatível com as seguintes métricas:

ha_cluster_pacemaker_stonith_enabled

ha_cluster_corosync_quorate

hanadb_webdispatcher_service_started_status

ha_cluster_pacemaker_nodes

ha_cluster_corosync_ring_errors

ha_cluster_pacemaker_fail_count

Java

O CloudWatch Application Insights é compatível com as seguintes métricas:

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_threading_daemonthreadcount

java_lang_classloading_loadedclasscount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Amazon Elastic Container Service (Amazon ECS)

O CloudWatch Application Insights é compatível com as seguintes métricas:

Metrics

- Métricas integradas do CloudWatch
- Métricas do Container Insights
- Métricas do Container Insights Prometheus

Métricas integradas do CloudWatch

CPUReservation

CPUUtilization

MemoryReservation

MemoryUtilization

GPUReservation

Métricas do Container Insights

ContainerInstanceCount

CpuUtilized

CpuReserved

DeploymentCount

DesiredTaskCount

MemoryUtilized

MemoryReserved

NetworkRxBytes

NetworkTxBytes

PendingTaskCount

RunningTaskCount

ServiceCount

StorageReadBytes

StorageWriteBytes

TaskCount

TaskSetCount

instance_cpu_limit

instance_cpu_reserved_capacity

instance_cpu_usage_total

instance_cpu_utilization

instance_filesystem_utilization

instance_memory_limit

instance_memory_reserved_capacity

instance_memory_utilization

instance_memory_working_set

instance_network_total_bytes

instance_number_of_running_tasks

Métricas do Container Insights Prometheus

Métricas do Java JMX

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_classloading_loadedclasscount

java_lang_threading_daemonthreadcount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Kubernetes na AWS

O CloudWatch Application Insights é compatível com as seguintes métricas:

Metrics

- Métricas do Container Insights
- Métricas do Container Insights Prometheus

Métricas do Container Insights

cluster_failed_node_count

cluster_node_count

namespace_number_of_running_pods

node_cpu_limit

node_cpu_reserved_capacity

node_cpu_usage_total

node_cpu_utilization

node_filesystem_utilization

node_memory_limit

node_memory_reserved_capacity

node_memory_utilization

node_memory_working_set

node_network_total_bytes

node_number_of_running_containers

node_number_of_running_pods

pod_cpu_reserved_capacity

pod_cpu_utilization

pod_cpu_utilization_over_pod_limit

pod_memory_reserved_capacity

pod_memory_utilization

pod_memory_utilization_over_pod_limit

pod_network_rx_bytes

pod_network_tx_bytes

service_number_of_running_pods

Métricas do Container Insights Prometheus

Métricas do Java JMX

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_classloading_loadedclasscount

java_lang_threading_daemonthreadcount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Amazon FSx

O CloudWatch Application Insights é compatível com as seguintes métricas:

DataReadBytes

DataWriteBytes

DataReadOperations

DataWriteOperations

MetadataOperations

FreeStorageCapacity

FreeDataStorageCapacity

LogicalDiskUsage

PhysicalDiskUsage

Amazon VPC

O CloudWatch Application Insights é compatível com as seguintes métricas:

NetworkAddressUsage

Uso de endereço de rede emparelhado

VPCFirewallQueryVolume

Gateways de NAT da Amazon VPC

O CloudWatch Application Insights é compatível com as seguintes métricas:

ErrorPortAllocation

IdleTimeoutCount

Verificação de integridade do Amazon Route 53

O CloudWatch Application Insights é compatível com as seguintes métricas:

ChildHealthCheckHealthyCount

ConnectionTime

HealthCheckPercentageHealthy

HealthCheckStatus

SSLHandshakeTime

TimeToFirstByte

Zona hospedada do Amazon Route 53

O CloudWatch Application Insights é compatível com as seguintes métricas:

DNSQueries

DNSSecInternalFailure

DNSSECKeySigningKeysNeedingAction

DNSSECKeySigningKeyMaxNeedingActionAge

DNSSECKeySigningKeyAge

Endpoint do Amazon Route 53 Resolver

O CloudWatch Application Insights é compatível com as seguintes métricas:

EndpointHealthyENICount

EndpointUnHealthyENICount

InboundQueryVolume

OutboundQueryVolume

OutboundQueryAggregateVolume

Grupo de regras do AWS Network Firewall

O CloudWatch Application Insights é compatível com as seguintes métricas:

FirewallRuleGroupQueryVolume

associação do grupos de regras AWS Network Firewall

O CloudWatch Application Insights é compatível com as seguintes métricas:

FirewallRuleGroupVpcQueryVolume

Métricas com requisitos de pontos de dados

Para métricas sem um limite padrão óbvio para a criação de alarmes, o Application Insights aguarda até que a métrica tenha pontos de dados suficientes para prever um limite razoável para criar o alarme. Os requisitos de pontos de dados da métrica que o CloudWatch Application Insights verifica antes da criação de um alarme são:

- A métrica tem pelo menos 100 pontos de dados dos últimos 15 dias aos últimos 2 dias.
- A métrica tem pelo menos 100 pontos de dados do último dia.

As métricas a seguir seguem esses requisitos de pontos de dados. Observe que as métricas do atendente do CloudWatch exigem até uma hora para criar alarmes.

Metrics

- AWS/ApplicationELB
- AWS/AutoScaling
- AWS/EC2
- Elastic Block Store (EBS)
- AWS/ELB
- AWS/RDS
- AWS/Lambda

- AWS/SQS
- AWS/CWAgent
- AWS/DynamoDB
- AWS/S3
- AWS/States
- AWS/ApiGateway
- AWS/SNS

AWS/ApplicationELB

ActiveConnectionCount

ConsumedLCUs

HTTPCode_ELB_4XX_Count

HTTPCode_Target_2XX_Count

HTTPCode_Target_3XX_Count

HTTPCode_Target_4XX_Count

HTTPCode_Target_5XX_Count

NewConnectionCount

ProcessedBytes

TargetResponseTime

UnHealthyHostCount

AWS/AutoScaling

GroupDesiredCapacity

GroupInServiceInstances

GroupMaxSize

GroupMinSize

GroupPendingInstances

GroupStandbyInstances

GroupTerminatingInstances

GroupTotalInstances

AWS/EC2

CPUCreditBalance

CPUCreditUsage

CPUSurplusCreditBalance

CPUSurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBSByteBalance%

EBSIOBalance%

EBSReadBytes

EBSReadOps

EBSWriteBytes

EBSWriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

Elastic Block Store (EBS)

VolumeReadBytes

VolumeWriteBytes

VolumeReadOps

VolumeWriteOps

VolumeTotalReadTime

VolumeTotalWriteTime

VolumeIdleTime

VolumeQueueLength

VolumeThroughputPercentage

VolumeConsumedReadWriteOps

BurstBalance

AWS/ELB

EstimatedALBActiveConnectionCount

EstimatedALBConsumedLCUs

EstimatedALBNewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

Latência

RequestCount

SurgeQueueLength

UnHealthyHostCount

AWS/RDS

ActiveTransactions

AuroraBinlogReplicaLag

AuroraReplicaLag

BackupRetentionPeriodStorageUsed

BinLogDiskUsage

BlockedTransactions

CPUCreditBalance

CommitLatency

CommitThroughput

DDLLatency

DDLThroughput

DMLLatency

DMLThroughput

DatabaseConnections

Deadlocks

DeleteLatency

DeleteThroughput

DiskQueueDepth

EngineUptime

FreeLocalStorage

FreeStorageSpace

FreeableMemory

InsertLatency

InsertThroughput

LoginFailures

NetworkReceiveThroughput

NetworkThroughput

NetworkTransmitThroughput

Consultas

ReadIOPS

ReadThroughput

SelectLatency

SelectThroughput

SnapshotStorageUsed

TotalBackupStorageBilled

UpdateLatency

UpdateThroughput

VolumeBytesUsed

VolumeReadIOPs

VolumeWriteIOPs

WriteIOPS

WriteThroughput

AWS/Lambda

Erros

DeadLetterErrors

Duração

Controles de utilização

IteratorAge

ProvisionedConcurrencySpilloverInvocations

AWS/SQS

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

AWS/CWAgent

% de espaço livre de disco lógico

% de bytes confirmados em uso na memória

Mbytes de memória disponíveis

Total/s de bytes de interface de rede

% de uso de arquivo de paginação

% de tempo de disco de disco físico

Média de disco físico S/leitura de disco

Média de disco físico S/gravação de disco

Bytes/s de leitura de disco do disco físico

Leituras/s de disco do disco físico

Bytes/s de gravação de disco do disco físico

Gravações/s de disco do disco físico

% de tempo ocioso do processador

% de tempo de interrupção do processador

% de tempo de processador do processador

% de tempo de usuário do processador

SQLServer: gravações/s encaminhadas dos métodos de acesso

SQLServer: divisões/s da página de métodos de acesso

SQLServer: proporção de acertos do cache do buffer do gerenciador de buffer

SQLServer: expectativa de vida da página do gerenciador de buffer

SQLServer: bytes de arquivo de réplica de banco de dados recebidos por segundo

SQLServer: bytes de log de réplica de banco de dados recebidos por segundo

SQLServer: log de réplica do banco de dados restante para desfazer

SQLServer: fila de envio de log de réplica de banco de dados

SQLServer: transações de gravação espelhada de réplica de banco de dados por segundo

SQLServer: fila de recuperação de réplica de banco de dados

SQLServer: bytes para refazer réplica do banco de dados restante

SQLServer: bytes da réplica de banco de dados refeita por segundo

SQLServer: log total da réplica de banco de dados que deve ser desfeita

SQLServer: atraso da transação de réplica de banco de dados

SQLServer: processos de estatísticas gerais bloqueados

SQLServer: solicitações em lote/s de estatísticas do SQL

SQLServer: compilações do SQL/s de estatísticas do SQL

SQLServer: novas compilações do SQL/s de estatísticas do SQL

Tamanho da fila do processador do sistema

Conexões TCPv4 estabelecidas

Conexões TCPv6 estabelecidas

AWS/DynamoDB

ConsumedReadCapacityUnits

ConsumedWriteCapacityUnits

ReadThrottleEvents

WriteThrottleEvents

TimeToLiveDeletedItemCount

ConditionalCheckFailedRequests

TransactionConflict

ReturnedRecordsCount

PendingReplicationCount

ReplicationLatency

AWS/S3

ReplicationLatency

BytesPendingReplication

OperationsPendingReplication

4xxErrors

5xxErrors

AllRequests

GetRequests

PutRequests

DeleteRequests

HeadRequests

PostRequests

SelectRequests

ListRequests

SelectScannedBytes

SelectReturnedBytes

FirstByteLatency

TotalRequestLatency

BytesDownloaded

BytesUploaded

AWS/States

ActivitiesScheduled

ActivitiesStarted

ActivitiesSucceeded

ActivityScheduleTime

AtivityRuntime

ActivityTime

LambdaFunctionsScheduled

LambdaFunctionsStarted

LambdaFunctionsSucceeded

LambdaFunctionScheduleTime

LambdaFunctionRuntime

LambdaFunctionTime

ServiceIntegrationsScheduled

ServiceIntegrationsStarted

ServiceIntegrationsSucceeded

ServiceIntegrationScheduleTime

ServiceIntegrationRuntime

ServiceIntegrationTime

ProvisionedRefillRate

ProvisionedBucketSize

ConsumedCapacity

ThrottledEvents

AWS/ApiGateway

4XXError

IntegrationLatency

Latência

DataProcessed

CacheHitCount

CacheMissCount

AWS/SNS

NumberOfNotificationsDelivered

NumberOfMessagesPublished

NumberOfNotificationsFailed

NumberOfNotificationsFilterredOut

NumberOfNotificationsFilterEredout-InvalidAttributes

NumberOfNotificationsFilterredout-NomessageAttributes

NumberOfNotificationsRedrivenToDlq

NumberOfNotificationsFailedToRedriveToDlq

SMSSuccessRate

Métricas recomendadas

A tabela a seguir lista as métricas recomendadas para cada tipo de componente.

Tipo de componente	Tipo de workload	Métrica recomendada
Instância do EC2 (servidores Windows)	Padrão/personalizada	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória % de espaço livre de disco lógico Mbytes de memória disponíveis
	Active Directory	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória Mbytes de memória disponíveis Banco de dados ==> % de acertos de cache do banco de dados de instâncias Operações de replicação pendentes do DRA do DirectoryServices

Tipo de componente	Tipo de workload	Métrica recomendada
		Sincronizações de replicação pendentes do DRA do DirectoryServices Falhas de consultas recursivas do DNS/s Média de LogicalDisk Comprimento da fila de discos

Tipo de componente	Tipo de workload	Métrica recomendada
	Aplicação Java	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória Mbytes de memória disponíveis java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Tipo de componente	Tipo de workload	Métrica recomendada
	Microsoft IIS/.NET Web Front-End	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória Mbytes de memória disponíveis Nº de exceções do CLR .NET de exceções ocorridas por segundo Nº total de bytes confirmados de memória do CLR .NET % de tempo de memória do CLR .NET em GC Solicitações de aplicações ASP.NET na fila de aplicações Solicitações do ASP.NET em fila A aplicação ASP.NET é reiniciada

Tipo de componente	Tipo de workload	Métrica recomendada
	Camada do banco de dados do Microsoft SQL Server	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória Mbytes de memória disponíveis % de uso de arquivo de paginação Tamanho da fila do processador do sistema Total/s de bytes de interface de rede % de tempo de disco de disco físico SQLServer: proporção de acertos do cache do buffer do gerenciador de buffer SQLServer: expectativa de vida da página do gerenciador de buffer SQLServer: processos de estatísticas gerais bloqueados SQLServer: conexões de usuário de estatísticas gerais

Tipo de componente	Tipo de workload	Métrica recomendada
		SQLServer: bloqueia o número de bloqueios/s SQLServer: solicitações em lote/s de estatísticas do SQL
	MySQL	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória % de espaço livre de disco lógico Mbytes de memória disponíveis

Tipo de componente	Tipo de workload	Métrica recomendada
	Workerpool .NET /nível intermediário	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória Mbytes de memória disponíveis Nº de exceções do CLR .NET de exceções ocorridas por segundo Nº total de bytes confirmados de memória do CLR .NET % de tempo de memória do CLR .NET em GC
	Nível do núcleo do .NET	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória Mbytes de memória disponíveis

Tipo de componente	Tipo de workload	Métrica recomendada
	Oracle	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória % de espaço livre de disco lógico Mbytes de memória disponíveis
	Postgres	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória % de espaço livre de disco lógico Mbytes de memória disponíveis

Tipo de componente	Tipo de workload	Métrica recomendada
	SharePoint	CPUUtilization StatusCheckFailed % de tempo de processador do processador % de bytes confirmados em uso na memória Mbytes de memória disponíveis Recortes de API do cache de aplicações ASP.NET Solicitações de ASP.NET rejeitadas Processo de operador de ASP.NET é reiniciado Páginas de memória/s Cache de publicação do SharePoint Liberações de cache de publicações/segundo Solicitação de tempo de execução/página do SharePoint Foundation Número total de compactações de cache do cache baseado em disco do SharePoint

Tipo de componente	Tipo de workload	Métrica recomendada
		Taxa de acertos de cache de blob do cache baseado em disco do SharePoint Taxa de enchimento do cache de blob do cache baseado em disco do SharePoint Liberações de cache de blob do cache baseado em disco do SharePoint/segundo Solicitações do ASP.NET em fila Solicitações de aplicações ASP.NET na fila de aplicações A aplicação ASP.NET é reiniciada Média de LogicalDisk S/ gravação de disco Média de LogicalDisk S/leitura de disco % de tempo de interrupção do processador
Instância do EC2 (servidores Linux)	Padrão/personalizada	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Tipo de componente	Tipo de workload	Métrica recomendada
	Aplicação Java	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize
	Nível de núcleo do .NET ou nível de banco de dados do SQL Server	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Tipo de componente	Tipo de workload	Métrica recomendada
	Oracle	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent
	Postgres	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Tipo de componente	Tipo de workload	Métrica recomendada
Grupo de instâncias do EC2	Nó único ou com vários nós do SAP HANA	• hanadb_server_startup_time_variations_seconds • hanadb_level_5_alerts_count • hanadb_level_4_alerts_count • hanadb_out_of_memory_events_count • hanadb_max_trigger_read_ratio_percent • hanadb_max_trigger_write_ratio_percent • hanadb_log_switch_race_ratio_percent • hanadb_time_since_last_savepoint_seconds • hanadb_disk_usage_highlevel_percent • hanadb_current_allocation_limit_used_percent • hanadb_table_allocation_limit_used_percent • hanadb_cpu_usage_percent • hanadb_plan_cache_hit_ratio_percent • hanadb_last_data_backup_age_days

Tipo de componente	Tipo de workload	Métrica recomendada
Volume do EBS	Any	VolumeReadBytes VolumeWriteBytes VolumeReadOps VolumeWriteOps VolumeQueueLength VolumeThroughputPercentage VolumeConsumedRead WriteOps BurstBalance
Classic ELB	Any	HTTPCode_Backend_4XX HTTPCode_Backend_5XX Latência SurgeQueueLength UnHealthyHostCount
Aplicativo ELB	Any	HTTPCode_Target_4X X_Count HTTPCode_Target_5X X_Count TargetResponseTime UnHealthyHostCount

Tipo de componente	Tipo de workload	Métrica recomendada
Instância de banco de dados do RDS	Any	CPUUtilization ReadLatency WriteLatency BurstBalance FailedSQLServerAgentJobsCount
Cluster de banco de dados do RDS	Any	CPUUtilization CommitLatency DatabaseConnections Deadlocks FreeableMemory NetworkThroughput VolumeBytesUsed
Função do Lambda	Any	Duração Erros IteratorAge ProvisionedConcurrencySpilloverInvocations Controles de utilização

Tipo de componente	Tipo de workload	Métrica recomendada
Fila do SQS	Any	ApproximateAgeOfOldestMessage ApproximateNumberOfMessagesVisible NumberOfMessagesSent
Tabela do Amazon DynamoDB	Any	SystemErrors UserErrors ConsumedReadCapacityUnits ConsumedWriteCapacityUnits ReadThrottleEvents WriteThrottleEvents ConditionalCheckFailedRequests TransactionConflict

Tipo de componente	Tipo de workload	Métrica recomendada
Bucket do Amazon S3	Any	Se a configuração de replicação com o Replication Time Control (RTC) estiver habilitada: ReplicationLatency BytesPendingReplication OperationsPendingReplication Se as métricas de solicitação estiverem ativadas: 5xxErrors 4xxErrors BytesDownloaded BytesUploaded

Tipo de componente	Tipo de workload	Métrica recomendada
AWS Step Functions	Any	Geral • ExecutionThrottled • ExecutionsAborted • ProvisionedBucketSize • ProvisionedRefillRate • ConsumedCapacity Se o tipo de máquina de estado for **`EXPRESS`** ou nível do grupo de log for **`OFF`** • ExecutionsFailed • ExecutionsTimedOut Se a máquina de estado tiver funções do Lambda • LambdaFunctionsFailed • LambdaFunctionsTimedOut Se a máquina de estado tiver atividades • ActivitiesFailed • ActivitiesTimedOut • ActivitiesHeartbea tTimedOut Se a máquina de estado tiver integrações de serviço • ServiceIntegrationsFailed

Tipo de componente	Tipo de workload	Métrica recomendada
		• ServiceIntegrationsTimedOut
Etapa da API REST do API Gateway	Any	• 4XXErrors • 5XXErrors • Latência

Tipo de componente	Tipo de workload	Métrica recomendada
Cluster do ECS	Any	CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes CPUReservation (somente tipo de inicialização do EC2) CPUUtilization (somente tipo de inicialização do EC2) MemoryReservation (somente tipo de inicialização do EC2) MemoryUtilization (somente tipo de inicialização do EC2) GPUReservation (somente tipo de inicialização do EC2) instance_cpu_utilization (somente tipo de inicialização do EC2) instance_filesystem_utilization (somente tipo de inicialização do EC2)

Tipo de componente	Tipo de workload	Métrica recomendada
		instance_memory_utilization (somente tipo de inicialização do EC2) instance_network_total_bytes (somente tipo de inicialização do EC2)

Tipo de componente	Tipo de workload	Métrica recomendada
	Aplicação Java	CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes CPUReservation (somente tipo de inicialização do EC2) CPUUtilization (somente tipo de inicialização do EC2) MemoryReservation (somente tipo de inicialização do EC2) MemoryUtilization (somente tipo de inicialização do EC2) GPUReservation (somente tipo de inicialização do EC2) instance_cpu_utilization (somente tipo de inicialização do EC2) instance_filesystem_utilization (somente tipo de inicialização do EC2)

Tipo de componente	Tipo de workload	Métrica recomendada
		instance_memory_utilization (somente tipo de inicialização do EC2) instance_network_total_bytes (somente tipo de inicialização do EC2) java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Tipo de componente	Tipo de workload	Métrica recomendada
Serviço do ECS	Any	CPUUtilization
		MemoryUtilization
		CpuUtilized
		MemoryUtilized
		NetworkRxBytes
		NetworkTxBytes
		RunningTaskCount
		PendingTaskCount
		StorageReadBytes
		StorageWriteBytes

Tipo de componente	Tipo de workload	Métrica recomendada
	Aplicação Java	CPUUtilization MemoryUtilization CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Tipo de componente	Tipo de workload	Métrica recomendada
Cluster do EKS	Any	cluster_failed_node_count
		node_cpu_reserved_capacity
		node_cpu_utilization
		node_filesystem_utilization
		node_memory_reserv ed_capacity
		node_memory_utilization
		node_network_total_bytes
		pod_cpu_reserved_capacity
		pod_cpu_utilization
		pod_cpu_utilization_over_po d_limit
		pod_memory_reserve d_capacity
		pod_memory_utilization
		pod_memory_utilization_over _pod_limit
		pod_network_rx_bytes
		pod_network_tx_bytes

Tipo de componente	Tipo de workload	Métrica recomendada
	Aplicação Java	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserved_capacity node_memory_utilization node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_utilization pod_cpu_utilization_over_pod_limit pod_memory_reserved_capacity pod_memory_utilization pod_memory_utilization_over_pod_limit pod_network_rx_bytes pod_network_tx_bytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount

Tipo de componente	Tipo de workload	Métrica recomendada
		java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Tipo de componente	Tipo de workload	Métrica recomendada
Cluster do Kubernetes no EC2	Any	cluster_failed_node_count
		node_cpu_reserved_capacity
		node_cpu_utilization
		node_filesystem_utilization
		node_memory_reserv ed_capacity
		node_memory_utilization
		node_network_total_bytes
		pod_cpu_reserved_capacity
		pod_cpu_utilization
		pod_cpu_utilization_over_po d_limit
		pod_memory_reserve d_capacity
		pod_memory_utilization
		pod_memory_utilization_over _pod_limit
		pod_network_rx_bytes
		pod_network_tx_bytes

Tipo de componente	Tipo de workload	Métrica recomendada
	Aplicação Java	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserved_capacity node_memory_utilization node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_utilization pod_cpu_utilization_over_pod_limit pod_memory_reserved_capacity pod_memory_utilization pod_memory_utilization_over_pod_limit pod_network_rx_bytes pod_network_tx_bytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount

Tipo de componente	Tipo de workload	Métrica recomendada
		java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

A tabela a seguir lista os processos e as métricas de processo recomendados para cada tipo de componente. O CloudWatch Application Insights não recomenda o monitoramento de processos que não são executados em uma instância.

Tipo de componente	Tipo de workload	Processo recomendado	Métrica recomendada

Instância do EC2 (servidores Windows)	Microsoft IIS/.NET Web Front-End	`w3wp`	`procstat cpu_usage ,`
			
			`procstat memory_rss ,`
			
			`procstat memory_vms ,`
			
			`procstat read_bytes ,`
			
			`procstat write_bytes`

	Camada do banco de dados do Microsoft SQL Server	`SQLAgent`	`procstat cpu_usage ,`

Tipo de componente	Tipo de workload	Processo recomendado	Métrica recomendada
			`procstat memory_rss` , `procstat memory_vms` , `procstat read_bytes` , `procstat write_bytes`
		`sqlservr`	`procstat cpu_usage` , `procstat memory_rss` , `procstat memory_vms` , `procstat read_bytes` , `procstat write_bytes`
		`sqlwriter`	`procstat cpu_usage` , `procstat memory_rss`

Tipo de componente	Tipo de workload	Processo recomendado	Métrica recomendada
		`ReportingServicesService`	`procstat cpu_usage ,` `procstat memory_rss`

		`MsDtsServr`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

		`Msmdsrv`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

Tipo de componente	Tipo de workload	Processo recomendado	Métrica recomendada
	Workerpool .NET / nível intermediário	`w3wp`	`procstat cpu_usage` , `procstat memory_rss` , `procstat memory_vms` , `procstat read_bytes` , `procstat write_bytes`
	Nível do núcleo do .NET	`w3wp`	`procstat cpu_usage` , `procstat memory_rss` , `procstat memory_vms` , `procstat read_bytes` , `procstat write_bytes`

Métricas de contador de performance

As métricas do Contador de performance são recomendadas para instâncias somente quando os conjuntos de Contador de performance correspondentes são instalados nas instâncias do Windows.

Nome da métrica de contador de performance	Nome do conjunto de contadores de performance
Nº de exceções do CLR .NET de exceções ocorridas	Exceções do CLR .NET
Nº de exceções do CLR .NET de exceções ocorridas por segundo	Exceções do CLR .NET
Nº de exceções do CLR .NET de filtros por segundo	Exceções do CLR .NET
Nº de exceções do CLR .NET de conclusões por segundo	Exceções do CLR .NET
Exceções do CLR .NET lançadas para capturar profundidade por segundo	Exceções do CLR .NET
Nº de interoperabilidade do CLR .NET de CCWs	Interoperabilidade do CLR .NET
Nº de interoperabilidade do CLR .NET de stubs	Interoperabilidade do CLR .NET
Nº de interoperabilidade do CLR .NET de exportações TLB/s	Interoperabilidade do CLR .NET
Nº de interoperabilidade do CLR .NET de importações TLB/s	Interoperabilidade do CLR .NET
Nº de interoperabilidade do CLR .NET de marshaling	Interoperabilidade do CLR .NET
% de tempo de Jit do CLR .NET no Jit	.NET CLR Jit
Falhas do Jit padrão do Jit do CLR .NET	.NET CLR Jit
% de tempo de carregamento do CLR .NET	CLR do .NET carregando
Taxa de carregamento do CLR .NET de falhas de carga	CLR do .NET carregando

Nome da métrica de contador de performance	Nome do conjunto de contadores de performance
Taxa de contenção/s LocksAndThreads do CLR .NET	LocksAndThreads do CLR .NET
Tamanho da fila/s LocksAndThreads do CLR .NET	LocksAndThreads do CLR .NET
Nº total de bytes confirmados de memória do CLR .NET	Memória do CLR .NET
% de tempo de memória do CLR .NET em GC	Memória do CLR .NET
Tempo médio da fila HttpWebRequest de rede 4.0.0.0 do CLR .NET	Rede do CLR .NET 4.0.0.0
HttpWebRequests canceladas/s de rede 4.0.0.0 do CLR .NET	Rede do CLR .NET 4.0.0.0
HttpWebRequests com falha/s de rede 4.0.0.0 do CLR .NET	Rede do CLR .NET 4.0.0.0
HttpWebRequests colocadas na fila por segundo de rede 4.0.0.0 do CLR .NET	Rede do CLR .NET 4.0.0.0
Total de falhas de ping do processo do operador APP_POOL_WAS	APP_POOL_WAS
A aplicação ASP.NET é reiniciada	ASP.NET
Solicitações de ASP.NET rejeitadas	ASP.NET
Processo de operador de ASP.NET é reiniciado	ASP.NET
Cortes de API do cache de aplicações ASP.NET	Aplicativos ASP.NET
% do tempo de processador gerenciado de aplicações ASP.NET (estimado)	Aplicativos ASP.NET

Nome da métrica de contador de performance	Nome do conjunto de contadores de performance
Total de erros de aplicações ASP.NET por segundo	Aplicativos ASP.NET
Erros de aplicações ASP.NET não processados durante a execução/s	Aplicativos ASP.NET
Solicitações de aplicações ASP.NET na fila de aplicações	Aplicativos ASP.NET
Solicitações de aplicações ASP.NET por segundo	Aplicativos ASP.NET
Tempo de espera de solicitação do ASP.NET	ASP.NET
Solicitações do ASP.NET em fila	ASP.NET
Banco de dados ==> % de acertos de cache do banco de dados de instâncias	Banco de dados ==> Instâncias
Banco de dados ==> latência média das leituras do banco de dados de E/S de instâncias	Banco de dados ==> Instâncias
Banco de dados ==> Leituras do banco de dados de E/S de instâncias/s	Banco de dados ==> Instâncias
Banco de dados ==> Latência média de gravações do log de E/S de instâncias	Banco de dados ==> Instâncias
Operações de replicação pendentes do DRA do DirectoryServices	DirectoryServices
Sincronizações de replicação pendentes do DRA do DirectoryServices	DirectoryServices
Tempo de vinculação de LDAP do Directory Services	DirectoryServices

Nome da métrica de contador de performance	Nome do conjunto de contadores de performance
Consultas recursivas do DNS/s	DNS
Falhas de consultas recursivas do DNS/s	DNS
Consultas de TCP do DNS recebidas/s	DNS
Total de consultas do DNS recebidas/s	DNS
Total de respostas do DNS enviadas/s	DNS
Consultas de UDP do DNS recebidas/s	DNS
Filas de solicitação de serviço HTTP CurrentQueueSize	Filas de solicitação de serviço HTTP
% de espaço livre de disco lógico	LogicalDisk
Média de LogicalDisk S/gravação de disco	LogicalDisk
Média de LogicalDisk S/leitura de disco	LogicalDisk
Média de LogicalDisk Comprimento da fila de discos	LogicalDisk
% de bytes confirmados em uso na memória	Memória
Mbytes de memória disponíveis	Memória
Páginas de memória/s	Memória
Tempo (s) de vida média do cache em espera de longo prazo da memória	Memória
Total/s de bytes de interface de rede	Interface de rede
Bytes de interface de rede recebidos/s	Interface de rede
Bytes de interface de rede enviados/s	Interface de rede

Nome da métrica de contador de performance	Nome do conjunto de contadores de performance
Largura de banda atual da Interface de Rede	Interface de rede
% de uso de arquivo de paginação	Arquivo de paginação
% de tempo de disco de disco físico	PhysicalDisk
Média de disco físico Comprimento da fila de discos	PhysicalDisk
Média de disco físico S/leitura de disco	PhysicalDisk
Média de disco físico S/gravação de disco	PhysicalDisk
Bytes/s de leitura de disco do disco físico	PhysicalDisk
Leituras/s de disco do disco físico	PhysicalDisk
Bytes/s de gravação de disco do disco físico	PhysicalDisk
Gravações/s de disco do disco físico	PhysicalDisk
% de tempo ocioso do processador	Processador
% de tempo de interrupção do processador	Processador
% de tempo de processador do processador	Processador
% de tempo de usuário do processador	Processador
Taxa de enchimento do cache de blob do cache baseado em disco do SharePoint	Cache baseado em disco do SharePoint
Liberações de cache de blob do cache baseado em disco do SharePoint/segundo	Cache baseado em disco do SharePoint
Taxa de acertos de cache de blob do cache baseado em disco do SharePoint	Cache baseado em disco do SharePoint

Nome da métrica de contador de performance	Nome do conjunto de contadores de performance
Número total de compactações de cache do cache baseado em disco do SharePoint	Cache baseado em disco do SharePoint
Solicitação de tempo de execução/página do SharePoint Foundation	SharePoint Foundation
Cache de publicação do SharePoint Liberações de cache de publicações/segundo	Cache de publicações do SharePoint
Autenticações Kerberos de estatísticas abrangendo todo o sistema de segurança	Estatísticas abrangendo todo o sistema de segurança
Autenticações NTLM de estatísticas abrangendo todo o sistema de segurança	Estatísticas abrangendo todo o sistema de segurança
SQLServer: gravações/s encaminhadas dos métodos de acesso	SQLServer: métodos de acesso
SQLServer: verificações completas dos métodos de acesso/s	SQLServer: métodos de acesso
SQLServer: divisões/s da página de métodos de acesso	SQLServer: métodos de acesso
SQLServer: proporção de acertos do cache do buffer do gerenciador de buffer	SQLServer: gerenciador de buffer
SQLServer: expectativa de vida da página do gerenciador de buffer	SQLServer: gerenciador de buffer
SQLServer: bytes de arquivo de réplica de banco de dados recebidos por segundo	SQLServer: réplica de banco de dados
SQLServer: bytes de log de réplica de banco de dados recebidos por segundo	SQLServer: réplica de banco de dados

Nome da métrica de contador de performance	Nome do conjunto de contadores de performance
SQLServer: log de réplica do banco de dados restante para desfazer	SQLServer: réplica de banco de dados
SQLServer: fila de envio de log de réplica de banco de dados	SQLServer: réplica de banco de dados
SQLServer: transações de gravação espelhada de réplica de banco de dados por segundo	SQLServer: réplica de banco de dados
SQLServer: fila de recuperação de réplica de banco de dados	SQLServer: réplica de banco de dados
SQLServer: bytes para refazer réplica do banco de dados restante	SQLServer: réplica de banco de dados
SQLServer: bytes da réplica de banco de dados refeita por segundo	SQLServer: réplica de banco de dados
SQLServer: log total da réplica de banco de dados que deve ser desfeita	SQLServer: réplica de banco de dados
SQLServer: atraso da transação de réplica de banco de dados	SQLServer: réplica de banco de dados
SQLServer: processos de estatísticas gerais bloqueados	SQLServer: estatísticas gerais
SQLServer: conexões de usuário de estatísticas gerais	SQLServer: estatísticas gerais
SQLServer: trava o tempo médio de espera (ms)	SQLServer: travas
SQLServer: bloqueia o tempo médio de espera (ms)	SQLServer: bloqueios

Nome da métrica de contador de performance	Nome do conjunto de contadores de performance
SQLServer: bloqueia os tempos limite de bloqueio/s	SQLServer: bloqueios
SQLServer: bloqueia a espera de bloqueio/s	SQLServer: bloqueios
SQLServer: bloqueia o número de bloqueios/s	SQLServer: bloqueios
SQLServer: gerenciador de memória concessão de memória pendente	SQLServer: gerenciador de memória
SQLServer: solicitações em lote/s de estatísticas do SQL	SQLServer: estatísticas de SQL
SQLServer: compilações do SQL/s de estatísticas do SQL	SQLServer: estatísticas de SQL
SQLServer: novas compilações do SQL/s de estatísticas do SQL	SQLServer: estatísticas de SQL
Tamanho da fila do processador do sistema	Sistema
Conexões TCPv4 estabelecidas	TCPv4
Conexões TCPv6 estabelecidas	TCPv6
Descargas de cache de arquivos W3SVC_W3WP	W3SVC_W3WP
Erros de cache de arquivos W3SVC_W3WP	W3SVC_W3WP
Solicitações W3SVC_W3WP/s	W3SVC_W3WP
Descargas de cache URI W3SVC_W3WP	W3SVC_W3WP
Erros de cache URI W3SVC_W3WP	W3SVC_W3WP
Bytes de serviço web recebidos por segundo	Serviço web

Nome da métrica de contador de performance	Nome do conjunto de contadores de performance
Bytes de serviço web enviados por segundo	Serviço web
Tentativas de conexão de web service/s	Serviço web
Conexões atuais de serviços web	Serviço web
Solicitações para obter web service/s	Serviço web
Solicitações para publicar web service/s	Serviço web

Usar a visualização de integridade de recursos no console do CloudWatch

É possível usar a visualização de integridade de recursos para detectar, gerenciar e exibir automaticamente a integridade e a performance dos hosts em suas aplicações em uma única visualização. Você pode visualizar a integridade de seus hosts por uma dimensão de performance, como CPU ou memória, e cortar e cortar centenas de hosts em uma única visualização usando filtros. É possível filtrar por etiquetas ou por casos de uso, como hosts no mesmo grupo do Auto Scaling ou hosts que usam o mesmo balanceador de carga.

Pré-requisitos

Para garantir que o benefício completo do modo de integridade do recurso, verifique se você conta com os pré-requisitos a seguir.

- Para ver a utilização da memória de seus hosts e usá-la como um filtro, é necessário instalar o agente do CloudWatch em seus hosts e configurá-lo para enviar uma métrica de memória ao CloudWatch no namespace `CWAgent`. Nas instâncias do Linux e do macOS, o agente do CloudWatch deverá enviar a métrica `mem_used_percent`. Em instâncias do Windows, o agente deverá enviar a métrica `Memory % Committed Bytes In Use`. Essas métricas serão incluídas se você usar o assistente para criar o arquivo de configuração do agente do CloudWatch e selecionar qualquer um dos conjuntos de métricas predefinidos. As métricas coletadas pelo agente do CloudWatch são cobradas como métricas personalizadas. Para obter mais informações, consulte Instalação do atendente do CloudWatch.

Ao usar o agente do CloudWatch para coletar essas métricas de memória a serem usadas com a visualização de integridade de recursos, você deverá incluir a seção a seguir no arquivo de configuração do agente do CloudWatch. Esta seção contém as configurações de dimensão padrão e é criada por padrão; Portanto, não altere nenhuma parte desta seção para nada diferente do que é demonstrado no exemplo a seguir.

```
"append_dimensions": {
  "ImageId": "${aws:ImageId}",
  "InstanceId": "${aws:InstanceId}",
  "InstanceType": "${aws:InstanceType}",
  "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
},
```

- Para visualizar todas as informações disponíveis no modo de integridade do recurso, é necessário estar conectado a uma conta que tenha as permissões a seguir. Caso sua sessão tenha menos permissões, você poderá continuar utilizando a visualização de integridade de recursos, mas alguns dados de performance não estarão visíveis.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "autoscaling:Describe*",
                "cloudwatch:Describe*",
                "cloudwatch:Describe*",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "logs:Get*",
                "logs:Describe*",
                "sns:Get*",
                "sns:List*",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeRegions"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
```

```
}
```

Para visualizar a integridade do recurso em sua conta

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Monitoramento da infraestrutura, Integridade de recursos.

   A página de integridade do recurso é exibida, mostrando um quadrado para cada host de sua conta. Cada quadrado recebe uma cor baseada no status atual desse host, de acordo com a configuração em Color by (Colorir por). Os quadrados de host com um símbolo de alarme atualmente têm um ou mais alarmes no estado ALARM.

   É possível ver até 500 hosts em uma única visualização. Caso tenha mais hosts em sua conta, use as configurações de filtro da etapa 6 deste procedimento.
3. Para alterar os critérios que serão usados para exibir a integridade de cada host, escolha uma configuração para Color by (Colorir por). Você pode escolher CPU Utilization (Utilização da CPU), Memory Utilization (Utilização da memória) ou Status check (Verificação de status). As métricas de utilização de memória estão disponíveis apenas para hosts que executam o agente do CloudWatch e o têm configurado para coletar métricas de memória e enviá-las ao namespace `CWAgent` padrão. Para obter mais informações, consulte Coletar métricas, logs e rastreamentos com o agente do CloudWatch.
4. Para alterar os limites e as cores que são usadas para os indicadores de integridade na grade, escolha o ícone de engrenagem acima da grade.
5. Para alternar se os alarmes serão exibidos na grade do host, escolha ou desmarque Show alarms across all metrics (Exibir alarmes em todas as métricas).
6. Para dividir os hosts no mapa em grupos, escolha um critério de agrupamento em Group by (Agrupar por).
7. Para restringir a visulização para menos hosts, escolha um critério de filtro para Filter by (Filtrar por). É possível filtrar por etiquetas e por agrupamentos de recursos, como grupo do Auto Scaling, tipo de instância, grupo de segurança etc.
8. Para classificar hosts, escolha um critério de classificação em Sort by (Classificar por). É possível classificar por resultados da verificação de status, estado da instância, utilização da CPU ou memória e o número de alarmes que estão no estado ALARM.

9. Para ver mais informações sobre um host, escolha o quadrado que representa esse host. É exibido um painel pop-up. Para então aprofundar as informações sobre esse host, escolha View dashboard (Visualizar painel) ou View on list (Visualizar na lista).

Observabilidade entre contas do CloudWatch

Com a observabilidade entre contas do Amazon CloudWatch, você pode monitorar e solucionar problemas de aplicações que abrangem várias contas em uma região. Pesquise, visualize e analise facilmente suas métricas, logs, rastreamentos e aplicações do Application Insights e Monitor de Internet em qualquer conta vinculada, sem limites entre as contas.

Configure uma ou mais contas da AWS como contas de monitoramento e vincule-as a várias contas de origem. Uma conta de monitoramento é uma conta central da AWS que pode visualizar e interagir com dados de observabilidade gerados das contas de origem. Uma conta de origem é uma conta individual da AWS que gera dados de observabilidade para os recursos que nela residem. As contas de origem compartilham os dados de observabilidade com a conta de monitoramento. Os dados de observabilidade compartilhados podem incluir os seguintes tipos de telemetria:

- Métricas no Amazon CloudWatch. É possível optar por compartilhar as métricas de todos os namespaces com a conta de monitoramento ou filtrar para um subconjunto de namespaces.
- Grupos de logs no Amazon CloudWatch Logs. É possível optar por compartilhar todos os grupos de logs com a conta de monitoramento ou filtrar para um subconjunto de grupos de logs.
- Rastreamentos no AWS X-Ray
- Aplicações no Amazon CloudWatch Application Insights
- Monitores no Monitor de Internet do CloudWatch

Para criar vínculos entre as contas de monitoramento e as contas de origem, você pode usar o console do CloudWatch. Ou então, use os comandos do Observability Access Manager na AWS CLI e na API. Para obter mais informações, consulte Observability Access Manager API Reference (Referência de API do Observability Access Manager).

Um coletor é um recurso que representa um ponto de conexão em uma conta de monitoramento. As contas de origem podem ser vinculadas ao coletor para compartilhar dados de observabilidade. Cada conta pode ter um coletor por região. Cada coletor é gerenciado pela conta de monitoramento em que está localizado. Um vínculo de observabilidade é um recurso que representa o vínculo estabelecido entre uma conta de origem e uma conta de monitoramento. Os vínculos são gerenciados pela conta de origem.

Para ver uma demonstração da configuração da observabilidade entre contas do CloudWatch, assista ao vídeo a seguir.

O próximo tópico explica como configurar a observabilidade entre contas do CloudWatch nas contas de monitoramento e nas contas de origem. Para obter informações sobre o painel entre contas e entre regiões do CloudWatch, consulte Console do CloudWatch entre contas e entre regiões.

Usar Organizations para contas de origem

Há duas opções para a vinculação das contas de origem à conta de monitoramento. Você pode usar uma ou ambas as opções.

- Use o AWS Organizations para vincular contas em uma organização ou unidade organizacional à conta de monitoramento.
- Conecte as contas individuais da AWS à conta de monitoramento.

Recomendamos que você use o Organizations para que novas contas da AWS criadas posteriormente na organização sejam automaticamente integradas à observabilidade entre contas como contas de origem.

Detalhes sobre como vincular as contas de monitoramento e as contas de origem

- Cada conta de monitoramento pode ser vinculada a até 100.000 contas de origem.
- Cada conta de origem pode compartilhar dados com até cinco contas de monitoramento.
- Você pode configurar uma única conta como conta de monitoramento e também como conta de origem. Se você fizer isso, essa conta só enviará seus próprios dados de observabilidade para a conta de monitoramento vinculada. Ela não retransmitirá os dados de suas contas de origem.
- Uma conta de monitoramento especifica quais tipos de telemetria podem ser compartilhados com ela. Uma conta de origem especifica quais tipos de telemetria ela deseja compartilhar.
  - Se houver mais tipos de telemetria selecionados na conta de monitoramento do que na conta de origem, as contas serão vinculadas. Somente os tipos de dados selecionados nas duas contas serão compartilhados.
  - Se houver mais tipos de telemetria selecionados na conta de monitoramento do que na conta de origem, a criação da vinculação falhará e nada será compartilhado.
  - Um nome de métrica não é exibido no console da conta de monitoramento até que essa métrica emita novos pontos de dados após a criação do vínculo.
- Para remover um vínculo entre as contas, faça isso na conta de origem.

- Para excluir um coletor em uma conta de monitoramento, primeiro é necessário remover todos os vínculos para esse coletor da conta de monitoramento.

Definição de preço

A observabilidade entre contas no CloudWatch não tem custo adicional para logs e métricas, e a primeira cópia de rastreamento é gratuita. Para obter mais informações sobre a definição de preço, consulte Preços do Amazon CloudWatch.

Sumário

Vincular contas de monitoramento a contas de origem

Os tópicos desta seção explicam como configurar vínculos entre as contas de monitoramento e as contas de origem.

Recomendamos que você crie uma nova conta da AWS para servir como conta de monitoramento para a sua organização.

Sumário

- Visão geral da configuração
- Etapa 1: configurar uma conta de monitoramento
- Etapa 2: (opcional) baixe um modelo do AWS CloudFormation ou uma URL
- Etapa 3: vincular as contas de origem
  - Use um modelo do AWS CloudFormation para configurar todas as contas de uma organização ou unidade organizacional como contas de origem
  - Use um modelo do AWS CloudFormation para configurar contas de origem individuais
  - Usar uma URL para configurar contas de origem individuais

Permissões requeridas

Para criar vínculos entre uma conta de monitoramento e uma conta de origem, você deve ter feito login com determinadas permissões.

- Para configurar uma conta de monitoramento, você deve ter total acesso de administrador à conta de monitoramento ou fazer login nessa conta com as seguintes permissões:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSinkModification",
            "Effect": "Allow",
            "Action": [
                "oam:CreateSink",
                "oam:DeleteSink",
                "oam:PutSinkPolicy",
                "oam:TagResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowReadOnly",
            "Effect": "Allow",
            "Action": ["oam:Get*", "oam:List*"],
            "Resource": "*"
        }
    ]
}
```

- Conta de origem, com o escopo de uma conta de monitoramento específica: para criar, atualizar e gerenciar os vínculos para apenas uma conta de monitoramento especificada, você deve entrar na conta com pelo menos as permissões a seguir. Neste exemplo, a conta de monitoramento é a 999999999999.

  Se o vínculo não compartilhar todos os cinco tipos de recursos (métricas, logs, rastreamentos, aplicações do Application Insights e monitoramentos do Monitor de Internet), será possível omitir `cloudwatch:Link`, `logs:Link`, `xray:Link`, `applicationinsights:Link` ou `internetmonitor:Link`, conforme necessário.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink",
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Effect": "Allow",
            "Resource": "arn:*:oam:*:*:link/*"
        },
        {
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Effect": "Allow",
            "Resource": "arn:*:oam:*:*:sink/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": [
                        "999999999999"
                    ]
                }
            }
        },
        {
            "Action": "oam:ListLinks",
            "Effect": "Allow",
```

```
            "Resource": "*"
        },
        {
            "Action": "cloudwatch:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "logs:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "xray:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
             "Action": "applicationinsights:Link",
             "Effect": "Allow",
             "Resource": "*"
         },
        {
             "Action": "internetmonitor:Link",
             "Effect": "Allow",
             "Resource": "*"
        }
    ]
}
```

- Conta de origem com permissões para realizar a vinculação com qualquer conta de monitoramento: para criar um vínculo para qualquer coletor de conta de monitoramento existente e compartilhar métricas, grupos de logs, rastreamentos, aplicações do Application Insights e monitoramentos do Monitor de Internet, você deverá iniciar uma sessão na conta de origem com permissões totais de administrador ou iniciar uma sessão com as permissões apresentadas a seguir.

  Se o vínculo não compartilhar todos os cinco tipos de recursos (métricas, logs, rastreamentos, aplicações do Application Insights e monitoramentos do Monitor de Internet), será possível omitir `cloudwatch:Link`, `logs:Link`, `xray:Link`, `applicationinsights:Link` ou `internetmonitor:Link`, conforme necessário.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:List*",
                "oam:Get*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Action": "cloudwatch:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "xray:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "logs:Link",
            "Effect": "Allow",
```

```
            "Resource": "*"
        },
        {
             "Action": "applicationinsights:Link",
             "Effect": "Allow",
             "Resource": "*"
        },
        {
             "Action": "internetmonitor:Link",
             "Effect": "Allow",
             "Resource": "*"
        }
    ]
}
```

Visão geral da configuração

As etapas de alto nível a seguir mostram como configurar a observabilidade entre contas do CloudWatch.

> **Note**
>
> Recomendamos a criação de uma nova conta da AWS para servir como conta de monitoramento para a sua organização.

1. Configure uma conta de monitoramento dedicada.
2. (Opcional) Baixe um modelo do AWS CloudFormation ou copie uma URL para vincular as contas de origem.
3. Vincule as contas de origem à conta de monitoramento.

Depois de concluir essas etapas, você poderá usar a conta de monitoramento para visualizar os dados de observabilidade das contas de origem.

Etapa 1: configurar uma conta de monitoramento

Siga as etapas desta seção para configurar uma conta da AWS como conta de monitoramento para a observabilidade entre contas do CloudWatch.

Pré-requisitos

- Se você estiver configurando contas em uma organização do AWS Organizations como contas de origem, obtenha o caminho da organização ou a ID da organização.
- Se você não estiver usando o Organizations para as contas de origem, obtenha as IDs das contas de origem.

Para configurar uma conta como conta de monitoramento, é necessário ter determinadas permissões. Para ter mais informações, consulte Permissões requeridas.

Para configurar uma conta de monitoramento

1. Faça login na conta que você deseja usar como uma conta de monitoramento.
2. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação à esquerda, escolha Configurações.
4. Em Monitoring account configuration (Configuração de conta de monitoramento), escolha Configure (Configurar).
5. Em Selecionar dados, escolha se esta conta de monitoramento poderá visualizar dados de Logs, Métricas, Rastreamentos, Aplicações do Application Insights e Monitoramentos do Monitor de Internet das contas de origem às quais está vinculada.
6. Em List source accounts (Listar contas de origem), insira as contas de origem que essa conta de monitoramento visualizará. Para identificar as contas de origem, insira IDs de conta individuais, caminhos de organizações ou os IDs de organizações. Se você inserir um caminho de organização ou um ID de organização, essa conta de monitoramento poderá visualizar dados de observabilidade de todas as contas vinculadas a essa organização.

   Separe com vírgulas as entradas dessa lista.

   > **Important**
   >
   > Ao inserir um caminho organizacional, siga o formato exato. O ou-id deve terminar com uma / (um caractere de barra). Por exemplo: `o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-def0-awsbbbb/`

7. Em Define a label to identify your source account (Definir um rótulo para identificar a conta de origem), especifique se você deseja usar nomes de conta ou endereços de e-mail para identificar as contas de origem quando usar a conta de monitoramento para visualizá-las.

8. Selecione Configurar.

> **Important**
>
> O vínculo entre as contas de monitoramento e de origem não estará completo até que você configure as contas de origem. Para obter mais informações, consulte as seções a seguir.

Etapa 2: (opcional) baixe um modelo do AWS CloudFormation ou uma URL

Para vincular contas de origem a uma conta de monitoramento, recomendamos usar um modelo do AWS CloudFormation ou uma URL.

- Se você estiver vinculando uma organização inteira, o CloudWatch fornece um modelo do AWS CloudFormation.
- Se você estiver vinculando contas individuais, use um modelo do AWS CloudFormation ou uma URL fornecida pelo CloudWatch.

Para usar um modelo do AWS CloudFormation, você deve baixá-lo durante essas etapas. Depois de vincular a conta de monitoramento a pelo menos uma conta de origem, o modelo do AWS CloudFormation não estará mais disponível para download.

Para baixar um modelo do AWS CloudFormation ou copiar uma URL para vincular contas de origem à conta de monitoramento

1. Faça login na conta que você deseja usar como uma conta de monitoramento.
2. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação à esquerda, escolha Configurações.
4. Em Monitoring account configuration (Configuração de conta de monitoramento), escolha Resources to link accounts (Recursos para vincular contas).
5. Execute um destes procedimentos:

   - Escolha a organização da AWS para obter um modelo a ser usado para vincular as contas de uma organização a essa conta de monitoramento.
   - Escolha Any account (Qualquer conta) para obter um modelo ou uma URL para configurar contas individuais como contas de origem.

6. Execute um destes procedimentos:
   - Se você escolheu Organização da AWS, escolha Baixar modelo do CloudFormation.
   - Se você escolheu Any account (Qualquer conta), escolha Download CloudFormation template (Baixar modelo do CloudFormation) ou Copy URL (Copiar URL).
7. (Opcional) Repita as etapas 5 e 6 para baixar ambos, o modelo do AWS CloudFormation e a URL.

Etapa 3: vincular as contas de origem

Use as etapas nestas seções para vincular contas de origem a uma conta de monitoramento.

Para vincular contas de monitoramento a contas de origem, é necessário ter determinadas permissões. Para ter mais informações, consulte Permissões requeridas.

Use um modelo do AWS CloudFormation para configurar todas as contas de uma organização ou unidade organizacional como contas de origem

Essas etapas pressupõem que você já tenha baixado o modelo AWS CloudFormation necessário executando as etapas em Etapa 2: (opcional) baixe um modelo do AWS CloudFormation ou uma URL.

Para usar um modelo do AWS CloudFormation para vincular as contas de uma organização ou unidade organizacional à conta de monitoramento

1. Faça login na conta de gerenciamento da organização.
2. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
3. Na barra de navegação esquerda, escolha StackSets.
4. Verifique se você fez login na região desejada e escolha Create StackSet (Criar StackSet).
5. Escolha Próximo.
6. Escolha Template is ready (O modelo está pronto) e escolha Upload a template file (Carregar um arquivo de modelo).
7. Escolha Choose file (Escolher arquivo), escolha o modelo que você baixou da conta de monitoramento e escolha Open (Abrir).
8. Escolha Próximo.

9. Na página Specify stack details (Especificar detalhes da pilha), digite o nome do StackSet e escolha Next (Avançar).
10. Em Add stacks to stack set (Adicionar pilhas ao conjunto de pilhas), escolha Deploy new stacks (Implantar novas pilhas).
11. Para Deployment targets (Destinos da implantação), escolha se deseja implantar em toda a organização ou nas unidades organizacionais especificadas.
12. Em Specify regions (Especificar regiões), escolha em quais regiões a observabilidade entre contas do CloudWatch será implantada.
13. Escolha Próximo.
14. Na página Review (Revisar), reveja as opções selecionadas e escolha Submit (Enviar).
15. Na guia Stack instances (Instâncias de pilha), atualize a tela até ver que as instâncias de pilha têm o status CREATE_COMPLETE.

Use um modelo do AWS CloudFormation para configurar contas de origem individuais

Essas etapas pressupõem que você já tenha baixado o modelo AWS CloudFormation necessário executando as etapas em Etapa 2: (opcional) baixe um modelo do AWS CloudFormation ou uma URL.

Para usar um modelo do AWS CloudFormation para configurar contas de origem individuais para observabilidade entre contas do CloudWatch

1. Faça login na conta de origem.
2. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
3. No painel de navegação esquerda, escolha Stacks (Pilhas).
4. Verifique se fez login na região desejada e escolha Create stack (Criar pilha), With new resources (standard) (Com novos recursos (padrão).
5. Escolha Próximo.
6. Selecione Carregar um arquivo de modelo.
7. Escolha Choose file (Escolher arquivo), escolha o modelo que você baixou da conta de monitoramento e escolha Open (Abrir).
8. Escolha Próximo.
9. Na página Specify stack details (Especificar detalhes da pilha), insira um nome para a pilha, e escolha Next (Avançar).

10. Na página Configurar opções de pilha, selecione Avançar.
11. Na página Review (Revisar), escolha Submit (Enviar).
12. Na página de status da pilha, atualize a tela até ver que a pilha tem o status CREATE_COMPLETE.
13. Para usar esse mesmo modelo para vincular mais contas de origem a essa conta de monitoramento, saia dessa conta e entre na próxima conta de origem. Depois, repita as etapas de 2 a 12.

Usar uma URL para configurar contas de origem individuais

Essas etapas pressupõem que você já tenha copiado a URL necessária executando as etapas em Etapa 2: (opcional) baixe um modelo do AWS CloudFormation ou uma URL.

Para usar uma URL para vincular contas de origem individuais à conta de monitoramento

1. Faça login na conta que você deseja usar como uma conta de origem.
2. Insira a URL que você copiou da conta de monitoramento.

   Você verá a página de configurações do CloudWatch, com algumas informações preenchidas.
3. Em Selecionar dados, escolha se esta conta de origem compartilhará dados de Logs, Métricas, Rastreamentos, Aplicações do Application Insights e Monitoramentos do Monitor de Internet com esta conta de monitoramento.

   Tanto para Logs quanto para Métricas, é possível escolher se deseja compartilhar todos os recursos ou um subconjunto com a conta de monitoramento.

   a. (Opcional) Para compartilhar um subconjunto dos grupos de logs desta conta com a conta de monitoramento, selecione Logs e escolha Filtrar logs. Em seguida, use a caixa Filtrar logs para definir a estrutura de uma consulta para localizar os grupos de logs que você deseja compartilhar. A consulta usará o termo `LogGroupName` e um ou mais dos operadores apresentados a seguir.

      - `= e !=`
      - `AND`
      - `OR`

- ^ indica LIKE e !^ indica NOT LIKE. Esses operadores podem ser usados somente como pesquisas de prefixo. Inclua um símbolo de % no final da string que você deseja pesquisar e incluir.
- IN e NOT IN, usando parênteses (( ))

A consulta completa não deve ter mais de 2 mil caracteres e está limitada a cinco operadores condicionais. Os operadores condicionais são AND e OR. Não há limite para o número de outros operadores.

> **Tip**
>
> Escolha Visualizar amostras de consultas para obter a sintaxe correta para os formatos de consulta comuns.

b. (Opcional) Para compartilhar um subconjunto de namespaces de métricas desta conta com a conta de monitoramento, selecione Métricas e escolha Filtrar métricas. Em seguida, use a caixa Filtrar métricas para definir a estrutura de uma consulta para localizar os namespaces de métricas que você deseja compartilhar. Use o termo Namespace e um ou mais dos operadores apresentados a seguir.

- = e !=
- AND
- OR
- LIKE e NOT LIKE. Esses operadores podem ser usados somente como pesquisas de prefixo. Inclua um símbolo de % no final da string que você deseja pesquisar e incluir.
- IN e NOT IN, usando parênteses (( ))

A consulta completa não deve ter mais de 2 mil caracteres e está limitada a cinco operadores condicionais. Os operadores condicionais são AND e OR. Não há limite para o número de outros operadores.

> **Tip**
>
> Escolha Visualizar amostras de consultas para obter a sintaxe correta para os formatos de consulta comuns.

4. Não altere o ARN em Enter monitoring account configuration ARN (Inserir ARN de configuração da conta de monitoramento).
5. A seção Define a label to identify your source account (Definir um rótulo para identificar sua conta de origem) é pré-preenchida com a opção de label da conta de monitoramento. Opcionalmente, escolha Edit (Editar) para alterar a URL.
6. Escolha Link (Vincular).
7. Insira **`Confirm`** na caixa e escolha Confirm (Confirmar).
8. Para usar essa mesma URL para vincular mais contas de origem a essa conta de monitoramento, saia dessa conta e entre na próxima conta de origem. Depois, repita as etapas de 2 a 7.

Gerenciar contas de monitoramento e contas de origem

Depois de configurar as contas de monitoramento e contas de origem, você poderá usar as etapas nestas seções para gerenciá-las.

Sumário

Vincular mais contas de origem a uma conta de monitoramento existente

Siga as etapas desta seção para adicionar vínculos de mais contas de origem a uma conta de monitoramento existente.

Cada conta de origem pode ser vinculada a até cinco contas de monitoramento. Cada conta de monitoramento pode ser vinculada a até 100.000 contas de origem.

Para gerenciar uma conta de origem, é necessário ter determinadas permissões. Para ter mais informações, consulte Permissões requeridas.

Para adicionar mais contas de origem a uma conta de monitoramento

1. Faça login na conta de monitoramento.
2. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação à esquerda, escolha Configurações.
4. Em Monitoring account configuration (Configuração de conta de monitoramento), escolha Manage source accounts (Gerenciar contas de origem).
5. Escolha a guia Configuration policy (Política de configuração).
6. Na caixa Configuration policy (Política de configuração), adicione a ID da nova conta de origem na linha Principal (Entidade principal).

   Por exemplo, suponha que a linha Principal (Entidade principal) seja atualmente a seguinte:

   ```
   "Principal": {"AWS": ["111111111111", "222222222222"]}
   ```

   Para adicionar 999999999999 como uma terceira conta de origem, edite a linha da seguinte forma:

   ```
   "Principal": {"AWS": ["111111111111", "222222222222", "999999999999"]}
   ```
7. Escolha Atualizar.
8. Escolha a guia Configuration details (Detalhes da configuração).
9. Escolha o ícone de cópia ao lado do ARN do coletor da conta de monitoramento.
10. Faça login na conta que você deseja usar como uma nova conta de origem.
11. Cole o ARN do coletor da conta de monitoramento que você copiou na etapa 9.

    Você verá a página de configurações do CloudWatch, com algumas informações preenchidas.
12. Em Selecionar dados, escolha se esta conta de origem compartilhará dados de Logs, Métricas, Rastreamentos e Aplicações - Application Insights com a conta de monitoramento com a qual ela está vinculada.
13. Não altere o ARN em Enter monitoring account configuration ARN (Inserir ARN de configuração da conta de monitoramento).

14. A seção Define a label to identify your source account (Definir um rótulo para identificar sua conta de origem) é pré-preenchida com a opção de label da conta de monitoramento. Opcionalmente, escolha Edit (Editar) para alterar a URL.
15. Escolha Link (Vincular).
16. Insira **`Confirm`** na caixa e escolha Confirm (Confirmar).

Remover o vínculo entre uma conta de monitoramento e uma conta de origem

Siga as etapas desta seção para interromper o envio de dados de uma conta de origem para uma conta de monitoramento.

Você deve ter as permissões requeridas para gerenciar uma conta de origem para realizar essa tarefa. Para ter mais informações, consulte Permissões requeridas.

Para remover o vínculo entre uma conta de origem e uma conta de monitoramento

1. Faça login na conta de origem.
2. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação à esquerda, escolha Configurações.
4. Em Source account information (Informações da conta de origem), escolha View monitoring accounts (Visualizar contas de monitoramento).
5. Marque a caixa de seleção ao lado da conta de monitoramento com a qual você deseja parar de compartilhar dados.
6. Escolha Stop sharing data (Parar de compartilhar dados), Confirm (Confirmar).
7. Faça login na conta de monitoramento.
8. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
9. Escolha Configurações.
10. Em Monitoring account information (Informações da conta de monitoramento), escolha View configuration (Exibir configuração).
11. Na caixa Policy (Política), exclua o ID da conta de origem da linha Principal (Entidade principal) e escolha Update (Atualizar).

Visualizar informações sobre uma conta de monitoramento

Siga as etapas desta seção para visualizar as configurações entre contas de uma conta de monitoramento.

Para gerenciar uma conta de monitoramento, é necessário ter determinadas permissões. Para ter mais informações, consulte Permissões requeridas.

Para gerenciar uma conta de monitoramento

1. Faça login na conta de monitoramento.
2. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação à esquerda, escolha Configurações.
4. Em Monitoring account configuration (Configuração de conta de monitoramento), escolha Manage source accounts (Gerenciar contas de origem).
5. Para visualizar a política do Observability Access Manager que permite que essa conta seja uma conta de monitoramento, escolha a guia Configuration policy (Política de configuração).
6. Para visualizar as contas de origem vinculadas a essa conta de monitoramento, escolha a guia Linked source accounts (Contas de origem vinculadas).
7. Para visualizar o ARN do coletor da conta de monitoramento e os tipos de dados que essa conta de monitoramento pode visualizar nas contas de origem vinculadas, escolha a guia Linked source accounts (Contas de origem vinculadas).

Métricas de consulta de outras fontes de dados

Você pode usar o CloudWatch para consultar, visualizar e criar alarmes para métricas de outras fontes de dados. Para fazê-lo, conecte o CloudWatch a outras fontes de dados. Isso proporciona uma experiência de monitoramento única e consolidada no console do CloudWatch. Você pode ter uma visão unificada das métricas da sua infraestrutura e das aplicações, onde quer que os dados estejam armazenados, ajudando você a identificar e resolver problemas com mais rapidez.

Depois que você se conecta a uma fonte de dados usando um assistente do CloudWatch, o CloudWatch cria uma pilha do AWS CloudFormation que implanta e configura uma função do AWS Lambda. Essa função do Lambda é executada sob demanda toda vez que você consulta a fonte de dados. O construtor de consultas do CloudWatch mostra em tempo real uma lista de elementos que podem ser consultados, como métricas, tabelas, campos ou rótulos. Conforme você faz escolhas, o criador de consultas preenche previamente uma consulta no idioma nativo da fonte selecionada.

O CloudWatch fornece assistentes guiados para você se conectar às fontes de dados a seguir. Para essas fontes de dados, forneça informações básicas para identificar a fonte de dados e as credenciais. Você também pode criar manualmente conectores para outras fontes de dados ao criar suas próprias funções do Lambda.

- Amazon OpenSearch Service: obtenha métricas dos logs e rastreamentos do OpenSearch Service.
- Amazon Managed Service for Prometheus: consulte essas métricas usando o PromQL.
- Amazon RDS para MySQL: use o SQL para converter dados armazenados em tabelas do Amazon RDS em métricas.
- Amazon RDS para PostgreSQL: use o SQL para converter dados armazenados em tabelas do Amazon RDS em métricas.
- Arquivos CSV do Amazon S3: exiba dados de métricas de um arquivo CSV armazenado em um bucket do Amazon S3.
- Microsoft Azure Monitor: consulte métricas da sua conta do Microsoft Azure Monitor.
- Prometheus: consulte essas métricas usando o PromQL.

Depois de criar conectores para fontes de dados, consulte Criar um gráfico de métricas com base em outra fonte de dados para obter informações sobre como representar graficamente uma métrica baseada em uma fonte de dados. Para obter informações sobre como configurar um alarme em uma

métrica de uma fonte de dados, consulte Criação de um alarme com base em uma fonte de dados conectada.

Tópicos

Gerenciar o acesso a fontes de dados

O CloudWatch usa o AWS CloudFormation para criar os recursos necessários na sua conta. Recomendamos que você use a condição `cloudformation:TemplateUrl` para controlar o acesso aos modelos do AWS CloudFormation ao conceder permissões `CreateStack` aos usuários do IAM.

> **Warning**
>
> Qualquer usuário a quem você concede permissão para invocar a fonte de dados pode consultar métricas dessa fonte de dados, mesmo que esse usuário não tenha permissões diretas do IAM para a fonte de dados. Por exemplo, se você conceder permissões `lambda:InvokeFunction` em uma função do Lambda da fonte de dados do Amazon Managed Service for Prometheus a um usuário, esse usuário poderá consultar métricas do espaço de trabalho correspondente do Amazon Managed Service for Prometheus, mesmo que você não tenha concedido a ele acesso direto do IAM a esse espaço de trabalho.

Você pode encontrar URLs de modelos para fontes de dados na página Criar pilha no console de configurações do CloudWatch.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect" : "Allow",
      "Action" : [ "cloudformation:CreateStack" ],
```

```
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "cloudformation:TemplateUrl" : [ data-source-template-url ]
        }
      }
    }
  ]
}
```

Para obter mais informações sobre o controle de acesso ao AWS CloudFormation, consulte Controlar o acesso com o AWS Identity and Access Management

Conectar-se a uma fonte de dados pré-criada com um assistente

Este tópico fornece instruções para usar o assistente para conectar o CloudWatch às fontes de dados a seguir.

- Amazon OpenSearch Service
- Amazon Managed Service para Prometheus
- Amazon RDS para MySQL
- Amazon RDS para PostgreSQL
- Arquivos CSV do Amazon S3
- Microsoft Azure Monitor
- Prometheus

Posteriormente, nesta seção, serão mostradas subseções com observações sobre gerenciamento e consulta com cada uma dessas fontes de dados.

Como criar um conector de fonte de dados

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Configurações.
3. Escolha a guia Fontes de dados de métricas.
4. Escolha Criar fonte de dados.
5. Selecione a fonte que você deseja e escolha Próximo.

6. Insira um nome para a fonte de dados.
7. Insira as outras informações necessárias, dependendo da fonte de dados que você escolheu. Isso pode incluir credenciais para acessar a fonte de dados e informações de identificação da fonte de dados, como nome do espaço de trabalho do Prometheus, nome do banco de dados ou nome do bucket do Amazon S3. Para serviços da AWS, o assistente descobre os recursos e os preenche na lista suspensa de seleção.

   Para obter mais observações sobre a fonte de dados que você está usando, consulte as seções depois desse procedimento.
8. Para que o CloudWatch se conecte à fonte de dados em uma VPC, escolha Usar uma VPC e selecione a VPC a ser usada. Em seguida, selecione a sub-rede e o grupo de segurança.
9. Escolha Confirmo que o AWS CloudFormation pode criar recursos do IAM. Esse recurso é o perfil de execução da função do Lambda.
10. Escolha Criar fonte de dados.

    A nova fonte que você acabou de adicionar não aparecerá até que a pilha AWS CloudFormation termine de criá-la. Para verificar o progresso, escolha Visualizar o status da minha pilha do CloudFormation. Como alternativa, você pode escolher o ícone de atualização para atualizar essa lista.

    Quando a nova fonte de dados for exibida na lista, ela estará pronta para ser usada. Você pode escolher Consulta das métricas do CloudWatch para começar a fazer consultas com ele. Para ter mais informações, consulte Criar um gráfico de métricas com base em outra fonte de dados.

Amazon Managed Service para Prometheus

Atualizar a configuração da fonte de dados

- Você pode atualizar a fonte de dados manualmente da seguinte maneira:
  - Para atualizar o ID do espaço de trabalho do Amazon Managed Service for Prometheus, atualize a variável de ambiente `AMAZON_PROMETHEUS_WORKSPACE_ID` para a função do Lambda do conector de fonte de dados.
  - Para atualizar a configuração da VPC, consulte Como configurar o acesso à VPC (console) para obter mais informações.

Consultar a fonte de dados

- Ao consultar o Amazon Managed Service for Prometheus, depois de selecionar a fonte de dados na guia Consulta a várias fontes e selecionar um conector do Amazon Managed Service for Prometheus, você pode usar o Auxiliar de consulta para descobrir métricas e rótulos e fornecer consultas do ProMQL simples. Você também pode usar o editor de consultas PromQL para criar uma consulta PromQL.
- As consultas de várias linhas não são compatíveis com os conectores de fonte de dados do CloudWatch. Cada feed de linha é substituído por um espaço quando a consulta é executada ou quando você cria um alarme ou um widget de painel com a consulta. Em alguns casos, isso pode tornar a consulta inválida. Por exemplo, se a consulta contiver um comentário de uma só linha, ela não será válida. Se você tentar criar um painel ou um alarme com uma consulta de várias linhas na linha de comando ou na Infraestrutura como Código, a API rejeitará a ação com um erro de análise.

Amazon OpenSearch Service

Criar a fonte de dados

Se o domínio do OpenSearch estiver habilitado para o FGAC, você deverá realizar o mapeamento do perfil de execução da função do Lambda do conector para um usuário no OpenSearch Service. Para obter mais informações, consulte a seção Mapping users to roles em Managing permissions na documentação do OpenSearch Service.

Se o domínio do OpenSearch puder ser acessado somente em uma nuvem privada virtual (VPC), será necessário incluir uma nova variável de ambiente, de forma manual, na função do Lambda chamada `AMAZON_OPENSEARCH_ENDPOINT`. O valor para esta variável deverá ser o domínio raiz do endpoint do OpenSearch. É possível obter esse domínio raiz ao remover `https://` e `<region>.es.amazonaws.com` do endpoint de domínio listado no console do OpenSearch Service. Por exemplo, se o endpoint do domínio fosse `https://sample-domain.us-east-1.es.amazonaws.com`, o domínio raiz seria `sample-domain`.

Atualizar a fonte de dados

- Você pode atualizar a fonte de dados manualmente da seguinte maneira:
  - Para atualizar o domínio do OpenSearch Service, atualize a variável de ambiente `AMAZON_OPENSEARCH_DOMAIN_NAME` para a função do Lambda do conector da fonte de dados.

- Para atualizar a configuração da VPC, consulte Como configurar o acesso à VPC (console) para obter mais informações.

Consultar a fonte de dados

- Ao consultar o OpenSearch Service, depois de selecionar a fonte de dados na guia Consulta de várias fontes, faça o seguinte:
  - Selecione o Índice a ser consultado.
  - Selecione o nome da Métrica (qualquer campo numérico no documento) e Estatística.
  - Selecione o eixo do Tempo (qualquer campo de data no documento).
  - Selecione Filtros a serem aplicados (qualquer campo de String no documento).
  - Escolha Representar consulta graficamente.

Amazon RDS para PostgreSQL e Amazon RDS para MySQL

Criar a fonte de dados

- Se sua fonte de dados só estiver acessível em uma VPC, você deverá incluir a configuração da VPC para o conector, conforme descrito em Conectar-se a uma fonte de dados pré-criada com um assistente. Se a fonte de dados se conectar à VPC para obter credenciais, o endpoint deverá ser configurado na VPC. Para obter mais informações, consulte Usar um endpoint da VPC no AWS Secrets Manager.

  Além disso, você deve criar um endpoint da VPC para o serviço do Amazon RDS. Para obter mais informações, consulte API do Amazon RDS e endpoints da VPC de interface (AWS PrivateLink).

Atualizar a fonte de dados

- Você pode atualizar a fonte de dados manualmente da seguinte maneira:
  - Para atualizar a instância do banco de dados, atualize a variável de ambiente `RDS_INSTANCE` para a função do Lambda do conector da fonte de dados.
  - Para atualizar o nome de usuário e a senha usados para conectar-se ao Amazon RDS, use AWS Secrets Manager. Você pode encontrar o ARN do segredo usado para a fonte de dados na variável de ambiente `RDS_SECRET` na função do Lambda da fonte de dados. Para obter mais

informações sobre como atualizar o segredo no AWS Secrets Manager, consulte Modificação de um segredo do AWS Secrets Manager.

- Para atualizar a configuração da VPC, consulte Como configurar o acesso à VPC (console) para obter mais informações.

Consultar a fonte de dados

- Ao consultar o Amazon RDS, depois de selecionar a fonte de dados na guia Consulta de várias fontes e selecionar um conector do Amazon RDS, você pode usar o detector de banco de dados para visualizar bancos de dados, tabelas e colunas disponíveis. Você também pode usar o editor SQL para criar uma consulta SQL.

  Você pode usar as seguintes variáveis na consulta:

  - `$start.iso`: a hora de início em formato de data ISO
  - `$end.iso`: a hora de término em formato de data ISO
  - `$period`: o período selecionado em segundos

  Por exemplo, você pode consultar `SELECT value, timestamp FROM table WHERE timestamp BETWEEN $start.iso and $end.iso`
- As consultas de várias linhas não são compatíveis com os conectores de fonte de dados do CloudWatch. Cada feed de linha é substituído por um espaço quando a consulta é executada ou quando você cria um alarme ou um widget de painel com a consulta. Em alguns casos, isso pode tornar a consulta inválida. Por exemplo, se a consulta contiver um comentário de uma só linha, ela não será válida. Se você tentar criar um painel ou um alarme com uma consulta de várias linhas na linha de comando ou na Infraestrutura como Código, a API rejeitará a ação com um erro de análise.

> **Note**
>
> Se nenhum campo de data for encontrado nos resultados, os valores de cada campo numérico serão somados a valores únicos e representados graficamente no intervalo de tempo fornecido. Se os carimbos de data/hora não estiverem alinhados com o período selecionado no CloudWatch, os dados serão automaticamente agregados usando `SUM` e alinhados com o período no CloudWatch.

Arquivos CSV do Amazon S3

Consultar a fonte de dados

- Ao consultar arquivos CSV do Amazon S3, depois de selecionar a fonte de dados na guia Consulta de várias fontes e selecionar um conector do Amazon S3, selecione o bucket e a chave do Amazon S3.

  O arquivo CSV deve estar nos seguintes formatos:

  - O carimbo de data e hora deve ser a primeira coluna.
  - A tabela deve ter uma linha de cabeçalho. Os cabeçalhos são usados para dar nome às métricas. O título da coluna referente ao carimbo de data e hora será ignorado; somente os títulos das colunas de métricas são usados.
  - Os carimbos de data e hora devem estar no formato de data ISO.
  - As métricas devem ser campos numéricos.

```
Timestamp, Metric-1, Metric-2, ...
```

Veja um exemplo a seguir:

timestamp	CPU (%)	Memory (%) (Memória (%))	Armazenamento (%)
2023-11-23T17:09:41+00:00	1	2	3
2023-11-23T17:04:41+00:00	4	5	6
2023-11-23T16:59:41+00:00	7	8	9
2023-11-23T16:54:41+00:00	10	11	12

> **Note**
>
> Se nenhum carimbo de data/hora for fornecido, os valores de cada métrica serão somados a valores únicos e representados graficamente no intervalo de tempo fornecido. Se os carimbos de data/hora não estiverem alinhados com o período selecionado no CloudWatch, os dados serão automaticamente agregados usando `SUM` e alinhados com o período no CloudWatch.

Microsoft Azure Monitor

Criar a fonte de dados

- Você deve fornecer seu ID de locatário, o ID do cliente e o segredo do cliente para se conectar ao Microsoft Azure Monitor. As credenciais serão armazenadas no AWS Secrets Manager. Para obter mais informações, consulte Criar um aplicativo do Microsoft Entra e uma entidade de serviço que possa acessar recursos na documentação da Microsoft.

Atualizar a fonte de dados

- Você pode atualizar a fonte de dados manualmente da seguinte maneira:
  - Para atualizar o ID do locatário, o ID do cliente e o segredo do cliente usados para se conectar ao Azure Monitor, você pode encontrar o ARN do segredo usado para a fonte de dados como a variável de ambiente `AZURE_CLIENT_SECRET` na função do Lambda da fonte de dados. Para obter mais informações sobre como atualizar o segredo no AWS Secrets Manager, consulte Modificação de um segredo do AWS Secrets Manager.

Consultar a fonte de dados

- Ao consultar o Azure Monitor, depois de selecionar a fonte de dados na guia Consulta de várias fontes e selecionar um conector do Azure Monitor, especifique a assinatura do Azure, o grupo de recursos e o recurso. Em seguida, você pode selecionar o namespace, a métrica e a agregação da métrica e filtrar por dimensões.

Prometheus

Criar a fonte de dados

- Você deve fornecer o endpoint do Prometheus e o usuário e a senha necessários para consultar o Prometheus. As credenciais serão armazenadas no AWS Secrets Manager.
- Se sua fonte de dados só estiver acessível em uma VPC, você deverá incluir a configuração da VPC para o conector, conforme descrito em Conectar-se a uma fonte de dados pré-criada com um assistente. Se a fonte de dados se conectar para obter credenciais, o endpoint deverá ser configurado na VPC. Para obter mais informações, consulte Usar um endpoint da VPC no AWS Secrets Manager.

Atualizar a configuração da fonte de dados

- Você pode atualizar a fonte de dados manualmente da seguinte maneira:
  - Para atualizar o endpoint do Prometheus, especifique o novo endpoint como a variável de ambiente `PROMETHEUS_API_ENDPOINT` na função do Lambda da fonte de dados.
  - Para atualizar o nome de usuário e a senha usados para se conectar ao Prometheus, você pode encontrar o ARN do segredo usado para a fonte de dados como a variável de ambiente `PROMETHEUS_API_SECRET` na função do Lambda da fonte de dados. Para obter mais informações sobre como atualizar o segredo no AWS Secrets Manager, consulte Modificação de um segredo do AWS Secrets Manager.
  - Para atualizar a configuração da VPC, consulte Como configurar o acesso à VPC (console) para obter mais informações.

Consultar a fonte de dados

> **Important**
>
> Os tipos de métricas do Prometheus são diferentes das métricas do CloudWatch e muitas métricas disponibilizadas por meio do Prometheus são cumulativas por projeto. Quando você consulta métricas do Prometheus, o CloudWatch não aplica qualquer transformação adicional aos dados: se você especificar somente o nome ou o rótulo da métrica, o valor exibido será cumulativo. Para obter mais informações, consulte Tipos de métrica na documentação do Prometheus.
>
> Para ver os dados das métricas do Prometheus como valores discretos, como as métricas do CloudWatch, você precisa editar a consulta antes de executá-la. Por exemplo, talvez seja necessário adicionar uma chamada à função de taxa em vez do nome da métrica

do Prometheus. Para obter a documentação sobre a função de taxa e outras funções do Prometheus, consulte rate() na documentação do Prometheus.

As consultas de várias linhas não são compatíveis com os conectores de fonte de dados do CloudWatch. Cada feed de linha é substituído por um espaço quando a consulta é executada ou quando você cria um alarme ou um widget de painel com a consulta. Em alguns casos, isso pode tornar a consulta inválida. Por exemplo, se a consulta contiver um comentário de uma só linha, ela não será válida. Se você tentar criar um painel ou um alarme com uma consulta de várias linhas na linha de comando ou na Infraestrutura como Código, a API rejeitará a ação com um erro de análise.

Notificação de atualizações disponíveis

Ocasionalmente, a Amazon poderá notificar você para recomendar a atualização dos conectores com uma versão mais recente disponível e fornecerá instruções sobre como fazê-lo.

Criar um conector personalizado para uma fonte de dados

Para conectar uma fonte de dados personalizada ao CloudWatch, você tem duas opções:

- Comece usando uma amostra de modelo fornecida pelo CloudWatch. Você pode usar JavaScript ou Python com esse modelo. Esses modelos incluem exemplos de código do Lambda que serão úteis para você criar a função do Lambda. Em seguida, você pode modificar a função do Lambda com base no modelo para se conectar à sua fonte de dados personalizada.
- Crie do zero uma função do AWS Lambda que implemente o conector da fonte de dados, a consulta de dados e a preparação da série temporal para uso pelo CloudWatch. Essa função deve pré-agregar ou mesclar pontos de dados, se necessário, e também alinhar o período e os carimbos de data/hora para serem compatíveis com o CloudWatch.

Sumário

    - (Opcional) Usar o AWS Secrets Manager para armazenar credenciais
    - (Opcional) Conecte-se a uma fonte de dados em uma VPC
- Etapa 2: criar uma política de permissões do Lambda
- Etapa 3: anexar uma tag de recurso à função do Lambda

Usar um modelo

O uso de um modelo cria um exemplo da função do Lambda e pode ajudar você a criar o conector personalizado com mais rapidez. Esses exemplos de funções fornecem exemplos de código para muitos cenários comuns envolvidos na criação de um conector personalizado. Você pode examinar o código do Lambda depois de criar um conector com um modelo e modificá-lo para usá-lo para se conectar à fonte de dados.

Além disso, se você usar o modelo, o CloudWatch se encarregará de criar a política de permissões do Lambda e anexará tags de recursos à função do Lambda.

Como usar o modelo para criar um conector para uma fonte de dados personalizada

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Configurações.
3. Escolha a guia Fontes de dados de métricas.
4. Escolha Criar fonte de dados.
5. Escolha o botão de opção em Personalizado: conceitos básicos do modelo e, em seguida, escolha Próximo.
6. Insira um nome para a fonte de dados.
7. Selecione um dos modelos listados.
8. Selecione Node.js ou Python.
9. Escolha Criar fonte de dados.

   A nova fonte personalizada que você acabou de adicionar não será exibida até que a pilha do AWS CloudFormation termine de criá-la. Para verificar o progresso, escolha Visualizar o status da minha pilha do CloudFormation. Como alternativa, você pode escolher o ícone de atualização para atualizar essa lista.

   Quando sua nova fonte de dados for exibida nessa lista, ela estará pronta para ser testada no console e modificada.

10. (Opcional) Para consultar os dados de teste dessa fonte no console, siga as instruções em Criar um gráfico de métricas com base em outra fonte de dados.
11. Modifique a função do Lambda de acordo com suas necessidades.

    a. No painel de navegação, selecione Configurações.

    b. Escolha a guia Fontes de dados de métricas.

    c. Escolha Visualizar no console do Lambda para a fonte que você deseja modificar.

    Agora, é possível modificar a função para acessar a fonte de dados. Para ter mais informações, consulte Etapa 1: criar a função.

    > **Note**
    >
    > Se usar o modelo quando escrever a função do Lambda, você não precisará seguir as instruções em Etapa 2: criar uma política de permissões do Lambda ou Etapa 3: anexar uma tag de recurso à função do Lambda. Essas etapas foram executadas pelo CloudWatch porque você usou o modelo.

Criar uma fonte de dados personalizada do zero

Siga as etapas nesta seção para criar uma função do Lambda que conecte o CloudWatch a uma fonte de dados.

Etapa 1: criar a função

O conector de uma fonte de dados personalizada deve ser compatível com eventos `GetMetricData` do CloudWatch. Opcionalmente, você também pode implementar um evento `DescribeGetMetricData` para fornecer documentação aos usuários no console do CloudWatch sobre como usar o conector. A resposta `DescribeGetMetricData` também pode ser usada para definir os padrões que serão usados no construtor de consultas personalizadas do CloudWatch.

O CloudWatch fornece trechos de código como exemplos para ajudar você a começar. Para obter mais informações, consulte o repositório de exemplos em https://github.com/aws-samples/cloudwatch-data-source-samples.

Restrições

- A resposta do Lambda deve ser menor que 6 Mb. Se a resposta exceder 6 Mb, a resposta `GetMetricData` marcará a função do Lambda como `InternalError` e nenhum dado será retornado.
- A função do Lambda deve concluir a execução em até dez segundos para fins de visualização e criação de painéis, ou em até 4,5 segundos para uso de alarmes. Se o tempo de execução exceder esse tempo, a resposta `GetMetricData` marcará a função do Lambda como `InternalError` e nenhum dado será retornado.
- A função do Lambda deve enviar a saída usando carimbos de data/hora de período em segundos.
- Se a função do Lambda não amostrar os dados novamente e, em vez disso, retornar dados que não correspondam à hora de início e à duração do período solicitados pelo usuário do CloudWatch, esses dados serão ignorados pelo CloudWatch. Os dados adicionais são descartados de qualquer visualização ou alarme. Qualquer dado que não esteja entre a hora de início e a hora de término também é descartado.

  Por exemplo, se o usuário solicitar dados das 10:00 às 11:00 com um período de cinco minutos, “10:00:00 a 10:04:59” e “10:05:00 a 10:09:59” serão os intervalos de tempo válidos para que os dados sejam retornados. Você deve retornar uma série temporal que inclua `10:00 value1`, `10:05 value2` e assim por diante. Se a função retornar `10:03 valueX`, por exemplo, ela será descartada porque 10:03 não corresponde à hora de início e ao período solicitados.
- As consultas de várias linhas não são compatíveis com os conectores de fonte de dados do CloudWatch. Cada feed de linha é substituído por um espaço quando a consulta é executada ou quando você cria um alarme ou um widget de painel com a consulta. Em alguns casos, isso pode tornar a consulta inválida.

Evento GetMetricData

Carga da solicitação

Veja a seguir um exemplo de uma carga da solicitação `GetMetricData` enviada como entrada para a função do Lambda.

```
{
  "EventType": "GetMetricData",
  "GetMetricDataRequest": {
    "StartTime": 1697060700,
    "EndTime": 1697061600,
    "Period": 300,
    "Arguments": ["serviceregistry_external_http_requests{host_cluster!=\"prod\"}"]
```

```
  }
}
```

- StartTime: o carimbo de data/hora que especifica os primeiros dados a serem retornados. O Tipo é o período do carimbo de data/hora em segundos.
- EndTime: o carimbo de data/hora que especifica os últimos dados a serem retornados. O Tipo é o período do carimbo de data/hora em segundos.
- Período: o número de segundos que cada agregação dos dados de métricas representa. O mínimo é 60 segundos. O Tipo é Segundos.
- Argumentos: matriz de argumentos a serem passados para a expressão matemática de métricas do Lambda. Para obter mais informações sobre como passar argumentos, consulte Como passar argumentos para sua função do Lambda.

Carga da resposta

Veja a seguir um exemplo de carga da resposta `GetMetricData` retornada pela função do Lambda.

```
{
   "MetricDataResults": [
      {
         "StatusCode": "Complete",
         "Label": "CPUUtilization",
         "Timestamps": [ 1697060700, 1697061000, 1697061300 ],
         "Values": [ 15000, 14000, 16000 ]
      }
   ]
}
```

A carga da resposta conterá um campo `MetricDataResults` ou um campo `Error`, mas não ambos.

Um campo `MetricDataResults` é uma lista de campos de séries temporais do tipo `MetricDataResult`. Cada um desses campos de séries temporais pode incluir os campos a seguir.

- StatusCode: (opcional) `Complete` indica que todos os pontos de dados no intervalo de tempo solicitado foram retornados. `PartialData` significa que um conjunto incompleto de pontos de dados foi retornado. Se isso for omitido, o padrão será `Complete`.

Valores válidos: `Complete` | `InternalError` | `PartialData` | `Forbidden`

- Mensagens: lista opcional de mensagens com informações adicionais sobre os dados retornados.

  Tipo: matriz de objetos MessageData com strings `Code` e `Value`.

- Rótulo: o rótulo legível por humanos associado aos dados.

  Tipo: sequência

- Carimbos de data/hora: os carimbos de data/hora dos pontos de dados, formatados em períodos. O número de carimbos de data/hora sempre corresponde ao número de valores, e o valor para `Timestamps[x]` é `Values[x]`.

  Tipo: matriz de carimbos de data/hora

- Valores: os valores dos pontos de dados da métrica, correspondentes a `Timestamps`. O número de valores sempre corresponde ao número de carimbos de data/hora, e o valor para `Timestamps[x]` é `Values[x]`.

  Tipo: matriz de duplas

Para obter mais informações sobre objetos de `Error`, consulte as seções a seguir.

Formatos de resposta de erro

Opcionalmente, você pode usar a resposta de erro para fornecer mais informações sobre erros. Recomendamos que você retorne um erro com validação de código quando ocorrer um erro de validação, como quando um parâmetro está ausente ou é do tipo errado.

Veja a seguir um exemplo da resposta quando a função do Lambda deseja gerar uma exceção de validação `GetMetricData`.

```
{
   "Error": {
      "Code": "Validation",
      "Value": "Invalid Prometheus cluster"
   }
}
```

Veja a seguir um exemplo da resposta quando a função do Lambda indica que não consegue retornar dados devido a um problema de acesso. A resposta é convertida em uma única série temporal com um código de status de `Forbidden`.

```
{
   "Error": {
      "Code": "Forbidden",
      "Value": "Unable to access ..."
   }
}
```

Veja a seguir um exemplo de quando a função do Lambda gera uma exceção geral `InternalError`, que é convertida em uma única série temporal com um código de status de `InternalError` e uma mensagem. Sempre que um código de erro tem um valor diferente de `Validation` ou `Forbidden`, o CloudWatch pressupõe que trata-se de um erro interno genérico.

```
{
   "Error": {
      "Code": "PrometheusClusterUnreachable",
      "Value": "Unable to communicate with the cluster"
   }
}
```

DescribeGetMetricData event

Carga da solicitação

Veja a seguir um exemplo de carga da solicitação `DescribeGetMetricData`.

```
{
  "EventType": "DescribeGetMetricData"
}
```

Carga da resposta

Veja a seguir um exemplo de carga da resposta `DescribeGetMetricData`.

```
{
    "Description": "Data source connector",
    "ArgumentDefaults": [{
        Value: "default value"
```

```
    }]
}
```

- Descrição: uma descrição de como usar o conector da fonte de dados. Essa descrição será exibida no console do CloudWatch. O Markdown é compatível.

  Tipo: sequência
- ArgumentDefaults: a matriz opcional de valores padrão de argumentos usada preenche previamente o construtor da fonte de dados personalizada.

  Se `[{ Value: "default value 1"}, { Value: 10}]` for retornado, o construtor de consultas no console do CloudWatch exibirá duas entradas: a primeira com “valor padrão 1” e a segunda com 10.

  Se `ArgumentDefaults` não for fornecida, uma única entrada será exibida com o padrão de tipo definido como `String`.

  Tipo: matriz de objetos contendo Valor e Tipo.
- Erro: (opcional) um campo de erro pode ser incluído em qualquer resposta. Você pode ver exemplos em Evento GetMetricData.

Considerações importantes para alarmes do CloudWatch

Se você usar a fonte de dados para definir alarmes do CloudWatch, deverá configurá-la para relatar dados com carimbos e data/hora a cada minuto para o CloudWatch. Para obter mais informações e outras considerações sobre a criação de alarmes com base em métricas de fontes de dados conectadas, consulte Criação de um alarme com base em uma fonte de dados conectada.

(Opcional) Usar o AWS Secrets Manager para armazenar credenciais

Se a função do Lambda precisar usar credenciais para acessar a fonte de dados, recomendamos usar o AWS Secrets Manager para armazenar essas credenciais em vez de codificá-las na função do Lambda. Para obter mais informações sobre como usar o AWS Secrets Manager com o Lambda, consulte Usar segredos do AWS Secrets Manager em funções do AWS Lambda.

(Opcional) Conecte-se a uma fonte de dados em uma VPC

Se sua fonte de dados estiver em uma VPC gerenciada pela Amazon Virtual Private Cloud, você deverá configurar sua função do Lambda para acessá-la. Para obter mais informações, consulte Como conectar as redes de saída aos recursos em uma VPC.

Talvez você também precise configurar endpoints de serviço da VPC para acessar serviços, como o AWS Secrets Manager. Para obter mais informações, consulte Acessar um serviço da AWS usando um endpoint da VPC de interface.

Etapa 2: criar uma política de permissões do Lambda

Você deve criar uma declaração de política que conceda permissão ao CloudWatch para usar a função do Lambda que você criou. Você pode usar a AWS CLI ou o console do Lambda para criar a declaração de política.

Como usar a AWS CLI para criar a declaração de política

- Insira o comando a seguir. Substitua *123456789012* pelo ID da sua conta, substitua *my-data-source-function* pelo nome da função do Lambda e substitua *MyDataSource-DataSourcePermission1234* por um valor exclusivo arbitrário.

```
aws lambda add-permission --function-name my-data-source-function --statement-
id MyDataSource-DataSourcePermission1234 --action lambda:InvokeFunction --principal
 lambda.datasource.cloudwatch.amazonaws.com --source-account 123456789012
```

Etapa 3: anexar uma tag de recurso à função do Lambda

O console do CloudWatch determina quais das funções do Lambda são conectores de fontes de dados usando uma tag. Quando você cria uma fonte de dados usando um dos assistentes, a tag é aplicada automaticamente pela pilha do AWS CloudFormation que a configura. Ao criar uma fonte de dados por conta própria, você pode usar a tag a seguir para a função do Lambda. Isso faz com que o conector seja exibido na lista suspensa Fonte de dados no console do CloudWatch quando você consulta métricas.

- Uma tag com `cloudwatch:datasource` como chave e `custom` como valor.

Usar sua fonte de dados personalizada

Depois de criar uma fonte de dados, você pode usá-la para consultar dados dessa fonte para visualizá-los e definir alarmes. Caso tenha usado o modelo para criar o conector da fonte de dados personalizada ou tenha adicionado a tag indicada em Etapa 3: anexar uma tag de recurso à função do Lambda, siga as etapas em Criar um gráfico de métricas com base em outra fonte de dados para consultá-la.

Você também poderá usar a função de matemática de métricas LAMBDA para consultá-la, conforme será explicado na seção a seguir.

Para obter informações sobre como criar um alarme em métricas de uma fonte de dados, consulte Criação de um alarme com base em uma fonte de dados conectada.

Como passar argumentos para sua função do Lambda

A forma recomendada de passar argumentos para a fonte de dados personalizada é usar o construtor de consultas no console do CloudWatch ao consultar a fonte de dados.

Você também pode usar a função do Lambda para recuperar dados da fonte de dados usando a nova expressão LAMBDA na matemática de métricas do CloudWatch.

```
LAMBDA("LambdaFunctionName" [, optional-arg]*)
```

`optional-arg` tem até 20 strings, números ou booleanos. Por exemplo, `param`, `3.14` ou `true`.

> **Note**
>
> Strings de várias linhas não são compatíveis com os conectores de fonte de dados do CloudWatch. Cada feed de linha é substituído por um espaço quando a consulta é executada ou quando você cria um alarme ou um widget de painel com a consulta. Em alguns casos, isso pode tornar a consulta inválida.

Ao usar a função matemática de métricas LAMBDA, você pode fornecer o nome da função (`"MyFunction"`). Se sua política de recursos permitir, você também poderá usar uma versão específica da função (`"MyFunction:22"`) ou um alias (`"MyFunction:MyAlias"`) da função do Lambda. Não é possível usar *

Veja a seguir alguns exemplos de chamada para a função LAMBDA.

```
LAMBDA("AmazonOpenSearchDataSource", "MyDomain", "some-query")
```

```
LAMBDA("MyCustomDataSource", true, "fuzzy", 99.9)
```

A função matemática de métricas LAMBDA retorna uma lista de séries temporais que podem ser retornadas ao solicitante ou combinadas com outras funções matemáticas de métricas. Veja a seguir um exemplo de combinação de LAMBDA com outras funções matemáticas de métricas.

```
FILL(LAMBDA("AmazonOpenSearchDataSource", "MyDomain", "some-query"), 0)
```

Excluir um conector de uma fonte de dados

Para excluir um conector de uma fonte de dados, siga as instruções mostradas nesta seção.

Como excluir um conector de uma fonte de dados

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Configurações.
3. Escolha a guia Fontes de dados de métricas.
4. Escolha Gerenciar no CloudFormation na linha da fonte de dados que você deseja excluir.

   Você será direcionado para o console do AWS CloudFormation.
5. Na seção com o nome da sua fonte de dados, escolha Excluir.
6. No pop-up de confirmação, escolha Excluir.

Coletar métricas, logs e rastreamentos com o agente do CloudWatch

O atendente unificado do CloudWatch permite que você:

- Colete métricas internas no nível do sistema de instâncias do Amazon EC2 entre sistemas operacionais. As métricas podem incluir métricas de convidados, além das métricas para instâncias do EC2. As métricas adicionais que podem ser coletadas são listadas em Métricas coletadas pelo atendente do CloudWatch.
- Colete métricas no nível do sistema dos servidores on-premises. Isso pode incluir servidores em um ambiente híbrido, bem como servidores não gerenciados pela AWS.
- Recupere métricas personalizadas de suas aplicações ou serviços usando os protocolos `StatsD` e `collectd`. O protocolo `StatsD` é compatível com os servidores Linux e que executam o Windows Server. `collectd` tem suporte somente em servidores Linux.
- Colete os logs das instâncias do Amazon EC2 e dos servidores on-premises que executam o Linux ou o Windows Server.

  > **Note**
  >
  > O atendente do CloudWatch não é compatível com a coleta de logs de pipes FIFO.

- A versão 1.300031.0 e as versões posteriores podem ser usadas para habilitar o CloudWatch Application Signals. Para ter mais informações, consulte Application Signals.
- A versão 1.300025.0 e posteriores podem coletar rastros dos SDKs do cliente OpenTelemetry ou do X-Ray e enviá-los ao X-Ray.

  O uso do agente do CloudWatch permite coletar rastreamentos sem a necessidade de executar um daemon separado para a coleta de rastreamento, o que ajuda a reduzir o número de agentes que você executa e gerencia.

É possível armazenar e visualizar as métricas que você coletar com o atendente do CloudWatch no CloudWatch da mesma forma como faz com qualquer outra métrica do CloudWatch. O namespace padrão para métricas coletadas pelo atendente do CloudWatch é `CWAgent`, embora seja possível especificar um namespace diferente quando você configura o atendente.

Os logs coletados pelo atendente unificado do CloudWatch são processados e armazenados no Amazon CloudWatch Logs, da mesma forma como os logs coletados pelo atendente mais antigo do CloudWatch Logs. Para obter informações sobre o preço do CloudWatch Logs, consulte Preço do Amazon CloudWatch.

As métricas coletadas pelo atendente do CloudWatch são cobradas como métricas personalizadas. Para obter mais informações sobre o preço de métricas do CloudWatch, consulte Preço do Amazon CloudWatch.

O atendente do CloudWatch tem código aberto sob a licença do MIT e é hospedado no GitHub. Para desenvolver, personalizar ou contribuir com o atendente do CloudWatch, consulte o repositório do GitHub para obter as instruções mais recentes. Ao achar que encontrou um possível problema de segurança, não o publique no GitHub ou em qualquer fórum público. Em vez disso, siga as instruções em Relatório de vulnerabilidade ou envie um e-mail diretamente para a segurança da AWS.

As etapas desta seção explicam como instalar o atendente unificado do CloudWatch em instâncias do Amazon EC2 e em servidores on-premises. Para obter mais informações sobre as métricas que podem ser coletadas pelo atendente do CloudWatch, consulte Métricas coletadas pelo atendente do CloudWatch.

Sistemas operacionais com suporte

Há suporte para o agente do CloudWatch para a arquitetura x86-64 nos sistemas operacionais a seguir. Também há suporte para todas as atualizações de versões secundárias de cada uma das versões principais listadas aqui.

- Amazon Linux 2023
- Amazon Linux 2
- Ubuntu Server versões 23.10, 22.04, 20.04, 18.04, 16.04 e 14.04
- CentOS versões 9, 8 e 7
- Red Hat Enterprise Linux (RHEL) versões 9, 8 e 7
- Debian versões 12, 11 e 10
- SUSE Linux Enterprise Server (SLES) versões 15 e 12
- Oracle Linux versões 9, 8 e 7
- AlmaLinux versões 9 e 8
- Rocky Linux versões 9 e 8

- Os computadores macOS a seguir: instâncias Mac1 EC2 M1 e computadores executando macOS 14 (Sonoma), macOS 13 (Ventura), e macOS 12 (Monterey)
- Versões de 64 bits do Windows Server 2022, Windows Server 2019 e Windows Server 2016
- Windows 10 de 64 bits

Também há suporte para o agente para a arquitetura ARM64 nos sistemas operacionais a seguir. Também há suporte para todas as atualizações de versões secundárias de cada uma das versões principais listadas aqui.

- Amazon Linux 2023
- Amazon Linux 2
- Ubuntu Server versões 23.10, 22.04, 20.04, 18.04 e 16.04
- CentOS versões 9 e 8
- Red Hat Enterprise Linux (RHEL) versões 9, 8 e 7
- Debian versões 12, 11 e 10
- SUSE Linux Enterprise Server 15
- Os computadores macOS a seguir: macOS 14 (Sonoma), macOS 13 (Ventura), e macOS 12 (Monterey)

Visão geral do processo de instalação

Você pode baixar e instalar o atendente do CloudWatch manualmente usando a linha de comando ou integrá-lo com o SSM. O fluxo geral de instalação do atendente do CloudWatch usando qualquer método é o seguinte:

1. Criar funções ou usuários do IAM que permitem que o atendente colete métricas do servidor e, opcionalmente, integrar com o AWS Systems Manager.
2. Fazer download do pacote do atendente.
3. Modificar o arquivo de configuração do atendente do CloudWatch e especificar as métricas que você deseja coletar.
4. Instalar e iniciar o atendente em seus servidores. Ao instalar o atendente em uma instância do EC2, associe a função do IAM que você criou na etapa 1. Ao instalar o atendente em um servidor on-premises, especifique um perfil nomeado que contenha as credenciais do usuário do IAM que você criou na etapa 1.

Conteúdo

Instalação do atendente do CloudWatch

O agente do CloudWatch está disponível como um pacote no Amazon Linux 2023 e Amazon Linux 2. Caso esteja usando um desses sistemas operacionais, você poderá instalar o pacote digitando o comando a seguir. Também é necessário garantir que a função do IAM anexada à instância tenha o atributo CloudWatchAgentServerPolicy anexado. Para mais informações, consulte Criar funções do IAM a serem usadas com o atendente do CloudWatch em instâncias do Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

Em todos os sistemas operacionais compatíveis, incluindo Linux e Windows Server, você pode fazer download e instalar o atendente do CloudWatch usando a linha de comando com um link de download do Amazon S3, o Amazon EC2 Systems Manager ou um modelo do AWS CloudFormation. Para mais detalhes, consulte as seções a seguir.

Conteúdo

Instalar o atendente do CloudWatch usando a linha de comando

Use os tópicos a seguir para baixar, configurar e instalar o pacote do atendente do CloudWatch.

Tópicos

Baixar e configurar o atendente do CloudWatch usando a linha de comando

Use as seguintes etapas para baixar o pacote do atendente do CloudWatch, criar funções ou usuários do IAM e opcionalmente, modificar o arquivo de configuração comum.

Baixar o pacote do atendente do CloudWatch

> **Note**
>
> Para baixar o agente do CloudWatch, sua conexão deve usar o TLS 1.2 ou posterior.

O agente do CloudWatch está disponível como um pacote no Amazon Linux 2023 e Amazon Linux 2. Caso esteja usando esse sistema operacional, você pode instalar o pacote digitando o comando a seguir. Também é necessário garantir que a função do IAM anexada à instância tenha o atributo CloudWatchAgentServerPolicy anexado. Para mais informações, consulte Criar funções e usuários do IAM para uso com o atendente do CloudWatch.

```
sudo yum install amazon-cloudwatch-agent
```

Em todos os sistemas operacionais compatíveis, é possível baixar e instalar o atendente do CloudWatch usando a linha de comando.

Para cada link de download, há um link geral, bem como links para cada região. Por exemplo, para o Amazon Linux 2023 e Amazon Linux 2 e a arquitetura x86-64, três dos links de download válidos são:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

Também é possível fazer download de um arquivo README sobre as alterações mais recentes no atendente e de um arquivo que indica o número da versão que está disponível para download. Esses arquivos estão nos seguintes locais:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/RELEASE_NOTES` ou `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/RELEASE_NOTES`
- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/CWAGENT_VERSION` ou `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/CWAGENT_VERSION`

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
x86-64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazon	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/centos/amd64/latest/a

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
		s.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm	mazon-cloudwatch-agent.rpm.sig
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig
ARM64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	MacOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/arm64/latest/amazon-cloudwatch-agent.pkg https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/arm64/latest/amazon-cloudwatch-agent.pkg	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/arm64/latest/amazon-cloudwatch-agent.pkg.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/arm64/latest/amazon-cloudwatch-agent.pkg.sig

Como usar a linha de comando para baixar e instalar o pacote do atendente do CloudWatch

1. Baixe o atendente do CloudWatch.

Em um servidor Linux, digite o seguinte. Em *download-link*, use o link para download apropriado na tabela anterior.

```
wget download-link
```

Em um servidor que execute o Windows Server, faça download do seguinte arquivo:

```
https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi
```

2. Depois de fazer download do pacote, você pode, opcionalmente, verificar a assinatura do pacote. Para ter mais informações, consulte Verificar a assinatura do pacote do atendente do CloudWatch.
3. Instale o pacote . Se você tiver obtido por download um pacote RPM em um servidor Linux, mude para o diretório que contém o pacote e digite o seguinte:

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

Se você tiver obtido por download um pacote DEB em um servidor Linux, mude para o diretório que contém o pacote e digite o seguinte:

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Se você tiver obtido por download um pacote MSI em um servidor que esteja executando o Windows Server, mude para o diretório que contém o pacote e digite o seguinte:

```
msiexec /i amazon-cloudwatch-agent.msi
```

Esse comando também funciona no PowerShell. Para obter mais informações sobre opções de comando MSI, consulte Opções de linha de comando na documentação do Microsoft Windows.

Se você tiver obtido por download um pacote PKG em um servidor macOS, mude para o diretório que contém o pacote e digite o seguinte:

```
sudo installer -pkg ./amazon-cloudwatch-agent.pkg -target /
```

Criar e modificar o arquivo de configuração do atendente

Depois de baixar o atendente do CloudWatch, você deverá criar o arquivo de configuração antes de iniciar o atendente em qualquer servidor. Para ter mais informações, consulte Criar o arquivo de configuração do atendente do CloudWatch.

Criar funções e usuários do IAM para uso com o atendente do CloudWatch

O acesso aos recursos da AWS requer permissões. Você cria uma função do IAM, um usuário do IAM ou ambos para conceder permissões necessárias para o atendente do CloudWatch gravar métricas no CloudWatch. Se você pretende usar o atendente em instâncias do Amazon EC2, crie uma função do IAM. Se você pretende usar o atendente em servidores on-premises, é necessário criar um usuário do IAM.

> **Note**
>
> Os seguintes procedimentos foram modificados recentemente usando as novas políticas `CloudWatchAgentServerPolicy` e `CloudWatchAgentAdminPolicy` criadas pela Amazon, em vez de exigir que os próprios clientes criem essas políticas. Para gravar arquivos e baixar arquivos do Parameter Store, as políticas criadas pela Amazon oferecem suporte apenas a arquivos com nomes que começam com `AmazonCloudWatch-`. Caso você tenha um arquivo de configuração do atendente do CloudWatch com um nome de arquivo que não comece com `AmazonCloudWatch-`, essas políticas não poderão ser usadas para gravar o arquivo no Parameter Store nem baixá-lo no Parameter Store.

Se você executar o atendente do CloudWatch em instâncias do Amazon EC2, use as etapas a seguir para criar a função do IAM necessária. Essa função fornece permissões para ler informações da instância e gravá-las no CloudWatch.

Para criar a função do IAM necessária para executar o atendente do CloudWatch em instâncias do EC2

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Roles (Funções) e, depois, Create role (Criar função).
3. Verifique se o serviço da AWS está selecionado em Trusted entity type (Tipo de entidade confiável).

4. Em Use case (Caso de uso), escolha EC2 em Common use cases (Casos de uso comuns),
5. Escolha Próximo.
6. Na lista de políticas, marque a caixa de seleção ao lado de CloudWatchAgentServerPolicy. Se necessário, use a caixa de pesquisa para encontrar a política.
7. (Opcional) Se o agente estiver enviando rastreamentos para o X-Ray, você também precisará atribuir ao perfil a política AWSXRayDaemonWriteAccess. Para fazer isso, localize a política na lista e marque a caixa de seleção ao lado dela.
8. Escolha Próximo.
9. Em Role name (Nome da função), digite um nome para a função, como *`CloudWatchAgentServerRole`*. Como opção, atribua uma descrição a ela. Então, escolha Criar perfil.

   A função foi criada.
10. (Opcional) Se o atendente for enviar logs para o CloudWatch Logs e você quiser que ele possa definir políticas de retenção para esses grupos de log, será necessário adicionar a permissão `logs:PutRetentionPolicy` para a função. Para ter mais informações, consulte Permitir que o atendente do CloudWatch defina a política de retenção de logs.

Se você executar o atendente do CloudWatch em servidores on-premises, use as etapas a seguir para criar o usuário do IAM necessário.

> **Warning**
>
> Este cenário precisa de usuários do IAM com acesso programático e credenciais de longo prazo, o que representa um risco de segurança. Para ajudar a reduzir esse risco, recomendamos que você forneça a esses usuários somente as permissões necessárias para realizar a tarefa e que você os remova quando não forem mais necessários. As chaves de acesso podem ser atualizadas, se necessário. Para obter mais informações, consulte Atualização de chaves de acesso no Guia de usuário do IAM.

Para criar o usuário do IAM necessário para o atendente do CloudWatch ser executado em servidores on-premises

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.

2. No painel de navegação à esquerda, escolha Users (Usuários) e, depois, Add users (Adicionar usuários).
3. Digite o nome para o novo usuário.
4. Selecione Access key – Programmatic access (Chave de acesso – Acesso programático) e escolha Next: Permissions (Próximo: Permissões).
5. Escolha Anexar políticas existentes diretamente.
6. Na lista de políticas, marque a caixa de seleção ao lado de CloudWatchAgentServerPolicy. Se necessário, use a caixa de pesquisa para encontrar a política.
7. (Opcional) Se o agente for rastrear o X-Ray, você também precisará atribuir ao perfil a política AWSXRayDaemonWriteAccess. Para fazer isso, localize a política na lista e marque a caixa de seleção ao lado dela.
8. Escolha Próximo: etiquetas.
9. Ou crie etiquetas para o novo usuário do IAM e, em seguida, escolha Next:Review (Próximo:Análise).
10. Confirme se as políticas corretas estão listadas e selecione Create user (Criar usuário).
11. Ao lado do nome no novo usuário, escolha Mostrar. Copie a chave de acesso e a chave secreta em um arquivo para que você possa usá-las ao instalar o atendente. Escolha Fechar.

Permitir que o atendente do CloudWatch defina a política de retenção de logs

É possível configurar o atendente do CloudWatch para definir a política de retenção para grupos de logs para os quais ele envia eventos de log. Se você fizer isso, deve conceder o `logs:PutRetentionPolicy` à função do IAM ou ao usuário que o atendente usa. O atendente usa uma função do IAM para executar em instâncias do Amazon EC2 e usa um usuário do IAM para servidores on-premises.

Para conceder permissão à função do IAM do atendente do CloudWatch para definir políticas de retenção de log

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Roles.
3. Na caixa de pesquisa, digite o início do nome da função do IAM do atendente do CloudWatch. Você escolheu esse nome ao criar a função. Ele pode se chamar `CloudWatchAgentServerRole`.

Quando você vir a função, escolha o nome da função.

4. Na guia Permissions (Permissões), escolha Add permissions (Adicionar permissões), Create inline policy (Criar política em linha).
5. Escolha a guia JSON e copie a seguinte política na caixa, substituindo o JSON padrão na caixa:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:PutRetentionPolicy",
      "Resource": "*"
    }
  ]
}
```

6. Escolha Revisar política.
7. Em Name (Nome), insira **`CloudWatchAgentPutLogsRetention`** ou algo semelhante e escolha Create policy (Criar política).

Para conceder permissão ao usuário do IAM do atendente do CloudWatch para definir políticas de retenção de log

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Usuários.
3. Na caixa de pesquisa, digite o início do nome do usuário do IAM do atendente do CloudWatch. Você escolheu esse nome ao criar o usuário.

   Quando você vir o usuário, escolha o nome do usuário.
4. Na guia Permissions (Permissões), escolha Add inline policy (Adicionar política em linha).
5. Escolha a guia JSON e copie a seguinte política na caixa, substituindo o JSON padrão na caixa:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": "logs:PutRetentionPolicy",
      "Resource": "*"
    }
  ]
}
```

6. Escolha Revisar política.
7. Em Name (Nome), insira **`CloudWatchAgentPutLogsRetention`** ou algo semelhante e escolha Create policy (Criar política).

Como instalar e executar o atendente do CloudWatch em seus servidores

Depois de criar o arquivo de configuração do atendente que você deseja e criar uma função do IAM ou um usuário do IAM, use as seguintes etapas para instalar e executar o atendente em seus servidores, usando essa configuração. Primeiro, associe uma função do IAM ou um usuário do IAM ao servidor que executará o atendente. Depois, nesse servidor, faça download do pacote do atendente e inicie-o usando a configuração do atendente que você criou.

Baixar o pacote do atendente do CloudWatch usando um link para download do S3

> **Note**
>
> Para baixar o agente do CloudWatch, sua conexão deve usar o TLS 1.2 ou posterior.

É necessário instalar o atendente em cada servidor no qual você executará o atendente.

AMIs do Amazon Linux

O agente do CloudWatch está disponível como um pacote no Amazon Linux 2023 e Amazon Linux 2. Caso esteja usando esse sistema operacional, você pode instalar o pacote digitando o comando a seguir. Também é necessário garantir que a função do IAM anexada à instância tenha o atributo CloudWatchAgentServerPolicy anexado. Para mais informações, consulte Criar funções do IAM a serem usadas com o atendente do CloudWatch em instâncias do Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

Todos os sistemas operacionais

Em todos os sistemas operacionais compatíveis, é possível baixar e instalar o atendente do CloudWatch usando a linha de comando com um link para baixar o Amazon S3, conforme descrito nas etapas seguintes.

Para cada link de download, há um link geral, bem como links para cada região. Por exemplo, para o Amazon Linux 2023 e Amazon Linux 2 e a arquitetura x86-64, três dos links de download válidos são:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
x86-64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazon	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaw

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
		s.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm	s.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazons.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazons.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazons.com/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig
ARM64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Como usar a linha de comando para instalar o atendente do CloudWatch em uma instância do Amazon EC2

1. Baixe o atendente do CloudWatch. Para um servidor Linux, digite o seguinte. Em *download-link*, use o link para download apropriado na tabela anterior.

   ```
   wget download-link
   ```

   Para um servidor que execute o Windows Server, faça download do seguinte arquivo:

   ```
   https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi
   ```

2. Depois de fazer download do pacote, você pode, opcionalmente, verificar a assinatura do pacote. Para ter mais informações, consulte Verificar a assinatura do pacote do atendente do CloudWatch.
3. Instale o pacote . Se você tiver obtido por download um pacote RPM em um servidor Linux, mude para o diretório que contém o pacote e digite o seguinte:

   ```
   sudo rpm -U ./amazon-cloudwatch-agent.rpm
   ```

   Se você tiver obtido por download um pacote DEB em um servidor Linux, mude para o diretório que contém o pacote e digite o seguinte:

   ```
   sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
   ```

   Se você tiver obtido por download um pacote MSI em um servidor que esteja executando o Windows Server, mude para o diretório que contém o pacote e digite o seguinte:

   ```
   msiexec /i amazon-cloudwatch-agent.msi
   ```

   Esse comando também funciona no PowerShell. Para obter mais informações sobre opções de comando MSI, consulte Opções de linha de comando na documentação do Microsoft Windows.

(Instalar uma instância do EC2) Associar uma função do IAM

Para permitir que o atendente do CloudWatch envie dados da instância, associe uma função do IAM à instância. A função a ser anexada é CloudWatchAgentServerRole. Você deveria ter criado esse perfil anteriormente. Para ter mais informações, consulte Criar funções e usuários do IAM para uso com o atendente do CloudWatch.

Para obter mais informações sobre como anexar uma função do IAM a uma instância, consulte Anexar uma função do IAM a uma instância, no Manual do usuário do Amazon EC2 para instâncias do Windows.

(Instalar um servidor on-premises) Especificar credenciais do IAM e região da AWS

Para permitir que o CloudWatch envie dados de um servidor on-premises, é necessário especificar a chave de acesso e a chave secreta do usuário do IAM que você criou anteriormente. Para obter mais informações sobre como criar esse usuário, consulte Criar funções e usuários do IAM para uso com o atendente do CloudWatch.

Você também deve especificar a região da AWS para a qual enviar as métricas, usando o campo `region` na seção `[AmazonCloudWatchAgent]` do arquivo de configuração da AWS, como no exemplo a seguir.

```
[profile AmazonCloudWatchAgent]
region = us-west-1
```

A seguir, temos um exemplo de uso do comando `aws configure` para criar um perfil nomeado para o atendente do CloudWatch. Esse exemplo pressupõe que você esteja usando o nome de perfil padrão do `AmazonCloudWatchAgent`.

Para criar o perfil AmazonCloudWatchAgent para o atendente do CloudWatch

1. Caso ainda não tenha feito isso, instale a AWS Command Line Interface. Para obter mais informações, consulte Instalar a AWS CLI.
2. Em servidores Linux, digite este comando e siga os avisos:

   ```
   sudo aws configure --profile AmazonCloudWatchAgent
   ```

   No Windows Server, abra o PowerShell como administrador, digite o seguinte comando e siga as instruções.

   ```
   aws configure --profile AmazonCloudWatchAgent
   ```

Verificar o acesso à Internet

Suas instâncias do Amazon EC2 devem ter acesso à Internet de saída para enviar dados ao CloudWatch ou ao CloudWatch Logs. Para obter mais informações sobre como configurar o acesso à Internet, consulte Gateways da Internet no Manual do usuário da Amazon VPC.

Os endpoints e portas para configurar em seu proxy são os seguintes:

- Se estiver usando o atendente para coletar métricas, você deverá incluir os endpoints do CloudWatch das regiões apropriadas na lista de permissões. Esses endpoints estão listados em Endpoints e cotas do Amazon CloudWatch.
- Se estiver usando o atendente para coletar métricas, você deverá incluir os endpoints de logs do CloudWatch das regiões apropriadas na lista de permissões. Esses endpoints estão listados em Endpoints e cotas do Amazon CloudWatch Logs.

- Se estiver usando o Systems Manager para instalar o atendente ou o Parameter Store para armazenar o arquivo de configuração, você deverá adicionar os endpoints do Systems Manager das regiões apropriadas na lista de permissões. Esses endpoints estão listados em Endpoints e cotas do AWS Systems Manager.

(Opcional) Modificar a configuração comum para informações de proxy ou região

O atendente do CloudWatch inclui um arquivo de configuração chamado `common-config.toml`. Você pode, opcionalmente, usar esse arquivo para especificar as informações de proxy e região.

Em um servidor que executa o Linux, esse arquivo encontra-se no diretório `/opt/aws/amazon-cloudwatch-agent/etc`. Em um servidor que executa o Windows Server, esse arquivo encontra-se no diretório `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

> **Note**
>
> Recomendamos usar o arquivo `common-config.toml` para fornecer configurações e credenciais compartilhadas ao executar o agente do CloudWatch em modo on-premises, e ele também pode ser útil quando você estiver realizando execuções no Amazon EC2 e desejar reutilizar perfis e arquivos de credenciais compartilhados existentes. Realizar a habilitação por meio do arquivo `common-config.toml` tem a vantagem adicional de que, se o arquivo de credenciais compartilhadas for alternado com credenciais renovadas após a expiração, as novas credenciais serão automaticamente obtidas pelo agente sem a necessidade de reinicialização.

O `common-config.toml` padrão é o seguinte.

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for the on-premises case by
 default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"
```

```
## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Todas as linhas são comentadas inicialmente. Para definir o perfil de credencial ou as configurações de proxy, remova o # da linha e especifique um valor. Você pode editar esse arquivo manualmente ou usando o Run Command `RunShellScript` no Systems Manager:

- `shared_credential_profile`: para servidores on-premises, essa linha especifica o perfil de credenciais do usuário do IAM a ser usado para enviar dados ao CloudWatch. Se você mantiver esta linha comentada, `AmazonCloudWatchAgent` será usado. Para obter mais informações sobre a criação desse perfil, consulte (Instalar um servidor on-premises) Especificar credenciais do IAM e região da AWS.

  Em uma instância do EC2, você poderá usar essa linha para que o atendente do CloudWatch envie dados dessa instância para o CloudWatch em outra região da AWS. Para fazer isso, especifique um perfil nomeado que inclua um campo `region` especificando o nome da região para a qual enviar.

  Se você especificar um `shared_credential_profile`, também deverá remover o símbolo # do início da linha `[credentials]`.
- `shared_credential_file`: para que o atendente procure credenciais em um arquivo localizado em um caminho diferente do padrão, especifique esse caminho completo e o nome do arquivo aqui. O caminho padrão é `/root/.aws` no Linux e `C:\\Users\\Administrator\\.aws` no Windows Server.

  O primeiro exemplo a seguir mostra a sintaxe de uma linha `shared_credential_file` válida para servidores Linux, e o segundo exemplo é válido para o Windows Server. No Windows Server, você deve fazer o escape dos caracteres \.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

```
shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
```

Se você especificar um `shared_credential_file`, também deverá remover o símbolo # do início da linha `[credentials]`.

- Configurações do proxy: se seus servidores usarem os proxies HTTP ou HTTPS para entrar em contato com os produtos da AWS, especifique esses proxies nos campos `http_proxy` e `https_proxy`. Se houver URLs que devem ser excluídas do proxy, as especifique no campo `no_proxy`, separadas por vírgulas.

Iniciar o atendente do CloudWatch usando a linha de comando

Siga estas etapas para usar a linha de comando para iniciar o atendente do CloudWatch em um servidor.

Para usar a linha de comando para iniciar o atendente do CloudWatch em um servidor

1. Copie o arquivo de configuração do atendente que você deseja usar no servidor onde você irá executar o atendente. Anote o nome do caminho onde você o copiar.
2. Nesse comando, `-a fetch-config` faz com que o atendente carregue a última versão do arquivo de configuração do atendente do CloudWatch e o `-s` inicia o atendente.

   Insira um dos seguintes comandos: Substitua *`configuration-file-path`* pelo caminho para o arquivo de configuração do atendente. Este arquivo é chamado `config.json`, se você o criou com o assistente, e pode ser chamado `amazon-cloudwatch-agent.json`, se você o criou manualmente.

   Em uma instância do EC2 que execute o Linux, insira o comando a seguir.

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:configuration-file-path
   ```

   Em um servidor on-premises que executa o Linux, digite o seguinte:

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m onPremise -s -c file:configuration-file-path
   ```

Em uma instância do EC2 que execute o Windows Server, digite o seguinte no console do PowerShell:

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m ec2 -s -c file:configuration-file-path
```

Em um servidor on-premises que executa o Windows Server, digite o seguinte no console do PowerShell:

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m onPremise -s -c file:configuration-file-path
```

Instalar o atendente do CloudWatch usando o AWS Systems Manager

Use os tópicos a seguir para instalar e executar o atendente do CloudWatch usando o AWS Systems Manager.

Tópicos

Criar funções e usuários do IAM para uso com o atendente do CloudWatch

O acesso aos recursos da AWS requer permissões. É possível criar funções e usuários do IAM que incluem as permissões necessárias para o atendente do CloudWatch gravar métricas no CloudWatch e para o atendente do CloudWatch se comunicar com o Amazon EC2 e o AWS Systems Manager. Utilize funções do IAM nas instâncias do Amazon EC2 e usuários do IAM em servidores on-premises.

Uma função ou um usuário permite que o atendente do CloudWatch seja instalado em um servidor e envie métricas ao CloudWatch. A outra função ou o outro usuário é necessário para armazenar sua configuração do atendente do CloudWatch no Systems Manager Parameter Store. O Parameter Store permite que vários servidores usem uma configuração do atendente do CloudWatch.

A capacidade de gravação no Parameter Store é uma permissão ampla e avançada. Você deve usá-la apenas quando necessário e não deve associá-la a várias instâncias em sua implantação. Se você armazenar sua configuração do atendente do CloudWatch no Parameter Store, recomendamos o seguinte:

- Configure uma instância na qual executar essa configuração.
- Use a função do IAM com permissões para gravar no Parameter Store somente nessa instância.
- Use a função do IAM com permissões para gravar no Parameter Store somente enquanto estiver trabalhando com o arquivo de configuração do atendente do CloudWatch e salvando-o.

> **Note**
>
> Os seguintes procedimentos foram modificados recentemente usando as novas políticas `CloudWatchAgentServerPolicy` e `CloudWatchAgentAdminPolicy` criadas pela Amazon, em vez de exigir que os próprios clientes criem essas políticas. Para usar essas políticas a fim de gravar o arquivo de configuração do atendente no Parameter Store e baixá-lo do Parameter Store, o nome desse arquivo deverá começar com `AmazonCloudWatch-`. Caso você tenha um arquivo de configuração do atendente do CloudWatch com um nome de arquivo que não comece com `AmazonCloudWatch-`, essas políticas não poderão ser usadas para gravar o arquivo no Parameter Store nem baixá-lo do Parameter Store.

Criar funções do IAM a serem usadas com o atendente do CloudWatch em instâncias do Amazon EC2

O primeiro procedimento cria a função do IAM que você precisa associar a cada instância do Amazon EC2 que executará o atendente do CloudWatch. Essa função fornece permissões para ler informações da instância e gravá-las no CloudWatch.

O segundo procedimento cria a função do IAM que você deve anexar à instância do Amazon EC2 que está sendo usada para criar o arquivo de configuração do atendente do CloudWatch. Esta etapa será necessária se você for armazenar esse arquivo no Systems Manager Parameter Store para que outros servidores possam usá-lo. Essa função fornece permissões para gravar no Parameter Store, além de permissões para ler informações da instância e gravá-las no CloudWatch. Essa função inclui permissões suficientes para executar o atendente do CloudWatch, bem como para gravar no Parameter Store.

> **Note**
>
> O Parameter Store oferece suporte a parâmetros nos níveis padrão e avançado. Esses níveis de parâmetro não estão relacionados aos níveis básico, padrão e avançado de detalhes disponíveis nos conjuntos de métricas predefinidas do atendente do CloudWatch.

Para criar a função do IAM necessária para cada servidor executar o atendente do CloudWatch

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Funções e Criar função.
3. Em Select type of trusted entity (Selecionar o tipo de entidade confiável), escolha AWS service (serviço).
4. Logo em Common use cases (Casos de uso comuns), selecione EC2 e Next: Permissions (Próximo: permissões).
5. Na lista de políticas, marque a caixa de seleção ao lado de CloudWatchAgentServerPolicy. Se necessário, use a caixa de pesquisa para encontrar a política.
6. Para usar o Systems Manager com o intuito de instalar ou configurar o atendente do CloudWatch, marque a caixa ao lado de AmazonSSMManagedInstanceCore. Essa política gerenciada da AWS permite que uma instância use a funcionalidade básica do serviço Systems Manager. Se necessário, use a caixa de pesquisa para encontrar a política. Essa política não é necessária ao iniciar e configurar o atendente apenas por meio da linha de comando.
7. Escolha Próximo: etiquetas.
8. (Opcional) Adicione um ou mais pares de chave-valor para organizar, rastrear ou controlar o acesso a esta função e escolha Next: Review (Próximo: revisar).
9. Em Role name (Nome da função), insira um nome para a nova função, como **`CloudWatchAgentServerRole`** ou outro nome que você preferir.
10. (Opcional) Em Role description (Descrição da função), insira uma descrição.
11. Confirme se CloudWatchAgentServerPolicy e, opcionalmente, AmazonSSMManagedInstanceCore aparecem ao lado de Policies (Políticas).
12. Selecione Criar função.

    A função foi criada.

O procedimento a seguir cria a função do IAM que também pode gravar no Parameter Store. É possível usar essa função para armazenar o arquivo de configuração do atendente no Parameter Store para que outros servidores possam recuperá-lo.

As permissões para gravar no Parameter Store fornecem acesso amplo. Essa função não deve ser associada a todos os servidores, e apenas os administradores devem usá-la. Depois de criar o arquivo de configuração do atendente e copiá-lo no Parameter Store, você deve desvincular essa função da instância e usar `CloudWatchAgentServerRole` no lugar dela.

Para criar a função do IAM para um administrador gravar no Parameter Store

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Funções e Criar função.
3. Em Select type of trusted entity (Selecionar o tipo de entidade confiável), escolha AWS service (serviço).
4. Imediatamente em Choose the service that will use this role (Escolher o serviço que usará essa função), selecione EC2 e Next: Permissions (Próximo: permissões).
5. Na lista de políticas, marque a caixa de seleção ao lado de CloudWatchAgentAdminPolicy. Se necessário, use a caixa de pesquisa para encontrar a política.
6. Para usar o Systems Manager com o intuito de instalar ou configurar o atendente do CloudWatch, marque a caixa ao lado de AmazonSSMManagedInstanceCore. Essa política gerenciada da AWS permite que uma instância use a funcionalidade básica do serviço Systems Manager. Se necessário, use a caixa de pesquisa para encontrar a política. Essa política não é necessária ao iniciar e configurar o atendente apenas por meio da linha de comando.
7. Escolha Próximo: etiquetas.
8. (Opcional) Adicione um ou mais pares de chave-valor para organizar, rastrear ou controlar o acesso a esta função e escolha Next: Review (Próximo: revisar).
9. Em Role name (Nome da função), insira um nome para a nova função, como **`CloudWatchAgentAdminRole`** ou outro nome que você preferir.
10. (Opcional) Em Role description (Descrição da função), insira uma descrição.
11. Confirme se CloudWatchAgentAdminPolicy e, opcionalmente, AmazonSSMManagedInstanceCore aparecem ao lado de Policies (Políticas).
12. Selecione Criar função.

    A função foi criada.

Criar usuários do IAM para usar com o atendente do CloudWatch em servidores on-premises

O primeiro procedimento cria o usuário do IAM necessário para executar o atendente do CloudWatch. Esse usuário fornece permissões para enviar dados ao CloudWatch.

O segundo procedimento cria o usuário do IAM que você pode usar ao criar o arquivo de configuração do atendente do CloudWatch. Use este procedimento para armazenar esse arquivo no Systems Manager Parameter Store para que outros servidores possam usá-lo. Esse usuário fornece permissões para gravar no Parameter Store, além de permissões para gravar dados no CloudWatch.

> **Note**
>
> O Parameter Store oferece suporte a parâmetros nos níveis padrão e avançado. Esses níveis de parâmetro não estão relacionados aos níveis básico, padrão e avançado de detalhes disponíveis nos conjuntos de métricas predefinidas do atendente do CloudWatch.

Para criar o usuário do IAM necessário para o atendente do CloudWatch gravar dados no CloudWatch

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Usuários e depois Adicionar usuário.
3. Digite o nome para o novo usuário.
4. Em Access type (Tipo de acesso), selecione Programmatic access (Acesso programático) e selecione Next: Permissions (Próximo: permissões).
5. Em Set permissions (Definir permissões), selecione Attach existing policies directly (Anexar políticas existentes diretamente).
6. Na lista de políticas, marque a caixa de seleção ao lado de CloudWatchAgentServerPolicy. Se necessário, use a caixa de pesquisa para encontrar a política.
7. Para usar o Systems Manager com o intuito de instalar ou configurar o atendente do CloudWatch, marque a caixa ao lado de AmazonSSMManagedInstanceCore. Essa política gerenciada da AWS permite que uma instância use a funcionalidade básica do serviço Systems Manager. (Se necessário, use a caixa de pesquisa para localizar a política. Essa política não será necessária se você iniciar e configurar o atendente somente por meio da linha de comando.)

8. Escolha Próximo: etiquetas.
9. (Opcional) Adicione um ou mais pares de chave-valor para organizar, rastrear ou controlar o acesso a esta função e escolha Next: Review (Próximo: revisar).
10. Confirme se as políticas corretas estão listadas e selecione Create user (Criar usuário).
11. Na linha para o novo usuário, selecione Show (Mostrar). Copie a chave de acesso e a chave secreta em um arquivo para que você possa usá-las ao instalar o atendente. Escolha Fechar.

O procedimento a seguir cria o usuário do IAM que também pode gravar no Parameter Store. Você precisará usar esse usuário do IAM se for armazenar o arquivo de configuração do atendente no Parameter Store para que outros servidores possam usá-lo. Esse usuário do IAM fornece permissões para gravar no Parameter Store. Esse usuário fornece permissões para ler informações da instância e gravá-las no CloudWatch. As permissões para gravar no Systems Manager Parameter Store fornecem acesso amplo. Esse usuário do IAM não deve ser associado a todos os servidores, e apenas os administradores devem usá-lo. Use esse usuário do IAM somente quando for armazenar o arquivo de configuração do atendente no Parameter Store.

Para criar o usuário do IAM necessário para armazenar o arquivo de configuração no Parameter Store e enviar informações ao CloudWatch

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Usuários e depois Adicionar usuário.
3. Digite o nome para o novo usuário.
4. Em Access type (Tipo de acesso), selecione Programmatic access (Acesso programático) e selecione Next: Permissions (Próximo: permissões).
5. Em Set permissions (Definir permissões), selecione Attach existing policies directly (Anexar políticas existentes diretamente).
6. Na lista de políticas, marque a caixa de seleção ao lado de CloudWatchAgentAdminPolicy. Se necessário, use a caixa de pesquisa para encontrar a política.
7. Para usar o Systems Manager com o intuito de instalar ou configurar o atendente do CloudWatch, marque a caixa ao lado de AmazonSSMManagedInstanceCore. Essa política gerenciada da AWS permite que uma instância use a funcionalidade básica do serviço Systems Manager. (Se necessário, use a caixa de pesquisa para localizar a política. Essa política não será necessária se você iniciar e configurar o atendente somente por meio da linha de comando.)

8. Escolha Próximo: etiquetas.
9. (Opcional) Adicione um ou mais pares de chave-valor para organizar, rastrear ou controlar o acesso a esta função e escolha Next: Review (Próximo: revisar).
10. Confirme se as políticas corretas estão listadas e selecione Create user (Criar usuário).
11. Na linha para o novo usuário, selecione Show (Mostrar). Copie a chave de acesso e a chave secreta em um arquivo para que você possa usá-las ao instalar o atendente. Escolha Fechar.

Baixar e configurar o atendente do CloudWatch

Esta seção explica como usar o Systems Manager para baixar o atendente e como criar seu arquivo de configuração do atendente. Antes de usar o Systems Manager para baixar o atendente, é necessário verificar se a instância está configurada corretamente para o Systems Manager.

Instalar ou atualizar o SSM Agent

Em uma instância do Amazon EC2, o atendente do CloudWatch exige que a instância esteja executando a versão 2.2.93.0 ou posterior. Antes de instalar o atendente do CloudWatch, atualize ou instale o SSM Agent na instância, caso ainda não tenha feito isso.

Para obter informações sobre como instalar ou atualizar o SSM Agent em uma instância que execute o Linux, consulte Instalar e configurar o SSM Agent em instâncias do Linux no Manual do usuário do AWS Systems Manager.

Para obter informações sobre como instalar ou atualizar o SSM Agent, consulte Trabalhar com o SSM Agent no Manual do usuário do AWS Systems Manager.

(Opcional) Verificar os pré-requisitos do Systems Manager

Verificar o acesso à Internet

Suas instâncias do Amazon EC2 devem ter acesso à Internet de saída para enviar dados ao CloudWatch ou ao CloudWatch Logs. Para obter mais informações sobre como configurar o acesso à Internet, consulte Gateways da Internet no Manual do usuário da Amazon VPC.

Os endpoints e portas para configurar em seu proxy são os seguintes:

- Se estiver usando o atendente para coletar métricas, você deverá incluir os endpoints do CloudWatch das regiões apropriadas na lista de permissões. Esses endpoints são listados no Amazon CloudWatch no Referência geral da Amazon Web Services.

- Se estiver usando o atendente para coletar logs, você deverá incluir os endpoints do CloudWatch Logs das regiões apropriadas na lista de permissões. Esses endpoints são listados no Amazon CloudWatch Logs no Referência geral da Amazon Web Services.
- Se estiver usando o Systems Manager para instalar o atendente ou o Parameter Store para armazenar o arquivo de configuração, você deverá incluir os endpoints do Systems Manager das regiões apropriadas na lista de permissões. Esses endpoints são listados no AWS Systems Manager no Referência geral da Amazon Web Services.

Use as seguintes etapas para baixar o pacote do atendente do CloudWatch usando o Systems Manager.

Para baixar o atendente do CloudWatch usando o Systems Manager

1. Abra o console do Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Executar comando.

   - ou -

   Se a página inicial do AWS Systems Manager for exibida, role para baixo e escolha Explore Run Command.
3. Selecione Run command.
4. Na lista Documento do comando, escolha AWS-ConfigureAWSPackage.
5. Na área Targets (Destinos), selecione a instância na qual o atendente do CloudWatch será instalado. Se você não visualizar uma instância específica, ela talvez não esteja configurada como uma instância gerenciada para uso com o Systems Manager. Para obter mais informações, consulte Configurar o AWS Systems Manager para ambientes híbridos no Manual do usuário do AWS Systems Manager.
6. Na lista Ação, escolha Instalar.
7. No campo Name (Nome), digite `AmazonCloudWatchAgent`.
8. Deixe a Version (Versão) definida como latest (mais recente) para instalar a versão mais recente do atendente.
9. Escolha Executar.
10. Opcionalmente, nas áreas Targets and outputs (Destinos e saídas), selecione o botão ao lado do nome da instância e escolha View output (Visualizar saída). O Systems Manager deve exibir que o atendente foi instalado corretamente.

Criar e modificar o arquivo de configuração do atendente

Depois de baixar o atendente do CloudWatch, você deverá criar o arquivo de configuração antes de iniciar o atendente em qualquer servidor.

Se você salvar o arquivo de configuração do atendente no Systems Manager Parameter Store, use uma instância do EC2 para salvar no Parameter Store. Além disso, é necessário primeiro anexar a função do IAM `CloudWatchAgentAdminRole` a essa instância. Para obter mais informações sobre como anexar funções, consulte Anexar uma função do IAM a uma instância, no Manual do usuário do Amazon EC2 para instâncias do Windows.

Para obter mais informações sobre como criar o arquivo de configuração do atendente do CloudWatch, consulte Criar o arquivo de configuração do atendente do CloudWatch.

Instalação do atendente do CloudWatch em instâncias do EC2 usando sua configuração do atendente

Depois que você tiver uma configuração do atendente do CloudWatch salva no Parameter Store, poderá usá-la quando instalar o atendente em outros servidores.

Tópicos

- Anexar uma função do IAM à instância
- Baixar o pacote do atendente do CloudWatch em uma instância do Amazon EC2
- (Opcional) Modificar a configuração comum e o perfil nomeado para o atendente do CloudWatch
- Iniciar o atendente do CloudWatch

Anexar uma função do IAM à instância

Anexe a função do IAM CloudWatchAgentServerRole à instância do EC2 para poder executar o atendente do CloudWatch na instância. Essa função permite que o atendente do CloudWatch execute ações na instância. Você deveria ter criado esse perfil anteriormente. Para ter mais informações, consulte Criar funções e usuários do IAM para uso com o atendente do CloudWatch.

Para obter mais informações, consulte Anexar uma função do IAM a uma instância no Manual do usuário do Amazon EC2 para instâncias do Windows.

Baixar o pacote do atendente do CloudWatch em uma instância do Amazon EC2

É necessário instalar o atendente em cada servidor no qual você executará o atendente. O agente do CloudWatch está disponível como um pacote no Amazon Linux 2023 e Amazon Linux 2. Caso esteja usando esse sistema operacional, você pode instalar o pacote digitando o comando a seguir. Também é necessário garantir que a função do IAM anexada à instância tenha o atributo CloudWatchAgentServerPolicy anexado. Para mais informações, consulte Criar funções do IAM a serem usadas com o atendente do CloudWatch em instâncias do Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

Em todos os sistemas operacionais compatíveis, é possível baixar o pacote do atendente do CloudWatch usando o Systems Manager Run Command ou um link para download do Amazon S3. Para obter informações sobre o uso de um link para download do Amazon S3, consulte Baixar o pacote do atendente do CloudWatch.

> **Note**
>
> Quando você instala ou atualiza o atendente do CloudWatch, somente a opção Uninstall and reinstall (Desinstalar e reinstalar) é compatível. Não é possível usar a opção In-place update (Atualizações no local).

Baixar o atendente do CloudWatch em uma instância do Amazon EC2 usando o Systems Manager

Antes de usar o Systems Manager para instalar o atendente do CloudWatch, é necessário verificar se a instância está configurada corretamente para o Systems Manager.

Instalar ou atualizar o SSM Agent

Em uma instância do Amazon EC2, o atendente do CloudWatch exige que a instância esteja executando a versão 2.2.93.0 ou posterior. Antes de instalar o atendente do CloudWatch, atualize ou instale o SSM Agent na instância, caso ainda não tenha feito isso.

Para obter informações sobre como instalar ou atualizar o SSM Agent em uma instância que execute o Linux, consulte Instalar e configurar o SSM Agent em instâncias do Linux no Manual do usuário do AWS Systems Manager.

Para obter informações sobre como instalar ou atualizar o SSM Agent em uma instância que executa o Windows Server, consulte Instalar e configurar o SSM Agent em instâncias do Windows no Manual do usuário do AWS Systems Manager.

(Opcional) Verificar os pré-requisitos do Systems Manager

Antes de usar o Run Command do Systems Manager para instalar e configurar a integração com o atendente do CloudWatch, verifique se as instâncias atendem aos requisitos mínimos do Systems Manager. Para obter mais informações, consulte Configurar o AWS Systems Manager no Manual do usuário do AWS Systems Manager.

Verificar o acesso à Internet

Suas instâncias do Amazon EC2 devem ter acesso à Internet de saída a fim de enviar dados ao CloudWatch ou ao CloudWatch Logs. Para obter mais informações sobre como configurar o acesso à Internet, consulte Gateways da Internet no Manual do usuário da Amazon VPC.

Baixar o pacote do atendente do CloudWatch

O Systems Manager Run Command permite gerenciar a configuração de suas instâncias. Você especifica um documento do Systems Manager, especifica parâmetros e executa o comando em uma ou mais instâncias. O SSM Agent na instância processa o comando e configura a instância conforme especificado.

Para baixar o atendente do CloudWatch usando o Run Command

1. Abra o console do Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Executar comando.

   - ou -

   Se a página inicial do AWS Systems Manager for exibida, role para baixo e escolha Explore Run Command.
3. Selecione Run command.
4. Na lista Documento do comando, escolha AWS-ConfigureAWSPackage.
5. Na área Targets (Destinos), selecione a instância na qual o atendente do CloudWatch será instalado. Se você não visualizar uma instância específica, ela pode não estar configurada para o Run Command. Para obter mais informações, consulte Configurar o AWS Systems Manager para ambientes híbridos no Manual do usuário do AWS Systems Manager.

6. Na lista Ação, escolha Instalar.
7. Na caixa Name (Nome), digite *AmazonCloudWatchAgent*.
8. Deixe a Version (Versão) definida como latest (mais recente) para instalar a versão mais recente do atendente.
9. Escolha Executar.
10. Opcionalmente, nas áreas Targets and outputs (Destinos e saídas), selecione o botão ao lado do nome da instância e escolha View output (Visualizar saída). O Systems Manager deve exibir que o atendente foi instalado corretamente.

(Opcional) Modificar a configuração comum e o perfil nomeado para o atendente do CloudWatch

O atendente do CloudWatch inclui um arquivo de configuração chamado `common-config.toml`. Opcionalmente, é possível usar esse arquivo para especificar as informações de proxy e região.

Em um servidor que executa o Linux, esse arquivo encontra-se no diretório `/opt/aws/amazon-cloudwatch-agent/etc`. Em um servidor que executa o Windows Server, esse arquivo encontra-se no diretório `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

O `common-config.toml` padrão é conforme segue:

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
```

```
#    no_proxy = "{domain}"
```

Todas as linhas são comentadas inicialmente. Para definir o perfil de credencial ou as configurações de proxy, remova o # da linha e especifique um valor. É possível editar esse arquivo manualmente ou usando o Run Command `RunShellScript` no Systems Manager:

- `shared_credential_profile`: para servidores on-premises, essa linha especifica o perfil de credenciais do usuário do IAM a ser usado para enviar dados ao CloudWatch. Se você mantiver esta linha comentada, `AmazonCloudWatchAgent` será usado.

  Em uma instância do EC2, você poderá usar essa linha para que o atendente do CloudWatch envie dados dessa instância para o CloudWatch em outra região da AWS. Para fazer isso, especifique um perfil nomeado que inclua um campo `region` especificando o nome da região para a qual enviar.

  Se você especificar um `shared_credential_profile`, também deverá remover o símbolo # do início da linha `[credentials]`.
- `shared_credential_file`: para que o atendente procure credenciais em um arquivo localizado em um caminho diferente do padrão, especifique esse caminho completo e o nome do arquivo aqui. O caminho padrão é `/root/.aws` no Linux e `C:\\Users\\Administrator\\.aws` no Windows Server.

  O primeiro exemplo a seguir mostra a sintaxe de uma linha `shared_credential_file` válida para servidores Linux, e o segundo exemplo é válido para o Windows Server. No Windows Server, você deve fazer o escape dos caracteres \.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Se você especificar um `shared_credential_file`, também deverá remover o símbolo # do início da linha `[credentials]`.
- Configurações do proxy: se seus servidores usarem os proxies HTTP ou HTTPS para entrar em contato com os produtos da AWS, especifique esses proxies nos campos `http_proxy` e `https_proxy`. Se houver URLs que devem ser excluídas do proxy, as especifique no campo `no_proxy`, separadas por vírgulas.

Iniciar o atendente do CloudWatch

Você também pode iniciar o atendente usando o Systems Manager Run Command ou a linha de comando.

Iniciar o CloudWatch usando o Systems Manager Run Command

Siga estas etapas para iniciar o atendente usando o Systems Manager Run Command.

Para iniciar o atendente do CloudWatch usando o Run Command

1. Abra o console do Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Executar comando.

   - ou -

   Se a página inicial do AWS Systems Manager for exibida, role para baixo e escolha Explore Run Command.
3. Selecione Run command.
4. Na lista Documento do comando, escolha AmazonCloudWatch-ManageAgent.
5. Na área Targets (Destinos), escolha a instância onde você instalou o atendente do CloudWatch.
6. Na lista Ação, escolha configurar.
7. Na lista Origem de configuração opcional, escolha ssm.
8. Na caixa Local de configuração opcional, insira o nome do arquivo de configuração do agente que você criou e salvou no Systems Manager Parameter Store, conforme explicado em Criar o arquivo de configuração do atendente do CloudWatch.
9. Na lista Reinicialização opcional, escolha sim para iniciar o atendente após ter concluído essas etapas.
10. Escolha Executar.
11. Opcionalmente, nas áreas Targets and outputs (Destinos e saídas), selecione o botão ao lado do nome da instância e escolha View output (Visualizar saída). O Systems Manager deve exibir que o atendente foi iniciado corretamente.

Iniciar o atendente do CloudWatch em uma instância do Amazon EC2 usando a linha de comando

Siga estas etapas para usar a linha de comando para instalar o atendente do CloudWatch em uma instância do Amazon EC2.

Como usar a linha de comando para iniciar o atendente do CloudWatch em uma instância do Amazon EC2

- Nesse comando, `-a fetch-config` faz com que o atendente carregue a última versão do arquivo de configuração do atendente do CloudWatch e o `-s` inicia o atendente.

  Linux e macOS: se você salvou o arquivo de configuração no Systems Manager Parameter Store, insira o seguinte:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m ec2 -s -c ssm:configuration-parameter-store-name
  ```

  Linux e macOS: se você salvou o arquivo de configuração no computador local, insira o comando a seguir. Substitua *`configuration-file-path`* pelo caminho para o arquivo de configuração do atendente. Este arquivo é chamado `config.json`, se você o criou com o assistente, e pode ser chamado `amazon-cloudwatch-agent.json`, se você o criou manualmente.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m ec2 -s -c file:configuration-file-path
  ```

  Windows Server: se você salvou o arquivo de configuração do atendente no Systems Manager Parameter Store, insira o seguinte no console do PowerShell:

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
  a fetch-config -m ec2 -s -c ssm:configuration-parameter-store-name
  ```

  Windows Server: se você salvou o arquivo de configuração do atendente no computador local, digite o seguinte no console do PowerShell:

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
   -a fetch-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
  \config.json"
  ```

Instalar o atendente do CloudWatch em servidores no on-premises

Se você baixou o atendente do CloudWatch em um computador e criou o arquivo de configuração do atendente desejado, use esse arquivo para instalar o atendente em outros servidores on-premises.

Baixar o atendente do CloudWatch em um servidor on-premises

É possível baixar o pacote do atendente do CloudWatch usando o Systems Manager Run Command ou um link para download do Amazon S3. Para obter informações sobre o uso de um link para download do Amazon S3, consulte Baixar o pacote do atendente do CloudWatch.

Baixar usando o Systems Manager

Para usar o Systems Manager Run Command, registre seu servidor on-premises no Amazon EC2 Systems Manager. Para obter mais informações, consulte Configurar o Systems Manager em ambientes híbridos no Manual do usuário do AWS Systems Manager.

Se você já registrou o servidor, atualize o SSM Agent para a versão mais recente.

Para obter informações sobre a como atualizar o SSM Agent em um servidor que executa o Linux, consulte Instalar o SSM Agent para um ambiente híbrido (Linux) no Manual do usuário do AWS Systems Manager.

Para obter informações sobre como atualizar o SSM Agent em um servidor que executa o Windows Server, consulte Instalar o SSM Agent para um ambiente híbrido (Windows) no Manual do usuário do AWS Systems Manager.

Para usar o SSM Agent para baixar o pacote do atendente do CloudWatch em um servidor on-premises

1. Abra o console do Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Executar comando.

   - ou -

   Se a página inicial do AWS Systems Manager for exibida, role para baixo e escolha Explore Run Command.
3. Selecione Run command.
4. Na lista Documento do comando, selecione o botão ao lado de AWS-ConfigureAWSPackage.
5. Na área Targets (Destinos), selecione o servidor no qual o atendente do CloudWatch será instalado. Se você não visualizar um servidor específico, ele pode não estar configurado para o Run Command. Para obter mais informações, consulte Configurar o AWS Systems Manager para ambientes híbridos no Manual do usuário do AWS Systems Manager.
6. Na lista Ação, escolha Instalar.

7. Na caixa Name (Nome), digite *AmazonCloudWatchAgent*.
8. Deixe Version (Versão) em branco para instalar a versão mais recente do atendente.
9. Escolha Executar.

   O pacote do atendente é obtido por download, e as próximas etapas são configurá-lo e iniciá-lo.

(Instalar um servidor on-premises) Especificar credenciais do IAM e região da AWS

Para permitir que o CloudWatch envie dados de um servidor on-premises, é necessário especificar a chave de acesso e a chave secreta do usuário do IAM que você criou anteriormente. Para obter mais informações sobre como criar esse usuário, consulte Criar funções e usuários do IAM para uso com o atendente do CloudWatch.

Você também deve especificar a região da AWS para a qual enviar as métricas usando o campo `region`.

Veja a seguir um exemplo desse arquivo.

```
[AmazonCloudWatchAgent]
aws_access_key_id=my_access_key
aws_secret_access_key=my_secret_key
region = us-west-1
```

Para *my_access_key* e *my_secret_key*, use as chaves do usuário do IAM que não têm permissões para gravar no Systems Manager Parameter Store. Para obter mais informações sobre os usuários do IAM necessários para o atendente do CloudWatch, consulte Criar usuários do IAM para usar com o atendente do CloudWatch em servidores on-premises.

Se você chamar esse perfil de `AmazonCloudWatchAgent`, não precisará fazer mais nada. Você também pode fornecer a ela outro nome e especificar esse nome como o valor para `shared_credential_profile` no arquivo `common-config.toml`, o que é explicado na seção a seguir.

Abaixo, há um exemplo de uso do comando aws configure para criar um perfil nomeado para o atendente do CloudWatch. Esse exemplo pressupõe que você esteja usando o nome de perfil padrão do `AmazonCloudWatchAgent`.

Para criar o perfil AmazonCloudWatchAgent para o atendente do CloudWatch

1. Caso ainda não tenha feito isso, instale a AWS Command Line Interface. Para obter mais informações, consulte Instalar a AWS CLI.
2. Em servidores Linux, digite este comando e siga os avisos:

   ```
   sudo aws configure --profile AmazonCloudWatchAgent
   ```

   No Windows Server, abra o PowerShell como administrador, digite o seguinte comando e siga as instruções.

   ```
   aws configure --profile AmazonCloudWatchAgent
   ```

(Opcional) Modificar a configuração comum e o perfil nomeado para o atendente do CloudWatch

O atendente do CloudWatch inclui um arquivo de configuração chamado `common-config.toml`. Você pode, opcionalmente, usar esse arquivo para especificar as informações de proxy e região.

Em um servidor que executa o Linux, esse arquivo encontra-se no diretório `/opt/aws/amazon-cloudwatch-agent/etc`. Em um servidor que executa o Windows Server, esse arquivo encontra-se no diretório `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

O `common-config.toml` padrão é conforme segue:

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
```

```
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Todas as linhas são comentadas inicialmente. Para definir o perfil de credencial ou as configurações de proxy, remova o # da linha e especifique um valor. É possível editar esse arquivo manualmente ou usando o Run Command `RunShellScript` no Systems Manager:

- `shared_credential_profile`: para servidores on-premises, essa linha especifica o perfil de credenciais do usuário do IAM a ser usado para enviar dados ao CloudWatch. Se você mantiver esta linha comentada, `AmazonCloudWatchAgent` será usado. Para obter mais informações sobre a criação desse perfil, consulte (Instalar um servidor on-premises) Especificar credenciais do IAM e região da AWS.

  Em uma instância do EC2, você poderá usar essa linha para que o atendente do CloudWatch envie dados dessa instância para o CloudWatch em outra região da AWS. Para fazer isso, especifique um perfil nomeado que inclua um campo `region` especificando o nome da região para a qual enviar.

  Se você especificar um `shared_credential_profile`, também deverá remover o símbolo # do início da linha `[credentials]`.
- `shared_credential_file`: para que o atendente procure credenciais em um arquivo localizado em um caminho diferente do padrão, especifique esse caminho completo e o nome do arquivo aqui. O caminho padrão é `/root/.aws` no Linux e `C:\\Users\\Administrator\\.aws` no Windows Server.

  O primeiro exemplo a seguir mostra a sintaxe de uma linha `shared_credential_file` válida para servidores Linux, e o segundo exemplo é válido para o Windows Server. No Windows Server, você deve fazer o escape dos caracteres \.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Se você especificar um `shared_credential_file`, também deverá remover o símbolo # do início da linha `[credentials]`.

- Configurações do proxy: se seus servidores usarem os proxies HTTP ou HTTPS para entrar em contato com os produtos da AWS, especifique esses proxies nos campos `http_proxy` e `https_proxy`. Se houver URLs que devem ser excluídas do proxy, as especifique no campo `no_proxy`, separadas por vírgulas.

Iniciar o atendente do CloudWatch

Você também pode iniciar o atendente do CloudWatch usando o Systems Manager Run Command ou a linha de comando.

Para usar o SSM Agent para iniciar o atendente do CloudWatch em um servidor on-premises

1. Abra o console do Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Executar comando.

   - ou -

   Se a página inicial do AWS Systems Manager for exibida, role para baixo e escolha Explore Run Command.
3. Selecione Run command.
4. Na lista Documento do comando, selecione o botão ao lado de AmazonCloudWatch-ManageAgent.
5. Na área Destinos, selecione a instância onde você instalou o atendente.
6. Na lista Ação, escolha configurar.
7. Na lista Modo, escolha No local.
8. Na caixa Optional Configuration Location (Local de configuração opcional), insira o nome do arquivo de configuração do atendente que você criou com o assistente e armazenou no Parameter Store.
9. Escolha Executar.

   O atendente começa com a configuração especificada no arquivo de configuração.

Para usar a linha de comando para iniciar o atendente do CloudWatch em um servidor on-premises

- Nesse comando, `-a fetch-config` faz com que o atendente carregue a última versão do arquivo de configuração do atendente do CloudWatch e o `-s` inicia o atendente.

Linux: se você salvou o arquivo de configuração no Systems Manager Parameter Store, insira o seguinte:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m onPremise -s -c ssm:configuration-parameter-store-name
```

Linux: se você salvou o arquivo de configuração no computador local, insira o comando a seguir. Substitua *configuration-file-path* pelo caminho para o arquivo de configuração do atendente. Este arquivo é chamado `config.json`, se você o criou com o assistente, e pode ser chamado `amazon-cloudwatch-agent.json`, se você o criou manualmente.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m onPremise -s -c file:configuration-file-path
```

Windows Server: se você salvou o arquivo de configuração do atendente no Systems Manager Parameter Store, insira o seguinte no console do PowerShell:

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m onPremise -s -c ssm:configuration-parameter-store-name
```

Windows Server: se você salvou o arquivo de configuração do atendente no computador local, digite o comando a seguir no console do PowerShell. Substitua *configuration-file-path* pelo caminho para o arquivo de configuração do atendente. Este arquivo é chamado `config.json`, se você o criou com o assistente, e pode ser chamado `amazon-cloudwatch-agent.json`, se você o criou manualmente.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m onPremise -s -c file:configuration-file-path
```

Instalar o atendente do CloudWatch em novas instâncias usando o AWS CloudFormation

A Amazon carregou diversos modelos do AWS CloudFormation no GitHub para ajudar a instalar e atualizar o atendente do CloudWatch em novas instâncias do Amazon EC2. Para obter mais informações sobre como usar o AWS CloudFormation, consulte O que é o AWS CloudFormation?.

O local do modelo é Implantar o atendente do Amazon CloudWatch em instâncias do EC2 usando o AWS CloudFormation. Esse local inclui os diretórios `inline` e `ssm`. Cada um desses diretórios contém modelos de ambas as instâncias do Linux e do Windows.

- Os modelos no diretório `inline` têm a configuração do atendente do CloudWatch incorporada ao modelo do AWS CloudFormation. Por padrão, os modelos do Linux coletam as métricas `mem_used_percent` e `swap_used_percent`, e os modelos do Windows coletam `Memory % Committed Bytes In Use` e `Paging File % Usage`.

  Para modificar esses modelos para coletar métricas diferentes, modifique a seção a seguir do modelo. O exemplo a seguir é do modelo para servidores do Linux. Siga o formato e a sintaxe do arquivo de configuração do atendente para fazer essas alterações. Para ter mais informações, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

```
{
   "metrics":{
      "append_dimensions":{
         "AutoScalingGroupName":"${!aws:AutoScalingGroupName}",
         "ImageId":"${!aws:ImageId}",
         "InstanceId":"${!aws:InstanceId}",
         "InstanceType":"${!aws:InstanceType}"
      },
      "metrics_collected":{
         "mem":{
            "measurement":[
               "mem_used_percent"
            ]
         },
         "swap":{
            "measurement":[
               "swap_used_percent"
            ]
         }
      }
   }
}
```

> **Note**
>
> Nos modelos em linha, todas as variáveis de espaço reservado devem ter um ponto de exclamação (!) antes delas como um caractere de escape. Veja isso no modelo de exemplo. Se você adicionar outras variáveis de espaço reservado, adicione um ponto de exclamação antes do nome.

- Os modelos no diretório `ssm` carregam um arquivo de configuração do atendente do Parameter Store. Para usar esses modelos, é necessário primeiro criar um arquivo de configuração e carregá-lo no Parameter Store. Em seguida, dê o nome Parameter Store do arquivo no modelo. Crie o arquivo de configuração manualmente ou usando o assistente. Para ter mais informações, consulte Criar o arquivo de configuração do atendente do CloudWatch.

Use ambos os tipos de modelos para instalar o atendente do CloudWatch e atualizar a configuração do atendente.

Tutorial: instalar e configurar o atendente do CloudWatch usando um modelo em linha do AWS CloudFormation

Esse tutorial mostra como usar o AWS CloudFormation para instalar o atendente do CloudWatch em uma nova instância do Amazon EC2. Esse tutorial faz a instalação em uma nova instância que esteja executando o Amazon Linux 2 usando os modelos em linha, que não exigem o uso do arquivo de configuração JSON nem o Parameter Store. O modelo em linha inclui a configuração do atendente no modelo. Neste tutorial, você usa a configuração do atendente padrão contida no modelo.

Após o procedimento para instalar o atendente, o tutorial continua com como atualizar o atendente.

Para usar o AWS CloudFormation para instalar o atendente do CloudWatch em uma nova instância

1. Faça download do modelo do GitHub. Neste tutorial, baixe o modelo em linha do Amazon Linux 2 da seguinte forma:

   ```
   curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/
   master/aws/solutions/AmazonCloudWatchAgent/inline/amazon_linux.template
   ```

2. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
3. Selecione Criar pilha.

4. Em Choose a template (Escolher um modelo), selecione Upload a template to Amazon S3 (Fazer upload de um modelo no Amazon S3), escolha o modelo obtido por download e Next (Avançar).
5. Na página Specify Details (Especificar detalhes), preencha os parâmetros a seguir e escolha Next (Próximo):

   - Stack name (Nome da pilha): escolha um nome de pilha para a pilha do AWS CloudFormation.
   - IAMRole: escolha um perfil do IAM que tenha permissões para gravar métricas e logs e rastreamentos do CloudWatch. Para ter mais informações, consulte Criar funções do IAM a serem usadas com o atendente do CloudWatch em instâncias do Amazon EC2.
   - InstanceAMI: escolha uma AMI que seja válida na região onde você pretende iniciar a pilha.
   - InstanceType: escolha um tipo de instância válido.
   - KeyName: para habilitar o acesso SSH à nova instância, escolha um par de chaves do Amazon EC2 existente. Se ainda não tiver um par de chaves do Amazon EC2, você poderá criar um no AWS Management Console. Para obter mais informações, consulte Pares de chaves do Amazon EC2 no Manual do usuário do Amazon EC2 para instâncias do Linux.
   - SSHLocation: especifica o intervalo de endereços IP que podem ser usados para se conectar à instância usando SSH. O padrão permite o acesso de qualquer endereço IP.
6. Na página Options (Opções), você pode optar por marcar seus recursos de pilha. Escolha Próximo.
7. Na página Review (Revisão), revise as informações, confirme se a pilha pode criar recursos do IAM e selecione Create (Criar).

   Se atualizar o console, você verá que a nova pilha tem o status `CREATE_IN_PROGRESS`.
8. Quando a instância é criada, é possível vê-la no console do Amazon EC2. Também é possível se conectar ao host e verificar o progresso.

   Use o seguinte comando para confirmar se o atendente está instalado:

   ```
   rpm -qa amazon-cloudwatch-agent
   ```

   Use o seguinte comando para confirmar se o atendente está em execução:

   ```
   ps aux | grep amazon-cloudwatch-agent
   ```

O próximo procedimento demonstra como usar o AWS CloudFormation para atualizar o atendente do CloudWatch usando um modelo em linha. O modelo em linha padrão coleta a métrica `mem_used_percent`. Neste tutorial, altere a configuração do atendente para interromper a coleta dessa métrica.

Para usar o AWS CloudFormation para atualizar o atendente do CloudWatch

1. No modelo obtido por download no procedimento anterior, remova as seguintes linhas e salve o modelo:

   ```
   "mem": {

        "measurement": [
            "mem_used_percent"
          ]
    },
   ```

2. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
3. No painel do AWS CloudFormation, selecione a pilha criada e escolha Update Stack (Atualizar pilha).
4. Em Select Template (Selecionar modelo), selecione Upload a template to Amazon S3 (Fazer upload de um modelo para o Amazon S3), escolha o modelo modificado e Next (Avançar).
5. Na página Options (Opções), escolha Next (Próximo) e, depois, Next (Próximo).
6. Na página Review (Revisar), revise as informações e escolha Update (Atualizar).

   Depois de algum tempo, você verá `UPDATE_COMPLETE`.

Tutorial: instalar o atendente do CloudWatch usando o AWS CloudFormation e Parameter Store

Esse tutorial mostra como usar o AWS CloudFormation para instalar o atendente do CloudWatch em uma nova instância do Amazon EC2. Este tutorial instala em uma nova instância na qual o Amazon Linux 2 esteja em execução usando um arquivo de configuração do atendente que você criou e salvo no Parameter Store.

Após o procedimento para instalar o atendente, o tutorial continua com como atualizar o atendente.

Para usar o AWS CloudFormation para instalar o atendente do CloudWatch em uma nova instância usando uma configuração do Parameter Store

1. Se ainda não o fez, baixe o pacote do atendente do CloudWatch em um de seus computadores para poder criar o arquivo de configuração do atendente. Para obter mais informações mais detalhadas e baixar o atendente com o Parameter Store, consulte Baixar e configurar o atendente do CloudWatch. Para obter mais informações sobre o download do pacote usando a linha de comando, consulte Baixar e configurar o atendente do CloudWatch usando a linha de comando.
2. Crie o arquivo de configuração do atendente e salve-o no Parameter Store. Para ter mais informações, consulte Criar o arquivo de configuração do atendente do CloudWatch.
3. Faça download do modelo do GitHub da seguinte forma:

   ```
   curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/
   master/aws/solutions/AmazonCloudWatchAgent/ssm/amazon_linux.template
   ```
4. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
5. Selecione Criar pilha.
6. Em Choose a template (Escolher um modelo), selecione Upload a template to Amazon S3 (Fazer upload de um modelo no Amazon S3), escolha o modelo obtido por download e Next (Avançar).
7. Na página Specify Details (Especificar detalhes), preencha os seguintes parâmetros conforme necessário e, depois, escolha Next (Próximo):

   - Stack name (Nome da pilha): escolha um nome de pilha para a pilha do AWS CloudFormation.
   - IAMRole: escolha um perfil do IAM que tenha permissões para gravar métricas e logs e rastreamentos do CloudWatch. Para ter mais informações, consulte Criar funções do IAM a serem usadas com o atendente do CloudWatch em instâncias do Amazon EC2.
   - InstanceAMI: escolha uma AMI que seja válida na região onde você pretende iniciar a pilha.
   - InstanceType: escolha um tipo de instância válido.
   - KeyName: para habilitar o acesso SSH à nova instância, escolha um par de chaves do Amazon EC2 existente. Se ainda não tiver um par de chaves do Amazon EC2, você poderá criar um no AWS Management Console. Para obter mais informações, consulte Pares de chaves do Amazon EC2 no Manual do usuário do Amazon EC2 para instâncias do Linux.
   - SSHLocation: especifica o intervalo de endereços IP que podem ser usados para se conectar à instância usando SSH. O padrão permite o acesso de qualquer endereço IP.

- SSMKey: especifica o arquivo de configuração do atendente que você criou e salvou no Parameter Store.

8. Na página Options (Opções), você pode optar por marcar seus recursos de pilha. Escolha Próximo.
9. Na página Review (Revisão), revise as informações, confirme se a pilha pode criar recursos do IAM e selecione Create (Criar).

   Se atualizar o console, você verá que a nova pilha tem o status `CREATE_IN_PROGRESS`.
10. Quando a instância é criada, é possível vê-la no console do Amazon EC2. Também é possível se conectar ao host e verificar o progresso.

    Use o seguinte comando para confirmar se o atendente está instalado:

    ```
    rpm -qa amazon-cloudwatch-agent
    ```

    Use o seguinte comando para confirmar se o atendente está em execução:

    ```
    ps aux | grep amazon-cloudwatch-agent
    ```

O procedimento a seguir demonstra como usar o AWS CloudFormation para atualizar o atendente do CloudWatch com uma configuração do atendente salva no Parameter Store.

Para usar o AWS CloudFormation para atualizar o atendente do CloudWatch com uma configuração no Parameter Store

1. Altere o arquivo de configuração do atendente armazenado no Parameter Store para a nova configuração desejada.
2. No modelo do AWS CloudFormation obtido por download no tópico the section called “Tutorial: instalar o atendente do CloudWatch usando o AWS CloudFormation e Parameter Store”, altere o número da versão. Por exemplo, você poderia alterar `VERSION=1.0` para `VERSION=2.0`.
3. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
4. No painel do AWS CloudFormation, selecione a pilha criada e escolha Update Stack (Atualizar pilha).
5. Em Select Template (Selecionar modelo), selecione Upload a template to Amazon S3 (Fazer upload de um modelo para o Amazon S3), selecione o modelo recém-modificado e Next (Avançar).

6. Na página Options (Opções), escolha Next (Próximo) e, depois, Next (Próximo).
7. Na página Review (Revisar), revise as informações e escolha Update (Atualizar).

   Depois de algum tempo, você verá `UPDATE_COMPLETE`.

Solução de problemas de instalação do atendente do CloudWatch com o AWS CloudFormation

Esta seção ajuda a solucionar problemas na instalação e na atualização do atendente do CloudWatch usando o AWS CloudFormation.

Detectar quando uma atualização falha

Se você usar o AWS CloudFormation para atualizar a configuração do atendente do CloudWatch e usar uma configuração inválida, o atendente deixará de enviar métricas para o CloudWatch. Uma maneira rápida de verificar se uma atualização da configuração do atendente foi bem-sucedida é procurar no arquivo `cfn-init-cmd.log`. Em um servidor do Linux, o arquivo está localizado em `/var/log/cfn-init-cmd.log`. Em uma instância do Windows, o arquivo está localizado em `C:\cfn\log\cfn-init-cmd.log`.

As métricas não foram encontradas

Se você não vir métricas que deveria ver depois de instalar ou atualizar o atendente, confirme se o atendente está configurado para coletar essa métrica. Para fazer isso, verifique o arquivo `amazon-cloudwatch-agent.json` para garantir que a métrica esteja listada e que você esteja procurando no namespace de métrica correto. Para ter mais informações, consulte Arquivos e locais do atendente do CloudWatch.

Preferência de credenciais do agente do CloudWatch

Esta seção descreve a cadeia de provedores de credenciais que o agente do CloudWatch usa para obter credenciais ao se comunicar com outros serviços e APIs da AWS. A ordenação é apresentada a seguir. As preferências listadas nos números dois a cinco da lista a seguir seguem a mesma ordem de preferência definida no AWS SDK. Para obter mais informações, consulte Specifying Credentials na documentação do SDK.

1. Arquivos de configuração e de credenciais compartilhados conforme definido no arquivo `common-config.toml` do agente CloudWatch. Para ter mais informações, consulte (Opcional) Modificar a configuração comum para informações de proxy ou região.

2. Variáveis de ambiente do AWS SDK.

> **Important**
>
> No Linux, se você executar o agente do CloudWatch usando o script `amazon-cloudwatch-agent-ctl`, o script iniciará o agente como um serviço `systemd`. Nesse caso, variáveis de ambiente como `HOME`, `AWS_ACCESS_KEY_ID` e `AWS_SECRET_ACCESS_KEY` não podem ser acessadas pelo agente.

3. Arquivos de configuração e de credenciais compartilhados encontrados em `$HOME/%USERPROFILE%`.

> **Note**
>
> O agente do CloudWatch procura `.aws/credentials` em `$HOME` para os sistemas Linux e MacOS e procura em `%USERPROFILE%` para o sistema Windows. Ao contrário do AWS SDK, o agente do CloudWatch não tem métodos alternativos para determinar o diretório inicial se as variáveis de ambiente estiverem inacessíveis. Essa diferença de comportamento visa manter a compatibilidade retroativa com implementações anteriores do AWS SDK.
>
> Além disso, ao contrário das credenciais compartilhadas encontradas em `common-config.toml`, se as credenciais compartilhadas derivadas do AWS SDK expirarem e forem alternadas, as credenciais renovadas não serão obtidas automaticamente pelo agente do CloudWatch e exigirão uma reinicialização do agente para fazer isso.

4. Um perfil do AWS Identity and Access Management para tarefas, se houver uma aplicação que usa uma definição de tarefa do Amazon Elastic Container Service ou uma operação de API RunTask.
5. Um perfil de instância anexado a uma instância do Amazon EC2.

Como prática recomendada, sugerimos especificar as credenciais na ordem apresentada a seguir ao usar o agente do CloudWatch.

1. Use perfis do IAM para tarefas se sua aplicação usar uma definição de tarefa do Amazon Elastic Container Service ou uma operação de API RunTask.
2. Use perfis do IAM se sua aplicação for executada em uma instância do Amazon EC2.

3. Use o arquivo `common-config.toml` do agente do CloudWatch para especificar o arquivo de credenciais. Esse arquivo de credenciais é o mesmo usado por outros AWS SDKs e pela AWS CLI. Se já estiver usando um arquivo de credenciais compartilhadas, também poderá usá-lo para essa finalidade. Se você fornecê-lo ao usar o arquivo `common-config.toml` do agente do CloudWatch, garantirá que o agente consumirá credenciais rotacionadas quando elas expirarem e forem substituídas sem a necessidade de reiniciar o agente.
4. Use variáveis de ambiente. Definir variáveis de ambiente é útil se você estiver realizando um trabalho de desenvolvimento em um computador que não seja uma instância do Amazon EC2.

> **Note**
>
> Se você enviar telemetria para uma conta diferente, conforme explicado em Envio de métricas, logs e rastreamentos a uma conta diferente, o agente do CloudWatch usará a cadeia de provedores de credenciais descrita nesta seção para obter o conjunto inicial de credenciais. Em seguida, ele usa essas credenciais ao assumir o perfil do IAM especificado por `role_arn` no arquivo de configuração do agente do CloudWatch.

Verificar a assinatura do pacote do atendente do CloudWatch

Os arquivos de assinatura GPG estão incluídos para pacotes do atendente do CloudWatch em servidores Linux. Use a chave pública para verificar se o arquivo de download do atendente é original e não modificado.

Para o Windows Server, você pode usar o MSI para verificar a assinatura.

Para computadores macOS, a assinatura está incluída no pacote para baixar o atendente.

Para encontrar o arquivo signature correto, consulte a seguinte tabela. Para cada arquitetura e sistema operacional há um link geral, bem como links para cada região. Por exemplo, para o Amazon Linux 2023 e Amazon Linux 2 e a arquitetura x86-64, três dos links válidos são:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

> **Note**
>
> Para baixar o agente do CloudWatch, sua conexão deve usar o TLS 1.2 ou posterior.

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
x86-64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/centos/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
		/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm	/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/debian/amd64/latest/amazon-cloudwatch-agent.deb.sig

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb.sig
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/oracle_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/darwin/amd64/latest/amazon-cloudwatch-agent.pkg.sig

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi.sig
ARM64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/amazon_linux/arm64/latest/amazon-cloudwatch-agent.rpm.sig
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/redhat/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Arquitetura	Plataforma	Link para fazer download	Link do arquivo de assinatura
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/ubuntu/arm64/latest/amazon-cloudwatch-agent.deb.sig
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig https://amazoncloudwatch-agent-*region*.s3.*region*.amazonaws.com/suse/arm64/latest/amazon-cloudwatch-agent.rpm.sig

Para verificar o pacote do atendente do CloudWatch em um servidor Linux

1. Faça download da chave pública.

```
shell$ wget https://amazoncloudwatch-agent.s3.amazonaws.com/assets/amazon-
cloudwatch-agent.gpg
```

2. Importe a chave pública em seu token de autenticação.

```
shell$  gpg --import amazon-cloudwatch-agent.gpg
gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
```

Anote o valor da chave, pois ele será necessário na próxima etapa. No exemplo anterior, o valor da chave é 3B789C72.

3. Verifique a impressão digital, executando o comando a seguir, substituindo *chave-valor* pelo valor da etapa anterior:

```
shell$  gpg --fingerprint key-value
pub   2048R/3B789C72 2017-11-14
      Key fingerprint = 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
uid                  Amazon CloudWatch Agent
```

A string de impressão digital deve ser igual a esta:

```
9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
```

Se a string de impressão digital não coincidir, não instale o atendente. Entre em contato com a Amazon Web Services.

Depois de verificar a impressão digital, você pode usá-la para verificar a assinatura do pacote do atendente do CloudWatch.

4. Faça download do arquivo signature do pacote usando o wget. Para determinar o arquivo de assinatura correto, consulte a tabela anterior.

```
wget Signature File Link
```

5. Para verificar a assinatura, execute gpg --verify.

```
shell$ gpg --verify signature-filename agent-download-filename
gpg: Signature made Wed 29 Nov 2017 03:00:59 PM PST using RSA key ID 3B789C72
gpg: Good signature from "Amazon CloudWatch Agent"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Se a saída inclui a frase `BAD signature`, verifique se você executou o procedimento corretamente. Se você continuar recebendo essa resposta, entre em contato com a Amazon Web Services e evite usar o arquivo baixado.

Observe o aviso sobre confiança. Uma chave somente será confiável se você ou alguém em quem você confia a tiver assinado. Isso não significa que a assinatura é inválida, apenas que você não verificou a chave pública.

Para verificar o pacote do atendente do CloudWatch em um servidor executando Windows Server

1. Faça download e instale o GnuPG para Windows https://gnupg.org/download/. Ao instalar, inclua a opção Extensão Shell (GpgEx).

   Você pode executar as etapas restantes no Windows PowerShell.

2. Faça download da chave pública.

```
PS> wget https://amazoncloudwatch-agent.s3.amazonaws.com/assets/amazon-cloudwatch-agent.gpg -OutFile amazon-cloudwatch-agent.gpg
```

3. Importe a chave pública em seu token de autenticação.

```
PS>  gpg --import amazon-cloudwatch-agent.gpg
gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
```

   Anote o valor da chave, pois ele será necessário na próxima etapa. No exemplo anterior, o valor da chave é `3B789C72`.

4. Verifique a impressão digital, executando o comando a seguir, substituindo *`chave-valor`* pelo valor da etapa anterior:

```
PS>  gpg --fingerprint key-value
pub   rsa2048 2017-11-14 [SC]
      9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
uid           [ unknown] Amazon CloudWatch Agent
```

   A string de impressão digital deve ser igual a esta:

```
9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
```

   Se a string de impressão digital não coincidir, não instale o atendente. Entre em contato com a Amazon Web Services.

Depois de verificar a impressão digital, você pode usá-la para verificar a assinatura do pacote do atendente do CloudWatch.

5. Faça download do arquivo signature do pacote usando o wget. Para determinar o arquivo Signature correto, consulte Links de download do atendente do CloudWatch.
6. Para verificar a assinatura, execute gpg --verify.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Se a saída inclui a frase BAD `signature`, verifique se você executou o procedimento corretamente. Se você continuar recebendo essa resposta, entre em contato com a Amazon Web Services e evite usar o arquivo baixado.

Observe o aviso sobre confiança. Uma chave somente será confiável se você ou alguém em quem você confia a tiver assinado. Isso não significa que a assinatura é inválida, apenas que você não verificou a chave pública.

Para verificar o pacote do atendente do CloudWatch em um computador macOS

- Há dois métodos para verificação de assinatura no macOS.
  - Verifique a impressão digital executando o comando a seguir.

    ```
    pkgutil --check-signature amazon-cloudwatch-agent.pkg
    ```

    Será exibido um resultado semelhante ao exibido a seguir.

    ```
    Package "amazon-cloudwatch-agent.pkg":
           Status: signed by a developer certificate issued by Apple for
     distribution
           Signed with a trusted timestamp on: 2020-10-02 18:13:24 +0000
           Certificate Chain:
           1. Developer ID Installer: AMZN Mobile LLC (94KV3E626L)
           Expires: 2024-10-18 22:31:30 +0000
    ```

```
        SHA256 Fingerprint:
        81 B4 6F AF 1C CA E1 E8 3C 6F FB 9E 52 5E 84 02 6E 7F 17 21 8E FB
        0C 40 79 13 66 8D 9F 1F 10 1C

------------------------------------------------------------------------
        2. Developer ID Certification Authority
        Expires: 2027-02-01 22:12:15 +0000
        SHA256 Fingerprint:
        7A FC 9D 01 A6 2F 03 A2 DE 96 37 93 6D 4A FE 68 09 0D 2D E1 8D 03
        F2 9C 88 CF B0 B1 BA 63 58 7F

------------------------------------------------------------------------
        3. Apple Root CA
        Expires: 2035-02-09 21:40:36 +0000
        SHA256 Fingerprint:
        B0 B1 73 0E CB C7 FF 45 05 14 2C 49 F1 29 5E 6E DA 6B CA ED 7E 2C
        68 C5 BE 91 B5 A1 10 01 F0 24
```

- Ou baixe e utilize o arquivo .sig. Para utilizar esse método, siga estas etapas.
  - Instale a aplicação GPG no host do macOS inserindo o comando a seguir.

    ```
    brew install GnuPG
    ```

- Baixe o arquivo de assinatura do pacote usando curl. Para determinar o arquivo Signature correto, consulte Links de download do atendente do CloudWatch.
- Para verificar a assinatura, execute gpg --verify.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Se a saída inclui a frase `BAD signature`, verifique se você executou o procedimento corretamente. Se você continuar recebendo essa resposta, entre em contato com a Amazon Web Services e evite usar o arquivo baixado.

Observe o aviso sobre confiança. Uma chave somente será confiável se você ou alguém em quem você confia a tiver assinado. Isso não significa que a assinatura é inválida, apenas que você não verificou a chave pública.

Criar o arquivo de configuração do atendente do CloudWatch

Antes de executar o agente do CloudWatch em qualquer servidor, crie um ou mais arquivos de configuração do atendente do CloudWatch.

O arquivo de configuração do agente é um arquivo JSON que especifica as métricas, logs e rastreamentos que o agente deverá coletar, inclusive métricas personalizadas. Você pode criá-lo usando o assistente ou criando-o do zero. Você também pode usar o assistente para criar inicialmente o arquivo de configuração e, depois, modificá-lo manualmente. Se você criar ou modificar o arquivo manualmente, o processo será mais complexo, mas você terá maior controle sobre as métricas coletadas e poderá especificar métricas não disponíveis pelo assistente.

Sempre que você alterar o atendente do arquivo de configuração do atendente, deverá reiniciar o atendente para que as alterações entrem em vigor. Para reiniciar o atendente, siga as instruções em Iniciar o atendente do CloudWatch.

Depois de criar um arquivo de configuração, você poderá salvá-lo manualmente como um arquivo JSON e, depois, usar esse arquivo ao instalar o atendente em seus servidores. Se prererir, você tabém pode armazená-lo no Systems Manager Parameter Store se pretende usar o Systems Manager ao instalar o atendente em servidores.

O agente do CloudWatch aceita o uso de vários arquivos de configuração. Para ter mais informações, consulte Vários arquivos de configuração do atendente CloudWatch.

Métricas, logs e rastreamentos coletados pelo agente do CloudWatch incorrem em cobranças. Para obter mais informações sobre a definição de preço, consulte Preços do Amazon CloudWatch.

Conteúdo

Criar o arquivo de configuração do atendente do CloudWatch com o assistente

O assistente de arquivo de configuração do agente, `amazon-cloudwatch-agent-config-wizard`, faz uma série de perguntas para ajudá-lo a configurar o agente do CloudWatch para suas necessidades.

Credenciais necessárias

O assistente poderá detectar automaticamente as credenciais e a região da AWS a serem usadas se você tiver as credenciais e os arquivos de configuração da AWS antes de iniciar o assistente. Para obter mais informações sobre esses arquivos, consulte Arquivos de configuração e credenciais no Manual do usuário do AWS Systems Manager.

No arquivo de credenciais da AWS, o assistente verifica a existência de credenciais padrão e também procura uma seção `AmazonCloudWatchAgent` como a seguinte:

```
[AmazonCloudWatchAgent]
aws_access_key_id = my_access_key
aws_secret_access_key = my_secret_key
```

O assistente exibe as credenciais padrão, as credenciais do `AmazonCloudWatchAgent` e uma opção `Others`. Você pode selecionar as credenciais a serem usadas. Se você escolher `Others`, insira as credenciais.

Para *`my_access_key`* e *`my_secret_key`*, use as chaves do usuário do IAM que têm permissões para gravar no Systems Manager Parameter Store. Para obter mais informações sobre os usuários do IAM necessários para o atendente do CloudWatch, consulte Criar usuários do IAM para usar com o atendente do CloudWatch em servidores on-premises.

No arquivo de configuração da AWS, especifique a região para a qual o atendente enviará métricas, se ela for diferente da seção `[default]`. O padrão é publicar as métricas na região na qual a instância do Amazon EC2 está localizada. Se as métricas precisarem ser publicadas em uma região diferente, especifique-a aqui. No exemplo a seguir, as métricas são publicadas na região `us-west-1`.

```
[AmazonCloudWatchAgent]
region = us-west-1
```

Executar o assistente de configuração do atendente do CloudWatch

Para criar o arquivo de configuração do atendente do CloudWatch

1. Inicie o assistente de configuração do atendente do CloudWatch inserindo o seguinte:

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
   ```

   Em um servidor com o Windows Server, execute os comandos a seguir para iniciar o agente:

   ```
   cd "C:\Program Files\Amazon\AmazonCloudWatchAgent"
   ```

   ```
   .\amazon-cloudwatch-agent-config-wizard.exe
   ```

2. Responda às perguntas para personalizar o arquivo de configuração para o servidor.
3. Se você estiver armazenando o arquivo de configuração localmente, o arquivo de configuração `config.json` será armazenado em `/opt/aws/amazon-cloudwatch-agent/bin/` em servidores Linux e armazenado em `C:\Program Files\Amazon\AmazonCloudWatchAgent` no Windows Server. Depois, você pode copiar esse arquivo em outros servidores em que você deseja instalar o atendente.

   Se você for usar o Systems Manager para instalar e configurar o atendente, responda Yes (Sim) quando solicitado para armazenar o arquivo no Systems Manager Parameter Store. Você também pode optar por armazenar o arquivo no Parameter Store mesmo que não esteja usando o SSM Agent para instalar o atendente do CloudWatch. Para poder armazenar o arquivo no Parameter Store, você deve usar uma função do IAM com permissões suficientes. Para ter mais informações, consulte Criar funções e usuários do IAM para uso com o atendente do CloudWatch.

Conjuntos de métricas predefinidas do atendente CloudWatch

O assistente está configurado com conjuntos de métricas predefinidos, com diferentes níveis de detalhes. Esses conjuntos de métricas são mostrados nas tabelas a seguir. Para ter mais informações sobre essas métricas, consulte Métricas coletadas pelo atendente do CloudWatch.

> **Note**
>
> O Parameter Store oferece suporte a parâmetros nos níveis padrão e avançado. Esses níveis de parâmetro não estão relacionados aos níveis Básico, Padrão e Avançado de detalhes da métrica descritos nessas tabelas.

Instâncias do Amazon EC2 que executam o Linux

Nível de detalhes	Métricas incluídas
Basic	Mem: mem_used_percent Disk: disk_used_percent As métricas de `disk`, como, `disk_used_percent` , têm uma dimensão para `Partition` , o que significa que o número de métricas personalizadas geradas depende do número de partições associadas à instância. O número de partições de disco que você tem depende da AMI usada e do número de volumes do Amazon EBS que estão sendo anexados ao servidor.
Padrão	CPU: `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_user` , `cpu_usage_system` Disco: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` Mem: `mem_used_percent` Swap: `swap_used_percent`
Advanced (Avançado)	CPU: `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_user` , `cpu_usage_system` Disco: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads`

Nível de detalhes	Métricas incluídas
	Mem: `mem_used_percent` Netstat: `netstat_tcp_established` , `netstat_tcp_time_wait` Swap: `swap_used_percent`

Servidores on-premises que executam o Linux

Nível de detalhes	Métricas incluídas
Basic	Disco: `disk_used_percent` Diskio: `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem: `mem_used_percent` Rede: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Swap: `swap_used_percent`
Padrão	CPU: `cpu_usage_idle` , `cpu_usage_iowait` Disco: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem: `mem_used_percent` Rede: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Swap: `swap_used_percent`

Nível de detalhes	Métricas incluídas
Advanced (Avançado)	CPU: `cpu_usage_guest` , `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_steal` , `cpu_usage_user` , `cpu_usage_system` Disco: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem: `mem_used_percent` Rede: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Netstat: `netstat_tcp_established` , `netstat_tcp_time_wait` Swap: `swap_used_percent`

Instâncias do Amazon EC2 que executam o Windows Server

> **Note**
>
> Os nomes das métricas listados nesta tabela mostram como a métrica aparece quando visualizada no console. O nome real da métrica pode não incluir a primeira palavra. Por exemplo, o nome real da métrica `LogicalDisk % Free Space` é apenas `% Free Space`.

Nível de detalhes	Métricas incluídas
Basic	Memória: `Memory % Committed Bytes In Use` LogicalDisk: `LogicalDisk % Free Space`
Padrão	Memória: `Memory % Committed Bytes In Use` Paginação: `Paging File % Usage`

Nível de detalhes	Métricas incluídas
	Processador: `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` PhysicalDisk: `PhysicalDisk % Disk Time` LogicalDisk: `LogicalDisk % Free Space`
Advanced (Avançado)	Memória: `Memory % Committed Bytes In Use` Paginação: `Paging File % Usage` Processador: `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` TCP: `TCPv4 Connections Established` , `TCPv6 Connections Established`

Servidor on-premises que executa o Windows Server

> **Note**
>
> Os nomes das métricas listados nesta tabela mostram como a métrica aparece quando visualizada no console. O nome real da métrica pode não incluir a primeira palavra. Por exemplo, o nome real da métrica `LogicalDisk % Free Space` é apenas `% Free Space`.

Nível de detalhes	Métricas incluídas
Basic	Paginação: `Paging File % Usage` Processador: `Processor % Processor Time` LogicalDisk:`LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memória: `Memory % Committed Bytes In Use` Interface de rede: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`
Padrão	Paginação: `Paging File % Usage` Processador: `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memória: `Memory % Committed Bytes In Use` Interface de rede: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`
Advanced (Avançado)	Paginação:`Paging File % Usage`

Nível de detalhes	Métricas incluídas
	Processador: `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memória: `Memory % Committed Bytes In Use` Interface de rede: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec` TCP: `TCPv4 Connections Established` , `TCPv6 Connections Established`

Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch

O arquivo de configuração do agente do CloudWatch é um arquivo JSON com quatro seções: `agent`, `metrics`, `logs` e `traces`, descritas como a seguir:

- A seção `agent` inclui campos de configuração geral do atendente.
- A seção `metrics` especifica as métricas personalizadas para coleta e publicação do CloudWatch. Se estiver usando o atendente apenas para coletar logs, você poderá omitir a seção `metrics` do arquivo.
- A seção `logs` especifica quais arquivos de log sewrão publicados no CloudWatch Logs. Isso pode incluir eventos do Log de Eventos do Windows se o servidor executar o Windows Server.
- A seção `traces` especifica as fontes de rastreamentos que são coletados e enviados para AWS X-Ray.

As seções a seguir explicam a estrutura e os campos deste arquivo JSON. Você também pode visualizar a definição de esquema para esse arquivo de configuração. A definição de esquema está localizada em *installation-directory*/doc/amazon-cloudwatch-agent-schema.json nos servidores Linux e em *installation-directory*/amazon-cloudwatch-agent-schema.json nos servidores que executam o Windows Server.

Se criar ou editar o arquivo de configuração do atendente do manualmente, você poderá atribuir qualquer nome a ele. Para simplificar a solução de problemas, recomendamos que você nomeie-o /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json em um servidor Linux e $Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json nos servidores que executam o Windows Server. Depois de criar o arquivo, copie-o em outros servidores em que você deseja instalar o atendente.

> **Note**
>
> Métricas, logs e rastreamentos coletados pelo agente do CloudWatch incorrem em cobranças. Para obter mais informações sobre a definição de preço, consulte Preços do Amazon CloudWatch.

Arquivo de configuração do atendente do CloudWatch: seção do atendente

A seção agent pode incluir os seguintes campos. O assistente não cria uma seção agent. Em vez disso, o assistente a omite e usa os valores padrão para todos os campos nesta seção.

- metrics_collection_interval: opcional. Especifica a frequência com que todas as métricas especificadas nesse arquivo de configuração serão coletadas. Você pode substituir esse valor por tipos específicos de métricas.

  Esse valor é especificado em segundos. Por exemplo, a especificação de 10 faz com que as métricas sejam coletadas a cada 10 segundos. Uma configuração de 300 especifica que as métricas sejam coletadas a cada 5 minutos.

  Se você definir esse valor abaixo de 60 segundos, cada métrica será coletada como uma métrica de alta resolução. Para obter mais informações sobre as métricas de alta resolução, consulte Métricas de alta resolução.

  O valor padrão é 60.

- `region`: especifica a região a ser usada para o endpoint do CloudWatch quando uma instância do Amazon EC2 está sendo monitorada. As métricas coletadas são enviadas para essa região, como `us-west-1`. Se você omitir esse campo, o atendente enviará métricas para a região onde a instância do Amazon EC2 está localizada.

  Se você estiver monitorando um servidor on-premises, esse campo não será usado, e o atendente lerá a região do perfil `AmazonCloudWatchAgent` do arquivo de configuração da AWS.
- `credentials`: especifica um perfil do IAM a ser usado ao enviar métricas, logs e rastreamentos a uma conta da AWS diferente. Se especificado, esse campo contém um parâmetro, `role_arn`.
  - `role_arn`: especifica o nome do recurso da Amazon (ARN) de um perfil do IAM a ser usado para autenticação ao enviar métricas, logs e rastreamentos a uma conta da AWS diferente. Para ter mais informações, consulte Envio de métricas, logs e rastreamentos a uma conta diferente.
- `debug`: opcional. Especifica a execução do atendente do CloudWatch com mensagens do log de depuração. O valor padrão é `false`.
- `aws_sdk_log_level`: opcional. Com suporte somente para a versão 1.247350.0 e para as versões posteriores do agente do CloudWatch.

  Você pode especificar esse campo para que o atendente realizar o registro em log para endpoints do AWS SDK. O valor desse campo pode incluir uma ou mais das opções a seguir. Separe várias opções com o caractere `|`.
  - `LogDebug`
  - `LogDebugWithSigning`
  - `LogDebugWithHTTPBody`
  - `LogDebugRequestRetries`
  - `LogDebugWithEventStreamBody`

  Para mais informações sobre essas opções, consulte LogLevelType.
- `logfile`: especifica o local onde o atendente do CloudWatch grava mensagens de log. Se você especificar uma string vazia, o log irá para stderr. Se você não especificar essa opção, os locais padrão são os seguintes:
  - Linux: `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`
  - Windows Server: `c:\\ProgramData\\Amazon\\CloudWatchAgent\\Logs\\amazon-cloudwatch-agent.log`

O atendente do CloudWatch alterna automaticamente o arquivo de log criado. Um arquivo de log é alternado quando ele atinge 100 MB de tamanho. O atendente mantém os arquivos de log alternados por até sete dias, e ele mantém até cinco arquivos de log de backup que foram alternados. Os arquivos de log de backup têm um timestamp anexado ao seu nome de arquivo. O timestamp mostra a data e a hora em que o arquivo foi alternado: por exemplo, `amazon-cloudwatch-agent-2018-06-08T21-01-50.247.log.gz`.

- `omit_hostname`: opcional. Por padrão, o nome do host é publicado como uma dimensão de métricas coletadas pelo atendente, a menos que você esteja usando o campo `append_dimensions` da seção `metrics`. Defina `omit_hostname` como `true` para impedir que o nome do host seja publicado como uma dimensão, mesmo se você não estiver usando `append_dimensions`. O valor padrão é `false`.
- `run_as_user`: opcional. Especifica um usuário a ser usado para executar o atendente do CloudWatch. Se você não especificar esse parâmetro, o usuário raiz será usado. Essa opção é válida apenas em servidores Linux.

  Se você especificar essa opção, o usuário deverá existir antes que o atendente do CloudWatch seja iniciado. Para ter mais informações, consulte Executar o atendente do CloudWatch como um usuário diferente.
- `user_agent`: opcional. Especifica a string `user-agent` que é usada pelo atendente do CloudWatch quando ele faz chamadas de API ao backend do CloudWatch. O valor padrão é uma string que consiste na versão do atendente, a versão da linguagem de programação Go que foi usada para compilar o atendente, o sistema operacional e a arquitetura de runtime, o tempo de compilação e os plugins habilitados.
- `usage_data`: optional. Por padrão, o agente do CloudWatch envia dados de integridade e performande sobre si mesmo para o CloudWatch sempre que publica métricas ou logs no CloudWatch. Esses dados não geram custos para você. É possível impedir que o agente envie esses dados especificando `false` em `usage_data`. Se você omitir esse parâmetro, o padrão `true` será usado e o agente enviará dados de integridade e performance.

  Se você definir esse valor como `false`, deverá interromper e reiniciar o agente para que ele tenha efeito.

A seguir, temos um exemplo de uma seção `agent`.

```
"agent": {
   "metrics_collection_interval": 60,
```

```
  "region": "us-west-1",
  "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log",
  "debug": false,
  "run_as_user": "cwagent"
 }
```

Arquivo de configuração do atendente do CloudWatch: seção de métricas

Campos comuns ao Linux e Windows

Em servidores que executam o Linux ou o Windows Server, a seção `metrics` inclui os seguintes campos:

- `namespace`: opcional. O namespace a ser usado para as métricas coletadas pelo atendente. O valor padrão é `CWAgent`. O tamanho máximo é de 255 caracteres. Veja um exemplo a seguir:

  ```
  {
    "metrics": {
      "namespace": "Development/Product1Metrics",
     ......
     },
  }
  ```

- `append_dimensions`: opcional. Adiciona dimensões de métrica do Amazon EC2 a todas as métricas coletadas pelo atendente. Isso também faz com que o atendente não publique o nome do host como dimensão.

  Os únicos pares de chave-valor compatíveis com `append_dimensions` são exibidos na lista a seguir. Todos os outros pares de chave-valor são ignorados. O agente oferece suporte a esses pares de chave-valor exatamente da forma como são mostrados na lista a seguir. Não é possível alterar os valores de chaves para publicar nomes de dimensões diferentes para eles.

  - `"ImageId":"${aws:ImageId}"` define o ID da AMI da instância como o valor da dimensão `ImageId`.
  - `"InstanceId":"${aws:InstanceId}"` define o ID da instância como o valor da dimensão `InstanceId`.
  - `"InstanceType":"${aws:InstanceType}"` define o tipo de instância da instância como o valor da dimensão `InstanceType`.
  - `"AutoScalingGroupName":"${aws:AutoScalingGroupName}"` define o nome do grupo do Auto Scaling da instância como o valor da dimensão `AutoScalingGroupName`.

Se você quiser anexar dimensões a métricas com pares de chave/valor arbitrários, use o parâmetro `append_dimensions` no campo para esse tipo específico de métrica.

Se você especificar um valor que depende dos metadados do Amazon EC2 e usar proxies, deverá verificar se o servidor pode acessar o endpoint do Amazon EC2. Para obter mais informações sobre esses endpoints, consulte Amazon Elastic Compute Cloud (Amazon EC2) no Referência geral da Amazon Web Services.

- `aggregation_dimensions`: opcional. Especifica as dimensões nas quais as métricas coletadas serão agregadas. Por exemplo, se você acumular métricas na dimensão `AutoScalingGroupName`, as métricas de todas as instâncias em cada grupo do Auto Scaling serão agregadas e poderão ser visualizadas como um todo.

  Você pode acumular métricas em dimensões únicas ou múltiplas. Por exemplo, se você especificar `[["InstanceId"], ["InstanceType"], ["InstanceId","InstanceType"]]`, agregará métricas para ID de instância individualmente, tipo de instância individualmente e para a combinação das duas dimensões.

  Você também pode especificar `[]` para acumular todas as métricas em uma única coleção, desconsiderando todas as dimensões.

- `endpoint_override`: especifica um endpoint FIPS ou um link privado a ser usado como o endpoint onde o atendente envia métricas. Especificá-lo e definir um link privado permite enviar as métricas para um endpoint da Amazon VPC. Para obter mais informações, consulte O que é a Amazon VPC?

  O valor de `endpoint_override` deve ser uma string que seja um URL.

  Por exemplo, a parte a seguir da seção de métricas do arquivo de configuração define o atendente para usar um VPC endpoint ao enviar métricas.

```
{
  "metrics": {
    "endpoint_override": "vpce-XXXXXXXXXXXXXXXXXXXXXXXX.monitoring.us-
east-1.vpce.amazonaws.com",
   ......
   },
}
```

- `metrics_collected`: obrigatório. Especifica quais métricas serão coletadas, incluindo métricas personalizadas coletadas por meio do `StatsD` ou `collectd`. Essa seção inclui várias subseções.

  O conteúdo da seção `metrics_collected` varia dependendo do arquivo de configuração ser para um servidor que executa o Linux ou o Windows Server.

- `force_flush_interval`: especifica em segundos a quantidade máxima de tempo em que as métricas permanecem no buffer da memória antes de serem enviadas ao servidor. Qualquer que seja o valor configurado, se o volume de métricas no buffer atingir 1 MB ou 1000 métricas diferentes, as métricas serão enviadas imediatamente para o servidor.

  O valor padrão é 60.

- `credentials`: especifica uma função do IAM a ser usada ao enviar métricas a uma conta diferente. Se especificado, esse campo contém um parâmetro, `role_arn`.
  - `role_arn`: especifica o ARN de uma função do IAM a ser usada para autenticação ao enviar métricas a uma conta diferente. Para ter mais informações, consulte Envio de métricas, logs e rastreamentos a uma conta diferente. Se especificado aqui, esse valor substituirá o `role_arn` especificado na seção `agent` do arquivo de configuração, se houver.

Seção Linux

Em servidores que executem o Linux, a seção `metrics_collected` do arquivo de configuração também pode conter os campos a seguir.

Muitos desses campos podem incluir seções `measurement` que listam as métricas que você deseja coletar para esse recurso. Essas seções `measurement` podem especificar o nome completo da métrica, como `swap_used`, ou apenas a parte do nome da métrica que será acrescentada ao tipo de recurso. Por exemplo, especificar `reads` na seção `measurement` da seção `diskio` fará com que a métrica `diskio_reads` a seja coletada.

- `collectd`: opcional. Especifica que você deseja recuperar métricas personalizadas usando o protocolo `collectd`. USe o software `collectd` para enviar as métricas ao atendente do CloudWatch. Para obter mais informações sobre as opções de configuração disponíveis para o collectd, consulte Recuperar métricas personalizadas com o collectd.
- `cpu`: opcional. Especifica que as métricas da CPU deverão ser coletadas. Essa seção é válida apenas para instâncias do Linux. É necessário incluir pelo menos um dos campos `totalcpu` e `resources` para todas as métricas de CPU a serem coletadas. A seção pode incluir os seguintes campos:

- `drop_original_metrics`: opcional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.
- `resources`: optional. Especifique esse campo com um valor de `*` para fazer com que as métricas por CPU sejam coletadas. O único valor permitido é `*`.
- `totalcpu`: opcional. Especifica se as métricas de cpu agregadas em todos os núcleos de cpu serão relatadas. O padrão é true.
- `measurement`: especifica a matriz de métricas de cpu a serem coletadas. Os valores possíveis são `time_active`, `time_guest`, `time_guest_nice`, `time_idle`, `time_iowait`, `time_irq`, `time_nice`, `time_softirq`, `time_steal`, `time_system`, `time_user`, `usage_active`, `usage_guest`, `usage_guest_nice`, `usage_idle`, `usage_iowait`, `usage_irq`, `usage_nice`, `usage_softirq`, `usage_steal`, `usage_system` e `usage_user`. Esse campo será obrigatório se você incluir `cpu`.

  Por padrão, a unidade para métricas de `cpu_usage_*` é `Percent`, e as métricas `cpu_time_*` não têm uma unidade.

  Dentro da entrada de cada métrica individual, você também poderá especificar opcionalmente uma ou ambas as opções:

  - `rename`: especifica um nome diferente para essa métrica.
  - `unit`: especifica a unidade a ser usada para essa métrica, substituindo a unidade padrão de `None` para a métrica. A unidade que você especificar deverá ser uma unidade de métrica válida do CloudWatch, conforme listado na descrição do `Unit` em MetricDatum.
- `metrics_collection_interval`: opcional. Especifica a frequência da coleta de métricas de cpu, substituindo o `metrics_collection_interval` global especificado na seção `agent` do arquivo de configuração.

  Esse valor é especificado em segundos. Por exemplo, a especificação de 10 faz com que as métricas sejam coletadas a cada 10 segundos. Uma configuração de 300 especifica que as métricas sejam coletadas a cada 5 minutos.

Se você definir esse valor abaixo de 60 segundos, cada métrica será coletada como uma métrica de alta resolução. Para obter mais informações sobre as métricas de alta resolução, consulte Métricas de alta resolução.

- `append_dimensions`: optional. Dimensões adicionais a serem usadas somente para métricas de cpu. Se você especificar esse campo, ele será usado em complemento às dimensões especificadas no campo `append_dimensions` global que é usado para todos os tipos de métricas coletadas pelo atendente.

- `disk`: optional. Especifica que as métricas de disco deverão ser coletadas. Essa seção é válida apenas para instâncias do Linux. A seção pode incluir os seguintes campos:
  - `drop_original_metrics`: opcional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.
  - `resources`: optional. Especifica uma matriz de pontos de montagem de disco. Esse campo limita o CloudWatch a coletar métricas apenas dos pontos de montagem listados. Você pode especificar * como o valor para coletar métricas de todos os pontos de montagem. O valor padrão é coletar métricas de todos os pontos de montagem.
  - `measurement`: especifica a matriz de métricas de disco a serem coletadas. Os valores possíveis são `free`, `total`, `used`, `used_percent`, `inodes_free`, `inodes_used` e `inodes_total`. Esse campo será obrigatório se você incluir `disk`.

    > **Note**
    >
    > As métricas de `disk` têm uma dimensão para `Partition`, o que significa que o número de métricas personalizadas geradas depende do número de partições associadas à instância. O número de partições de disco que você tem depende da AMI usada e do número de volumes do Amazon EBS que estão sendo anexados ao servidor.

    Para ver as unidades padrão para cada métrica de `disk`, consulte Métricas coletadas pelo atendente do CloudWatch em instâncias do Linux e macOS.

Dentro da entrada de cada métrica individual, você também poderá especificar opcionalmente uma ou ambas as opções:

- `rename`: especifica um nome diferente para essa métrica.
- `unit`: especifica a unidade a ser usada para essa métrica, substituindo a unidade padrão de None de None para a métrica. A unidade que você especificar deverá ser uma unidade de métrica válida do CloudWatch, conforme listado na descrição do `Unit` em MetricDatum.

- `ignore_file_system_types`: especifica os tipos de sistema de arquivos a serem excluídos ao coletar métricas de disco. Os valores válidos incluem `sysfs`, `devtmpfs`, e assim por diante.
- `drop_device`: definir isso como `true` fará com que `Device` não seja incluído como uma dimensão para as métricas de disco.

  Impedir que `Device` seja usado como uma dimensão pode ser útil em instâncias que usam o sistema Nitro porque, nessas instâncias, os nomes dos dispositivos mudam para cada montagem de disco quando a instância é reinicializada. Isso pode causar dados inconsistentes em suas métricas e fazer com que os alarmes com base nessas métricas entrem no estado `INSUFFICIENT DATA`.

  O padrão é `false`.
- `metrics_collection_interval`: optional. Especifica a frequência da coleta de métricas de disco, substituindo o `metrics_collection_interval` global especificado na seção `agent` do arquivo de configuração.

  Esse valor é especificado em segundos.

  Se você definir esse valor abaixo de 60 segundos, cada métrica será coletada como uma métrica de alta resolução. Para ter mais informações, consulte Métricas de alta resolução.
- `append_dimensions`: opcional. Dimensões adicionais a serem usadas somente para métricas de disco. Se você especificar esse campo, ele será usado em complemento às dimensões especificadas no campo `append_dimensions` que é usado para todos os tipos de métricas coletadas pelo atendente.

- `diskio`: optional. Especifica que as métricas de e/s de disco deverão ser coletadas. Essa seção é válida apenas para instâncias do Linux. A seção pode incluir os seguintes campos:
  - `drop_original_metrics`: opcional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são

separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.

- `resources`: optional. Se você especificar uma matriz de dispositivos, o CloudWatch coletará métricas apenas desses dispositivos. Caso contrário, serão coletadas métricas para todos os dispositivos. Você também pode especificar * como valor para coletar métricas de todos os dispositivos.
- `measurement`: especifica a matriz de métricas de e/s de disco a serem coletadas. Os valores possíveis são `reads`, `writes`, `read_bytes`, `write_bytes`, `read_time`, `write_time`, `io_time` e `iops_in_progress`. Esse campo será obrigatório se você incluir `diskio`.

  Dentro da entrada de cada métrica individual, você também poderá especificar opcionalmente uma ou ambas as opções:

  - `rename`: especifica um nome diferente para essa métrica.
  - `unit`: especifica a unidade a ser usada para essa métrica, substituindo a unidade padrão de `None` de `None` para a métrica. A unidade que você especificar deverá ser uma unidade de métrica válida do CloudWatch, conforme listado na descrição do `Unit` em MetricDatum.
- `metrics_collection_interval`: opcional. Especifica a frequência da coleta de métricas de diskio, substituindo o `metrics_collection_interval` global especificado na seção `agent` do arquivo de configuração.

  Esse valor é especificado em segundos.

  Se você definir esse valor abaixo de 60 segundos, cada métrica será coletada como uma métrica de alta resolução. Para obter mais informações sobre as métricas de alta resolução, consulte Métricas de alta resolução.
- `append_dimensions`: optional. Dimensões adicionais a serem usadas somente para métricas de diskio. Se você especificar esse campo, ele será usado em complemento às dimensões especificadas no campo `append_dimensions` que é usado para todos os tipos de métricas coletadas pelo atendente.

- `swap`: optional. Especifica que as métricas de memória de permuta deverão ser coletadas. Essa seção é válida apenas para instâncias do Linux. A seção pode incluir os seguintes campos:
  - `drop_original_metrics`: opcional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados

agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.

- `measurement`: especifica a matriz de métricas de swap a serem coletadas. Os valores possíveis são `free`, `used` e `used_percent`. Esse campo será obrigatório se você incluir `swap`.

  Para ver as unidades padrão para cada métrica de `swap`, consulte Métricas coletadas pelo atendente do CloudWatch em instâncias do Linux e macOS.

  Dentro da entrada de cada métrica individual, você também poderá especificar opcionalmente uma ou ambas as opções:

  - `rename`: especifica um nome diferente para essa métrica.
  - `unit`: especifica a unidade a ser usada para essa métrica, substituindo a unidade padrão de `None` de `None` para a métrica. A unidade que você especificar deverá ser uma unidade de métrica válida do CloudWatch, conforme listado na descrição do `Unit` em MetricDatum.

- `metrics_collection_interval`: opcional. Especifica a frequência da coleta de métricas de memória de permuta, substituindo o `metrics_collection_interval` global especificado na seção `agent` do arquivo de configuração.

  Esse valor é especificado em segundos.

  Se você definir esse valor abaixo de 60 segundos, cada métrica será coletada como uma métrica de alta resolução. Para obter mais informações sobre as métricas de alta resolução, consulte Métricas de alta resolução.

- `append_dimensions`: optional. Dimensões adicionais a serem usadas somente para métricas de memória de permuta. Se você especificar esse campo, ele será usado em complemento às dimensões especificadas no campo `append_dimensions` global que é usado para todos os tipos de métricas coletadas pelo atendente. É coletado como uma métrica de alta resolução.

- `mem`: optional. Especifica que as métricas de memória deverão ser coletadas. Essa seção é válida apenas para instâncias do Linux. A seção pode incluir os seguintes campos:

  - `drop_original_metrics`: opcional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são

separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.

- `measurement`: especifica a matriz de métricas de memória a serem coletadas. Os valores possíveis são `active`, `available`, `available_percent`, `buffered`, `cached`, `free`, `inactive`, `total`, `used` e `used_percent`. Esse campo será obrigatório se você incluir `mem`.

  Para ver as unidades padrão para cada métrica de `mem`, consulte Métricas coletadas pelo atendente do CloudWatch em instâncias do Linux e macOS.

  Dentro da entrada de cada métrica individual, você também poderá especificar opcionalmente uma ou ambas as opções:

  - `rename`: especifica um nome diferente para essa métrica.
  - `unit`: especifica a unidade a ser usada para essa métrica, substituindo a unidade padrão de `None` para a métrica. A unidade que você especificar deverá ser uma unidade de métrica válida do CloudWatch, conforme listado na descrição do `Unit` em MetricDatum.
- `metrics_collection_interval`: opcional. Especifica a frequência da coleta de métricas de memória, substituindo o `metrics_collection_interval` global especificado na seção `agent` do arquivo de configuração.

  Esse valor é especificado em segundos.

  Se você definir esse valor abaixo de 60 segundos, cada métrica será coletada como uma métrica de alta resolução. Para obter mais informações sobre as métricas de alta resolução, consulte Métricas de alta resolução.
- `append_dimensions`: optional. Dimensões adicionais a serem usadas somente para métricas de memória. Se você especificar esse campo, ele será usado em complemento às dimensões especificadas no campo `append_dimensions` que é usado para todos os tipos de métricas coletadas pelo atendente.

- `net`: optional. Especifica que as métricas de redes deverão ser coletadas. Essa seção é válida apenas para instâncias do Linux. A seção pode incluir os seguintes campos:
  - `drop_original_metrics`: opcional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são

separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.

- `resources`: optional. Se você especificar uma matriz de interfaces de rede, o CloudWatch coletará métricas apenas dessas interfaces. Caso contrário, serão coletadas métricas para todos os dispositivos. Você também pode especificar `*` como o valor para coletar métricas de todas as interfaces.
- `measurement`: especifica a matriz de métricas de redes a serem coletadas. Os valores possíveis são `bytes_sent`, `bytes_recv`, `drop_in`, `drop_out`, `err_in`, `err_out`, `packets_sent` e `packets_recv`. Esse campo será obrigatório se você incluir `net`.

  Para ver as unidades padrão para cada métrica de `net`, consulte Métricas coletadas pelo atendente do CloudWatch em instâncias do Linux e macOS.

  Dentro da entrada de cada métrica individual, você também poderá especificar opcionalmente uma ou ambas as opções:

  - `rename`: especifica um nome diferente para essa métrica.
  - `unit`: especifica a unidade a ser usada para essa métrica, substituindo a unidade padrão de `None` para a métrica. A unidade que você especificar deverá ser uma unidade de métrica válida do CloudWatch, conforme listado na descrição do `Unit` em MetricDatum.
- `metrics_collection_interval`: opcional. Especifica a frequência da coleta de métricas de redes, substituindo o `metrics_collection_interval` global especificado na seção `agent` do arquivo de configuração.

  Esse valor é especificado em segundos. Por exemplo, a especificação de 10 faz com que as métricas sejam coletadas a cada 10 segundos. Uma configuração de 300 especifica que as métricas sejam coletadas a cada 5 minutos.

  Se você definir esse valor abaixo de 60 segundos, cada métrica será coletada como uma métrica de alta resolução. Para obter mais informações sobre as métricas de alta resolução, consulte Métricas de alta resolução.
- `append_dimensions`: optional. Dimensões adicionais a serem usadas somente para métricas de redes. Se você especificar esse campo, ele será usado em complemento às dimensões

especificadas no campo `append_dimensions` que é usado para todos os tipos de métricas coletadas pelo atendente.

- `netstat`: optional. Especifica que as métricas de estado de conexão TCP e conexão UDP deverão ser coletadas. Essa seção é válida apenas para instâncias do Linux. A seção pode incluir os seguintes campos:
  - `drop_original_metrics`: opcional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.
  - `measurement`: especifica a matriz de métricas de netstat a serem coletadas. Os valores possíveis são `tcp_close`, `tcp_close_wait`, `tcp_closing`, `tcp_established`, `tcp_fin_wait1`, `tcp_fin_wait2`, `tcp_last_ack`, `tcp_listen`, `tcp_none`, `tcp_syn_sent`, `tcp_syn_recv`, `tcp_time_wait` e `udp_socket`. Esse campo será obrigatório se você incluir `netstat`.

    Para ver as unidades padrão para cada métrica de `netstat`, consulte Métricas coletadas pelo atendente do CloudWatch em instâncias do Linux e macOS.

    Dentro da entrada de cada métrica individual, você também poderá especificar opcionalmente uma ou ambas as opções:
    - `rename`: especifica um nome diferente para essa métrica.
    - `unit`: especifica a unidade a ser usada para essa métrica, substituindo a unidade padrão de `None` para a métrica. A unidade que você especificar deverá ser uma unidade de métrica válida do CloudWatch, conforme listado na descrição do `Unit` em MetricDatum.
  - `metrics_collection_interval`: opcional. Especifica a frequência da coleta de métricas de netstat, substituindo o `metrics_collection_interval` global especificado na seção `agent` do arquivo de configuração.

    Esse valor é especificado em segundos.

Se você definir esse valor abaixo de 60 segundos, cada métrica será coletada como uma métrica de alta resolução. Para obter mais informações sobre as métricas de alta resolução, consulte Métricas de alta resolução.

- `append_dimensions`: optional. Dimensões adicionais a serem usadas somente para métricas de netstat. Se você especificar esse campo, ele será usado em complemento às dimensões especificadas no campo `append_dimensions` que é usado para todos os tipos de métricas coletadas pelo atendente.

- `processes`: optional. Especifica que as métricas de processo deverão ser coletadas. Essa seção é válida apenas para instâncias do Linux. A seção pode incluir os seguintes campos:
  - `drop_original_metrics`: opcional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.
  - `measurement`: especifica a matriz de métricas de processos a serem coletadas. Os valores possíveis são `blocked`, `dead`, `idle`, `paging`, `running`, `sleeping`, `stopped`, `total`, `total_threads`, `wait` e `zombies`. Esse campo será obrigatório se você incluir `processes`.

    Para todas as métricas `processes`, a unidade padrão é `None`.

    Dentro da entrada de cada métrica individual, você também poderá especificar opcionalmente uma ou ambas as opções:
    - `rename`: especifica um nome diferente para essa métrica.
    - `unit`: especifica a unidade a ser usada para essa métrica, substituindo a unidade padrão de `None` para a métrica. A unidade que você especificar deverá ser uma unidade de métrica válida do CloudWatch, conforme listado na descrição do `Unit` em MetricDatum.
  - `metrics_collection_interval`: opcional. Especifica a frequência da coleta de métricas de processos, substituindo o `metrics_collection_interval` global especificado na seção `agent` do arquivo de configuração.

Esse valor é especificado em segundos. Por exemplo, a especificação de 10 faz com que as métricas sejam coletadas a cada 10 segundos. Uma configuração de 300 especifica que as métricas sejam coletadas a cada 5 minutos.

Se você definir esse valor abaixo de 60 segundos, cada métrica será coletada como uma métrica de alta resolução. Para ter mais informações, consulte Métricas de alta resolução.

- `append_dimensions`: opcional. Dimensões adicionais a serem usadas somente para métricas de processo. Se você especificar esse campo, ele será usado em complemento às dimensões especificadas no campo `append_dimensions` que é usado para todos os tipos de métricas coletadas pelo atendente.

- `nvidia_gpu`: optional. Especifica que as métricas de GPU NVIDIA deverão ser coletadas. Esta seção é válida somente para instâncias Linux em hosts configurados com um acelerador de GPU NVIDIA e que tenham a NVIDIA System Management Interface (nvidia-smi) instalada.

  As métricas de GPU NVIDIA coletadas são prefixadas com a string `nvidia_smi_` para distingui-las das métricas coletadas para outros tipos de acelerador. A seção pode incluir os seguintes campos:

  - `drop_original_metrics`: opcional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.
  - `measurement`: especifica a matriz de métricas GPU NVIDIA a serem coletadas. Para obter uma lista dos valores de uso possíveis, consulte a coluna Metric (Métrica) na tabela em Colete métricas de GPU NVIDIA.

    Na entrada de cada métrica individual, você também poderá especificar uma ou ambas das seguintes opções:

    - `rename`: especifica um nome diferente para essa métrica.
    - `unit`: especifica a unidade a ser usada para essa métrica, substituindo a unidade padrão de `None` para a métrica. A unidade que você especificar deverá ser uma unidade de métrica válida do CloudWatch, conforme listado na descrição do `Unit` em MetricDatum.

  - `metrics_collection_interval`: opcional. Especifica a frequência da coleta de métricas GPU NVIDIA, substituindo o `metrics_collection_interval` global especificado na seção `agent` do arquivo de configuração.
- `procstat`: optional. Especifica se você deseja recuperar métricas de processos individuais. Para obter mais informações sobre as opções de configuração disponíveis para o procstat, consulte Coletar métricas de processo com o plugin procstat.
- `statsd`: opcional. Especifica que você deseja recuperar métricas personalizadas usando o protocolo `StatsD`. O atendente do CloudWatch funciona como daemon para o protocolo. Use qualquer cliente do `StatsD` padrão para enviar as métricas ao atendente do CloudWatch. Para obter mais informações sobre as opções de configuração disponíveis para o StatsD, consulte Recuperar métricas personalizadas com o StatsD .
- `ethtool`: opcional. Especifica que você deseja recuperar métricas de rede usando o plugin `ethtool`. Este plugin pode importar as métricas coletadas pelo utilitário ethtool padrão, bem como métricas de performance de rede das instâncias do Amazon EC2. Para obter mais informações sobre as opções de configuração disponíveis para o ethtool, consulte Coletar métricas de performance da rede.

Veja a seguir o exemplo de uma seção `metrics` para um servidor Linux. Neste exemplo, três métricas de CPU, três métricas de netstat, três métricas de processo e uma métrica de disco são coletadas, e o atendente é configurado para receber métricas adicionais a partir de um cliente `collectd`.

```
"metrics": {
    "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId",
 "InstanceType"],[]],
    "metrics_collected": {
      "collectd": {},
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit": "Percent"},
          {"name": "cpu_usage_nice", "unit": "Percent"},
          "cpu_usage_guest"
        ],
        "totalcpu": false,
        "drop_original_metrics": [ "cpu_usage_guest" ],
```

```
        "metrics_collection_interval": 10,
        "append_dimensions": {
          "test": "test1",
          "date": "2017-10-01"
        }
      },
      "netstat": {
        "measurement": [
          "tcp_established",
          "tcp_syn_sent",
          "tcp_close"
        ],
        "metrics_collection_interval": 60
      },
       "disk": {
        "measurement": [
          "used_percent"
        ],
        "resources": [
          "*"
        ],
        "drop_device": true
      },
      "processes": {
        "measurement": [
          "running",
          "sleeping",
          "dead"
        ]
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    }
  }
```

Windows Server

Na seção `metrics_collected` para Windows Server, você pode ter subseções para cada objeto de performance do Windows, como `Memory`, `Processor` e `LogicalDisk`. Para obter informações

sobre quais objetos e contadores estão disponíveis, consulte Contadores de performance na documentação do Microsoft Windows.

Na subseção de cada objeto, você especifica uma matriz de `measurement` de contadores a serem coletados. A matriz `measurement` é necessária para cada objeto que você especifica no arquivo de configuração. Você também pode especificar um campo `resources` para nomear as instâncias das quais as métricas serão coletadas. Você também pode especificar `*` para `resources` para coletar métricas separadas para cada instância. Se você omitir `resources` para contadores que tenham instâncias, os dados de todas as instâncias serão agregados em um conjunto. Se você omitir `resources` para contadores que não tenham instâncias, os contadores não são coletados pelo agente do CloudWatch. Para determinar se os contadores têm instâncias, é possível usar um dos comandos a seguir.

Powershell:

```
Get-Counter -ListSet *
```

Linha de comando (não na Powershell):

```
TypePerf.exe -q
```

Dentro de cada seção de objeto, você também pode especificar os campos opcionais a seguir:

- `metrics_collection_interval`: optional. Especifica a frequência da coleta de métricas para esse objeto, substituindo o `metrics_collection_interval` global especificado na seção `agent` do arquivo de configuração.

  Esse valor é especificado em segundos. Por exemplo, a especificação de 10 faz com que as métricas sejam coletadas a cada 10 segundos. Uma configuração de 300 especifica que as métricas sejam coletadas a cada 5 minutos.

  Se você definir esse valor abaixo de 60 segundos, cada métrica será coletada como uma métrica de alta resolução. Para ter mais informações, consulte Métricas de alta resolução.
- `append_dimensions`: opcional. Especifica dimensões adicionais a serem usadas somente para métricas desse objeto. Se você especificar esse campo, ele será usado em complemento às dimensões especificadas no campo `append_dimensions` global que é usado para todos os tipos de métricas coletadas pelo atendente.

- `drop_original_metrics`: optional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.

Dentro de cada seção de contador, você também pode especificar os campos opcionais a seguir:

- `rename`: especifica um nome diferente a ser usado no CloudWatch para essa métrica.
- `unit`: especifica a unidade a ser usada para essa métrica. A unidade que você especificar deverá ser uma unidade de métrica válida do CloudWatch, conforme listado na descrição do `Unit` em MetricDatum.

Há duas outras seções opcionais que é possível incluir em `metrics_collected`:

- `statsd`: permite recuperar métricas personalizadas usando o protocolo `StatsD`. O atendente do CloudWatch funciona como daemon para o protocolo. Use qualquer cliente do `StatsD` padrão para enviar as métricas ao atendente do CloudWatch. Para ter mais informações, consulte Recuperar métricas personalizadas com o StatsD .
- `procstat`: permite recuperar métricas de processos individuais. Para ter mais informações, consulte Coletar métricas de processo com o plugin procstat.

Veja a seguir um exemplo de uma seção `metrics` para uso em um Windows Server. Neste exemplo, muitas métricas do Windows são coletadas e o computador também é definido de modo a receber métricas adicionais de um cliente `StatsD`.

```
"metrics": {
    "metrics_collected": {
      "statsd": {},
      "Processor": {
        "measurement": [
          {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
          "% Interrupt Time",
          "% User Time",
```

```
          "% Processor Time"
        ],
        "resources": [
          "*"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      },
      "LogicalDisk": {
        "measurement": [
          {"name": "% Idle Time", "unit": "Percent"},
          {"name": "% Disk Read Time", "rename": "DISK_READ"},
          "% Disk Write Time"
        ],
        "resources": [
          "*"
        ]
      },
      "Memory": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Available Bytes",
          "Cache Faults/sec",
          "Page Faults/sec",
          "Pages/sec"
        ],
        "append_dimensions": {
          "d3": "win_bo"
        }
      },
      "Network Interface": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Bytes Received/sec",
          "Bytes Sent/sec",
          "Packets Received/sec",
          "Packets Sent/sec"
        ],
        "resources": [
          "*"
        ],
        "append_dimensions": {
```

```
          "d3": "win_bo"
        }
      },
      "System": {
        "measurement": [
          "Context Switches/sec",
          "System Calls/sec",
          "Processor Queue Length"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    },
    "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"], ["d1"],[]]
    }
  }
```

Arquivo de configuração do atendente do CloudWatch: seção do Logs

A seção `logs` inclui os seguintes campos:

- `logs_collected`: obrigatório, se a seção `logs` for incluída. Especifica quais arquivos de log e logs de eventos do Windows deverão ser coletados no servidor. Pode incluir dois campos: `files` e `windows_events`.
  - `files`: especifica os arquivos de log regulares que deverão ser coletados pelo atendente do CloudWatch. Contém um campo, `collect_list`, que define melhor esses arquivos.
    - `collect_list`: obrigatório se `files` for incluído. Contém uma matriz de entradas, sendo que cada uma delas especifica um arquivo de log para coletar. Cada uma dessas entradas pode incluir os seguintes campos:
      - `file_path`: especifica o caminho do arquivo de log para carregar para o CloudWatch Logs. Regras correspondentes de glob do Unix padrão são aceitas, com a adição de `**` como um superasterisco. Por exemplo, se você especificar `/var/log/**.log`, isso fará

com que todos os arquivos `.log` na árvore de diretórios `/var/log` sejam coletados. Para obter mais exemplos, consulte a Biblioteca do glob.

Você também pode usar o asterisco padrão como um curinga padrão. Por exemplo, `/var/log/system.log*` corresponde a arquivos, como `system.log_1111`, `system.log_2222`, e assim por diante em `/var/log`.

Somente o arquivo mais recente é enviado ao CloudWatch Logs com base no tempo de modificação do arquivo. Recomendamos usar caracteres curinga para especificar uma série de arquivos do mesmo tipo, como `access_log.2018-06-01-01` e `access_log.2018-06-01-02`, mas não vários tipos de arquivos, como `access_log_80` e `access_log_443`. Para especificar vários tipos de arquivos, adicione outra entrada de stream de log ao arquivo de configuração de atendente para que cada tipo de arquivo de log vá para um stream de log diferente.

- `auto_removal`: optional. Se isso for `true`, o agente do CloudWatch excluirá automaticamente esse arquivo de log depois de lê-lo e de ele ter sido rotacionado. Normalmente, os arquivos de log são excluídos depois que todo o conteúdo é carregado no CloudWatch Logs, mas se o agente chegar ao EOF (fim do arquivo) e detectar outro arquivo de log mais novo que corresponda ao mesmo `file_path`, o agente exclui o arquivo ANTIGO. Portanto, verifique se terminou de gravar no arquivo ANTIGO antes de criar o arquivo NOVO. A biblioteca de rastreamento RUST tem uma incompatibilidade conhecida porque é provável que crie um NOVO arquivo de log e ainda tente gravar no arquivo de log ANTIGO.

  O atendente remove somente arquivos completos dos logs que criam vários arquivos, como logs que criam arquivos separados para cada data. Se um log gravar continuamente em um único arquivo, ele não será removido.

  Se você já tiver um método de rotação ou remoção do arquivo de log em vigor, recomendamos que você omita esse campo ou defina-o como `false`.

  Se você omitir esse campo, o valor padrão de `false` será usado.

- `log_group_name`: optional. Especifica o que usar como nome de grupo de logs no CloudWatch Logs.

  Recomendamos que você use esse campo para especificar um nome para o grupo de logs com a finalidade de evitar confusões. Se você omitir `log_group_name`, o valor de `file_path` até o ponto final será usado como o nome do grupo de logs. Por exemplo, se o

caminho do arquivo for `/tmp/TestLogFile.log.2017-07-11-14`, o nome do grupo de logs será `/tmp/TestLogFile.log`.

Caso especifique um nome para o grupo de logs, você poderá usar `{instance_id}`, `{hostname}`, `{local_hostname}` e `{ip_address}` como variáveis no nome. `{hostname}` recupera o nome do host usando os metadados do EC2 e `{local_hostname}` usa o nome do host do arquivo de configuração de rede.

Se você usar essas variáveis para criar diferentes grupos de logs, lembre-se do limite de 1.000.000 de grupos de log por conta, por região.

Os caracteres permitidos incluem a-z, A-Z, 0-9, "_" (sublinhado), "-" (hífen), "/" (barra) e "." (ponto).

- `log_group_class`: optional. Especifica qual classe do grupo de logs será usada para o novo grupo de logs. Para obter mais informações sobre as classes do grupo de logs, consulte Log classes.

  Os valores válidos são `STANDARD` e `INFREQUENT_ACCESS`. Se você omitir esse campo, o padrão de `STANDARD` será usado.

  > **Important**
  >
  > Após a criação de um grupo de logs, a classe não poderá ser alterada.

- `log_stream_name`: optional. Especifica o que usar como nome do fluxo de logs no CloudWatch Logs. Como parte do nome, você pode usar `{instance_id}`, `{hostname}`, `{local_hostname}` e `{ip_address}` como variáveis dentro do nome. O `{hostname}` recupera o nome do host de metadados do EC2, e o `{local_hostname}` usa o nome de host do arquivo de configuração de rede.

  Se você omitir esse campo, o valor do parâmetro `log_stream_name` na seção global de `logs` será usado. Se isso também for omitido, o valor padrão de `{instance_id}` será usado.

  Caso um fluxo de logs ainda não exista, ele será criado automaticamente.

- `retention_in_days`: optional. Especifica o número de dias em que os eventos de log serão retidos no grupo de logs especificado.

- Se o agente estiver criando esse grupo de logs agora e você omitir esse campo, a retenção desse novo grupo será definida para nunca expirar.
- Se esse grupo de logs já existir e você especificar esse campo, a nova retenção definida será utilizada. Se você omitir esse campo para um grupo de logs existente, a retenção do grupo de logs não será alterada.

  O assistente do agente do CloudWatch usará `-1` como valor padrão esse campo quando ele for usado para criar o arquivo de configuração do agente e você não especificar um valor para a retenção de logs. Esse valor `-1` definido pelo assistente especifica que os eventos no grupo de logs não expiram. No entanto, editar manualmente esse valor para não `-1` tem efeito.

Os valores válidos são 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 2192, 2557, 2922, 3288 e 3653..

Se você configurar o atendente para gravar vários fluxos de log no mesmo grupo de logs, especificar o `retention_in_days` em um só lugar definirá a retenção de log para todo o grupo de logs. Se você especificar `retention_in_days` para o mesmo grupo de logs em vários locais, a retenção será definida se todos esses valores forem iguais. No entanto, se diferentes valores de `retention_in_days` forem especificados para o mesmo grupo de logs em vários locais, a retenção de log não será definida e o atendente será interrompido, retornando um erro.

> **Note**
>
> A função do IAM do atendente ou o usuário do IAM deve ter o `logs:PutRetentionPolicy` para poder definir políticas de retenção. Para ter mais informações, consulte Permitir que o atendente do CloudWatch defina a política de retenção de logs.

> **Warning**
>
> Se você definir `retention_in_days` para um grupo de logs que já existe, todos os logs nesse grupo de logs publicados antes do número de dias que você especificar são excluídos. Por exemplo, configurá-lo como 3 faria com que todos os logs de 3 dias atrás e anteriores a isso fossem excluídos.

- `filters`: optional. Pode conter uma matriz de entradas, cada uma das quais especifica uma expressão regular e um tipo de filtro para especificar se as entradas de log que correspondem ao filtro devem ser publicadas ou soltas. Se você omitir esse campo, todos os logs no arquivo de log serão publicados no CloudWatch Logs. Se você incluir esse campo, o atendente processará cada mensagem de log com todos os filtros especificados, e somente os eventos de log que passam todos os filtros serão publicados no CloudWatch Logs. As entradas de registro que não passam todos os filtros ainda permanecerão no arquivo de log do host, mas não serão enviadas para o CloudWatch Logs.

  Cada entrada na matriz de filtros pode incluir os seguintes campos:

  - `type`: denota o tipo de filtro. Os valores válidos são `include` e `exclude`. Com `include`, a entrada de log deve corresponder à expressão a ser publicada no CloudWatch Logs. Com `exclude`, cada entrada de log que corresponde ao filtro não é enviada para o CloudWatch Logs.
  - `expression`: uma string de expressão regular que segue a Sintaxe RE2.

  > **Note**
  >
  > O atendente do CloudWatch não verifica a performance de nenhuma expressão regular que você fornece nem restringe o runtime da avaliação das expressões regulares. Recomendamos que você tenha cuidado para não escrever uma expressão que seja cara de avaliar. Para obter mais informações sobre possíveis problemas, consulte Negação de serviço de expressão regular – REDoS

  Por exemplo, o trecho a seguir do arquivo de configuração do atendente CloudWatch publica logs que são solicitações PUT e POST para o CloudWatch Logs, mas exclui logs provenientes do Firefox.

```
"collect_list": [
  {
    "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
    "log_group_name": "test.log",
    "log_stream_name": "test.log",
    "filters": [
      {
        "type": "exclude",
        "expression": "Firefox"
      },
```

```
      {
        "type": "include",
        "expression": "P(UT|OST)"
      }
    ]
  },
  .....
]
```

> **Note**
>
> A ordem dos filtros no arquivo de configuração é importante para a performance. No exemplo anterior, o atendente descarta todos os logs que correspondem a `Firefox` antes de começar a avaliar o segundo filtro. Para fazer com que menos entradas de log sejam avaliadas por mais de um filtro, coloque o filtro do qual você espera descartar mais logs primeiro no arquivo de configuração.

- `timezone`: optional. Especifica o fuso horário a ser usado durante a colocação de time stamps em eventos de log. Os valores válidos são `UTC` e `Local`. O valor padrão é `Local`.

  Esse parâmetro será ignorado se você não especificar um valor para `timestamp_format`.

- `timestamp_format`: optional. Especifica o formato do time stamp, usando texto simples e símbolos especiais que começam com %. Se você omitir esse campo, o tempo atual será usado. Se você usar esse campo, poderá usar os símbolos na lista a seguir como parte do formato.

  Se uma única entrada de log contiver dois time stamps correspondentes ao formato, o primeiro time stamp será usado.

  Esta lista de símbolos é diferente da lista usada pelo atendente do CloudWatch Logs mais antigo. Para obter um resumo dessas diferenças, consulte Diferenças de carimbo de data/hora entre o atendente unificado do CloudWatch e o atendente mais antigo do CloudWatch Logs.

  %y

  Ano sem o século como um número decimal preenchido com zero. Por exemplo, 19 para representar 2019.

`%Y`

Ano com o século como um número decimal. Por exemplo, `2019`.

`%b`

Mês como o nome abreviado da localidade

`%B`

Mês como o nome completo da localidade

`%m`

mês como um número decimal preenchido com zeros

`%-m`

Mês como um número decimal (não preenchido com zeros)

`%d`

dia do mês como um número decimal preenchido com zeros

`%-d`

Dia do mês como um número decimal (não preenchido com zeros)

`%A`

Nome completo do dia da semana, como `Monday`

`%a`

Abreviação do dia da semana, como `Mon`

`%H`

Hora (no formato de 24 horas) como um número decimal preenchido com zeros

`%I`

Hora (no formato de 12 horas) como um número decimal preenchido com zeros

`%-I`

Hora (no formato de 12 horas) como número decimal (não preenchido com zeros)

`%p`

AM ou PM

`%M`

Minutos como um número decimal preenchido com zeros

`%-M`

Minutos como um número decimal (não preenchido com zeros)

`%S`

Segundos como um número decimal preenchido com zeros

`%-S`

Segundos como um número decimal (não preenchido com zeros)

`%f`

Segundos fracionários como um número decimal (1 a 9 dígitos), preenchido com zeros à esquerda.

`%Z`

Fuso horário, por exemplo `PST`

`%z`

Fuso horário, expresso como a diferença entre o fuso horário local e o UTC. Por exemplo, `-0700`. Há suporte apenas para esse formato. Por exemplo, `-07:00` não é um formato válido.

- `multi_line_start_pattern`: especifica o padrão para identificar o início de uma mensagem de log. Uma mensagem de log é feita de uma linha em conformidade com o padrão e as linhas subsequentes que não correspondem ao padrão.

  Se você omitir esse campo, o modo multilinhas será desabilitado, e qualquer linha que comece com um caractere diferente de espaço fechará a mensagem de log anterior e iniciará uma nova mensagem de log.

  Se incluir esse campo, você poderá especificar `{timestamp_format}` para usar a mesma expressão regular como o formato de time stamp. Caso contrário, você pode especificar

uma expressão regular diferente para o CloudWatch Logs usar para determinar as linhas de entradas de multilinha.

- `encoding`: especifica a codificação do arquivo de log para que o arquivo possa ser lido corretamente. Se você especificar uma codificação incorreta, poderá haver perda de dados porque os caracteres que não podem ser decodificados serão substituídos por outros caracteres.

  O valor padrão é `utf-8`. Os valores a seguir são todos possíveis:

  ```
  ascii, big5, euc-jp, euc-kr, gbk, gb18030, ibm866, iso2022-jp,
  iso8859-2, iso8859-3, iso8859-4, iso8859-5, iso8859-6, iso8859-7,
  iso8859-8, iso8859-8-i, iso8859-10, iso8859-13, iso8859-14,
  iso8859-15, iso8859-16, koi8-r, koi8-u, macintosh, shift_jis, utf-8,
  utf-16, utf-16le, UTF-16, UTF-16LE, windows-874, windows-1250,
  windows-1251, windows-1252, windows-1253, windows-1254,
  windows-1255, windows-1256, windows-1257, windows-1258, x-mac-
  cyrillic
  ```

- A seção `windows_events` especifica os tipos de eventos do Windows a serem coletados de servidores que executam o Windows Server. Isso inclui os seguintes campos:
  - `collect_list`: obrigatório se `windows_events` for incluído. Especifica os tipos e os níveis de eventos do Windows a serem coletados. Cada log a ser coletados tem uma entrada nessa seção, que pode incluir os seguintes campos:
    - `event_name`: especifica os tipos de eventos do Windows a serem registrados em log. Isso é equivalente ao nome do canal de log de eventos do Windows: por exemplo, `System`, `Security`, `Application`, etc. Esse campo é necessário para cada tipo de evento do Windows a ser registrado em log.

      > **Note**
      >
      > Quando o CloudWatch recupera mensagens de um canal de log do Windows, ele procura o canal de log com base na propriedade `Full Name`. Enquanto isso, o painel de navegação do Windows Event Viewer exibe a propriedade `Log Name` dos canais de log. O `Full Name` e o `Log Name` nem sempre são correspondentes. Para confirmar o`Full Name` de um canal, clique com o botão direito do mouse nele no Visualizador de Eventos do Windows e abraProperties (Propriedades).

- `event_levels`: especifica os níveis de evento a serem registrados. Você deve especificar cada nível a ser registrado em log. Os valores possíveis incluem `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` e `VERBOSE`. Esse campo é necessário para cada tipo de evento do Windows a ser registrado em log.
- `log_group_name`: obrigatório. Especifica o que usar como nome de grupo de logs no CloudWatch Logs.
- `log_stream_name`: optional. Especifica o que usar como nome do fluxo de logs no CloudWatch Logs. Como parte do nome, você pode usar `{instance_id}`, `{hostname}`, `{local_hostname}` e `{ip_address}` como variáveis dentro do nome. O `{hostname}` recupera o nome do host de metadados do EC2, e o `{local_hostname}` usa o nome de host do arquivo de configuração de rede.

  Se você omitir esse campo, o valor do parâmetro `log_stream_name` na seção global de `logs` será usado. Se isso também for omitido, o valor padrão de `{instance_id}` será usado.

  Caso um fluxo de logs ainda não exista, ele será criado automaticamente.
- `event_format`: optional. Especifica o formato a ser usado para armazenar eventos do Windows no CloudWatch Logs. O `xml` usa o formato XML como no Visualizador de eventos do Windows. O `text` usa o formato do atendente do CloudWatch Logs herdado.
- `retention_in_days`: optional. Especifica o número de dias em que os eventos do Windows serão retidos no grupo de logs especificado.
  - Se o agente estiver criando esse grupo de logs agora e você omitir esse campo, a retenção desse novo grupo será definida para nunca expirar.
  - Se esse grupo de logs já existir e você especificar esse campo, a nova retenção definida será utilizada. Se você omitir esse campo para um grupo de logs existente, a retenção do grupo de logs não será alterada.

    O assistente do agente do CloudWatch usará `-1` como valor padrão esse campo quando ele for usado para criar o arquivo de configuração do agente e você não especificar um valor para a retenção de logs. Esse valor `-1` definido pelo assistente especifica que os eventos no grupo de logs não expiram. No entanto, editar manualmente esse valor para não `-1` tem efeito.

  Os valores válidos são 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 2192, 2557, 2922, 3288 e 3653..

Se você configurar o atendente para gravar vários fluxos de log no mesmo grupo de logs, especificar o `retention_in_days` em um só lugar definirá a retenção de log para todo o grupo de logs. Se você especificar `retention_in_days` para o mesmo grupo de logs em vários locais, a retenção será definida se todos esses valores forem iguais. No entanto, se diferentes valores de `retention_in_days` forem especificados para o mesmo grupo de logs em vários locais, a retenção de log não será definida e o atendente será interrompido, retornando um erro.

> **Note**
>
> A função do IAM do atendente ou o usuário do IAM deve ter o `logs:PutRetentionPolicy` para poder definir políticas de retenção. Para ter mais informações, consulte Permitir que o atendente do CloudWatch defina a política de retenção de logs.

> **Warning**
>
> Se você definir `retention_in_days` para um grupo de logs que já existe, todos os logs nesse grupo de logs publicados antes do número de dias que você especificar são excluídos. Por exemplo, configurá-lo como 3 faria com que todos os logs de 3 dias atrás e anteriores a isso fossem excluídos.

- `log_stream_name`: obrigatório. Especifica o nome do fluxo de logs padrão a ser usado para todos os logs ou eventos do Windows que não tenham nomes de stream de logs individuais definidos no parâmetro `log_stream_name` em sua entrada na `collect_list`.
- `endpoint_override`: especifica um endpoint FIPS ou um link privado a ser usado como o endpoint onde o atendente envia logs. Especificar esse campo e definir um link privado permite enviar os logs a um endpoint da Amazon VPC. Para obter mais informações, consulte O que é a Amazon VPC?

  O valor de `endpoint_override` deve ser uma string que seja um URL.

  Por exemplo, a parte a seguir da seção de logs do arquivo de configuração define o atendente para usar um VPC endpoint ao enviar logs.

```
{
```

```
  "logs": {
    "endpoint_override": "vpce-XXXXXXXXXXXXXXXXXXXXXXXX.logs.us-
east-1.vpce.amazonaws.com",
   ......
   },
}
```

- `force_flush_interval`: especifica em segundos a quantidade máxima de tempo em que os logs permanecem no buffer da memória antes de serem enviados ao servidor. Não importa a configuração para esse campo, se o tamanho dos logs no buffer alcançar 1 MB, os logs serão enviados imediatamente para o servidor. O valor padrão é 5.

  Se você estiver usando o agente para relatar métricas de alta resolução em formato de métricas incorporadas e estiver configurando alarmes nessas métricas, mantenha esse parâmetro definido com o valor padrão de 5. Caso contrário, as métricas serão relatadas com um atraso que poderá causar alarme em dados parciais ou incompletos.
- `credentials`: especifica uma função do IAM a ser usada ao enviar logs para outra conta da AWS. Se especificado, esse campo contém um parâmetro, `role_arn`.
  - `role_arn`: especifica o ARN de uma função do IAM a ser usada para autenticação ao enviar logs para outra conta da AWS. Para ter mais informações, consulte Envio de métricas, logs e rastreamentos a uma conta diferente. Se especificado aqui, isso substitui o `role_arn` especificado na seção `agent` do arquivo de configuração, se houver.
- `metrics_collected`: este campo pode conter seções que especificam que o agente deve coletar logs para habilitar casos de uso, como o CloudWatch Application Signals e o Container Insights, com uma observabilidade aprimorada para o Amazon EKS.
  - `app_signals` (Opcional) Especifica que você deseja habilitar o CloudWatch Application Signals. Para obter mais informações sobre essa configuração, consulte Habilitar o CloudWatch Application Signals.
  - `kubernetes`: esse campo pode conter um parâmetro do `enhanced_container_insights`, que você pode usar para ativar o Container Insights com observabilidade aprimorada para o Amazon EKS.
    - `enhanced_container_insights`: defina isso como `true` para ativar o Container Insights com observabilidade aprimorada para o Amazon EKS. Para ter mais informações, consulte Container Insights com observabilidade aprimorada para o Amazon EKS.
    - `accelerated_compute_metrics` – Defina isso como `false` para optar por não coletar métricas de GPU Nvidia nos clusters Amazon EKS. Para ter mais informações, consulte Métricas da GPU NVIDIA.

- `emf` – Para coletar métricas incorporadas em logs, não é mais necessário adicionar esse campo `emf`. um campo herdado que especifica que o atendente deve coletar logs que estão em formato de métrica incorporada. É possível gerar dados de métrica a partir desses logs. Para ter mais informações, consulte Incorporação de métricas em logs.

A seguir, temos um exemplo de uma seção `logs`.

```
"logs":
   {
       "logs_collected": {
           "files": {
               "collect_list": [
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\Logs\\amazon-cloudwatch-agent.log",
                       "log_group_name": "amazon-cloudwatch-agent.log",
                       "log_stream_name": "my_log_stream_name_1",
                       "timestamp_format": "%H: %M: %S%y%b%-d"
                   },
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\Logs\\test.log",
                       "log_group_name": "test.log",
                       "log_stream_name": "my_log_stream_name_2"
                   }
               ]
           },
           "windows_events": {
               "collect_list": [
                   {
                       "event_name": "System",
                       "event_levels": [
                           "INFORMATION",
                           "ERROR"
                       ],
                       "log_group_name": "System",
                       "log_stream_name": "System"
                   },
                   {
                       "event_name": "CustomizedName",
                       "event_levels": [
                           "INFORMATION",
```

```
                        "ERROR"
                    ],
                    "log_group_name": "CustomizedLogGroup",
                    "log_stream_name": "CustomizedLogStream"
                }
            ]
        }
    },
    "log_stream_name": "my_log_stream_name",
    "metrics_collected": {
        "kubernetes": {
            "enhanced_container_insights": true
    }
  }
}
```

Arquivo de configuração do agente do CloudWatch: seção de rastreamentos

Ao adicionar uma seção `traces` ao arquivo de configuração do agente do CloudWatch, é possível habilitar o CloudWatch Application Signals ou coletar rastreamentos do X-Ray e do SDK de instrumentação do OpenTelemetry e enviá-los para o X-Ray.

> **Important**
>
> O perfil do IAM ou o usuário do IAM do agente devem ter a política AWSXrayWriteOnlyAccess para realizar o envio de dados de rastreamento ao X-Ray. Para ter mais informações, consulte Criar funções e usuários do IAM para uso com o atendente do CloudWatch.

Para começar rapidamente a coletar rastreamentos, é possível adicionar apenas o seguinte ao arquivo de configuração do agente do CloudWatch.

```
"traces_collected": {
        "xray": {
        },
        "otlp": {
        }
    }
```

Se você adicionar a seção anterior ao arquivo de configuração do agente do CloudWatch e reiniciar o agente, isso fará com que o agente comece a coletar rastreamentos usando as opções e valores padrão a seguir. Para obter mais informações sobre esses parâmetros, consulte as definições de parâmetros a seguir nesta seção.

```
"traces_collected": {
        "xray": {
            "bind_address": "127.0.0.1:2000",
            "tcp_proxy": {
              "bind_address": "127.0.0.1:2000"
            }
        },
        "otlp": {
            "grpc_endpoint": "127.0.0.1:4317",
            "http_endpoint": "127.0.0.1:4318"
        }
      }
```

A seção `traces` pode incluir os seguintes campos:

- `traces_collected`: obrigatório, se a seção `traces` for incluída. Especifica de quais SDKs coletar rastreamentos. Isso pode incluir os seguintes campos:
  - `app_signals`: optional. Especifica que você deseja habilitar o CloudWatch Application Signals. Para obter mais informações sobre essa configuração, consulte Habilitar o CloudWatch Application Signals.
  - `xray`: optional. Especifica que você deseja coletar rastreamentos do SDK do X-Ray. A seção pode incluir os seguintes campos:
    - `bind_address`: opcional. Especifica o endereço UDP para o agente do CloudWatch usar para escutar rastreamentos do X-Ray. O formato é `ip:port`. Este endereço deve corresponder ao endereço definido no SDK do X-Ray.

      Se você omitir esse campo, o padrão de `127.0.0.1:2000` será usado.
    - `tcp_proxy`: optional. Configura o endereço de um proxy usado para oferecer suporte à amostragem remota do X-Ray. Para obter mais informações, consulte Configuração de regras de amostragem na documentação do X-Ray.

      Essa seção pode conter o campo a seguir.

    - `bind_address`: optional. Especifica o endereço TCP para o qual o agente do CloudWatch deve configurar o proxy. O formato é `ip:port`. Este endereço deve corresponder ao endereço definido no SDK do X-Ray.

      Se você omitir esse campo, o padrão de `127.0.0.1:2000` será usado.

- `otlp`: optional. Especifica que você deseja coletar rastreamentos do SDK do OpenTelemetry. Para obter mais informações sobre o AWS Distro for OpenTelemetry, consulte AWS Distro for OpenTelemetry. Para obter mais informações sobre o AWS Distro para SDKs do OpenTelemetry, consulte a Introdução.

  A seção pode incluir os seguintes campos:

  - `grpc_endpoint`: opcional. Especifica o endereço que o agente do CloudWatch deve usar para escutar rastreamentos de OpenTelemetry enviados usando chamadas de procedimento remoto gRPC. O formato é `ip:port`. Esse endereço deve corresponder ao endereço definido para o exportador de gRPC no SDK do OpenTelemetry.

    Se você omitir esse campo, o padrão de `127.0.0.1:4317` será usado.

  - `http_endpoint`: optional. Especifica o endereço para o agente do CloudWatch usar para escutar rastreamentos de OTLP enviados por HTTP. O formato é `ip:port`. Esse endereço deve corresponder ao endereço definido para o exportador de HTTP no SDK do OpenTelemetry.

    Se você omitir esse campo, o padrão de `127.0.0.1:4318` será usado.

- `concurrency`: optional. Especifica o número máximo de chamadas simultâneas para o X-Ray que podem ser usadas para carregar rastreamentos. O valor padrão é 8
- `local_mode`: optional. Se `true`, o agente não coletará metadados da instância do Amazon EC2. O padrão é `false`
- `endpoint_override`: optional. Especifica um endpoint de FIPS ou um link privado a ser usado como o endpoint onde o agente do CloudWatch envia rastreamentos. Especificar esse campo e definir um link privado permite enviar os rastreamentos a um endpoint da Amazon VPC. Para obter mais informações, consulte O que é a Amazon VPC

  O valor de `endpoint_override` deve ser uma string que seja um URL.
- `region_override`: optional. Especifica a região a usar para o endpoint do X-Ray. O agente do CloudWatch envia os rastreamentos para o X-Ray na região especificada. Se você omitir esse

campo, o agente enviará os rastreamentos para a região onde a instância do Amazon EC2 está localizada.

Se você especificar uma região aqui, ela terá precedência sobre a configuração do parâmetro `region` na seção `agent` do arquivo de configuração.

- `proxy_override`: optional. Especifica o endereço do servidor proxy para o agente do CloudWatch usar ao enviar solicitações para o X-Ray. O protocolo do servidor proxy deve ser especificado como parte desse endereço.
- `credentials`: especifica um perfil do IAM a ser usado ao enviar rastreamentos para outra conta da AWS. Se especificado, esse campo contém um parâmetro, `role_arn`.
  - `role_arn`: especifica o ARN de um perfil do IAM a ser usado para autenticação ao enviar rastreamentos para outra conta da AWS. Para ter mais informações, consulte Envio de métricas, logs e rastreamentos a uma conta diferente. Se especificado aqui, isso substitui o `role_arn` especificado na seção `agent` do arquivo de configuração, se houver.

Arquivo de configuração do atendente do CloudWatch: exemplos completos

A seguir, veja um exemplo de um arquivo de configuração do atendente completo do CloudWatch para um servidor Linux.

Os itens listados nas seções `measurement` para as métricas que você deseja coletar podem especificar o nome completo da métrica, ou apenas a parte do nome da métrica que será acrescentada ao tipo de recurso. Por exemplo, especificar `reads` ou `diskio_reads` na seção `measurement` da seção `diskio` fará com que a métrica `diskio_reads` seja coletada.

Este exemplo inclui as duas maneiras de especificar métricas na seção `measurement`.

```
{
  "agent": {
    "metrics_collection_interval": 10,
    "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log"
  },
  "metrics": {
    "namespace": "MyCustomNamespace",
    "metrics_collected": {
      "cpu": {
        "resources": [
          "*"
        ],
```

```
            "measurement": [
              {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit":
"Percent"},
              {"name": "cpu_usage_nice", "unit": "Percent"},
              "cpu_usage_guest"
            ],
            "totalcpu": false,
            "metrics_collection_interval": 10,
            "append_dimensions": {
              "customized_dimension_key_1": "customized_dimension_value_1",
              "customized_dimension_key_2": "customized_dimension_value_2"
            }
          },
          "disk": {
            "resources": [
              "/",
              "/tmp"
            ],
            "measurement": [
              {"name": "free", "rename": "DISK_FREE", "unit": "Gigabytes"},
              "total",
              "used"
            ],
             "ignore_file_system_types": [
              "sysfs", "devtmpfs"
            ],
            "metrics_collection_interval": 60,
            "append_dimensions": {
              "customized_dimension_key_3": "customized_dimension_value_3",
              "customized_dimension_key_4": "customized_dimension_value_4"
            }
          },
          "diskio": {
            "resources": [
              "*"
            ],
            "measurement": [
              "reads",
              "writes",
              "read_time",
              "write_time",
              "io_time"
            ],
            "metrics_collection_interval": 60
```

```
      },
      "swap": {
        "measurement": [
          "swap_used",
          "swap_free",
          "swap_used_percent"
        ]
      },
      "mem": {
        "measurement": [
          "mem_used",
          "mem_cached",
          "mem_total"
        ],
        "metrics_collection_interval": 1
      },
      "net": {
        "resources": [
          "eth0"
        ],
        "measurement": [
          "bytes_sent",
          "bytes_recv",
          "drop_in",
          "drop_out"
        ]
      },
      "netstat": {
        "measurement": [
          "tcp_established",
          "tcp_syn_sent",
          "tcp_close"
        ],
        "metrics_collection_interval": 60
      },
      "processes": {
        "measurement": [
          "running",
          "sleeping",
          "dead"
        ]
      }
    },
    "append_dimensions": {
```

```
          "ImageId": "${aws:ImageId}",
          "InstanceId": "${aws:InstanceId}",
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"],
 ["d1"],[]],
        "force_flush_interval" : 30
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "log_stream_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
              },
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
                "log_group_name": "test.log",
                "log_stream_name": "test.log",
                "timezone": "Local"
              }
            ]
          }
        },
        "log_stream_name": "my_log_stream_name",
        "force_flush_interval" : 15,
        "metrics_collected": {
           "kubernetes": {
                "enhanced_container_insights": true
      }
    }
  }
}
```

A seguir, temos um exemplo de arquivo de configuração do atendente completo do CloudWatch para um servidor que executa o Windows Server.

```
{
```

```
      "agent": {
        "metrics_collection_interval": 60,
        "logfile": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\\amazon-
cloudwatch-agent.log"
      },
      "metrics": {
        "namespace": "MyCustomNamespace",
        "metrics_collected": {
          "Processor": {
            "measurement": [
              {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
              "% Interrupt Time",
              "% User Time",
              "% Processor Time"
            ],
            "resources": [
              "*"
            ],
            "append_dimensions": {
              "customized_dimension_key_1": "customized_dimension_value_1",
              "customized_dimension_key_2": "customized_dimension_value_2"
            }
          },
          "LogicalDisk": {
            "measurement": [
              {"name": "% Idle Time", "unit": "Percent"},
              {"name": "% Disk Read Time", "rename": "DISK_READ"},
              "% Disk Write Time"
            ],
            "resources": [
              "*"
            ]
          },
          "customizedObjectName": {
            "metrics_collection_interval": 60,
            "customizedCounterName": [
              "metric1",
              "metric2"
            ],
            "resources": [
              "customizedInstances"
            ]
          },
          "Memory": {
```

```
            "metrics_collection_interval": 5,
            "measurement": [
              "Available Bytes",
              "Cache Faults/sec",
              "Page Faults/sec",
              "Pages/sec"
            ]
          },
          "Network Interface": {
            "metrics_collection_interval": 5,
            "measurement": [
              "Bytes Received/sec",
              "Bytes Sent/sec",
              "Packets Received/sec",
              "Packets Sent/sec"
            ],
            "resources": [
              "*"
            ],
            "append_dimensions": {
              "customized_dimension_key_3": "customized_dimension_value_3"
            }
          },
          "System": {
            "measurement": [
              "Context Switches/sec",
              "System Calls/sec",
              "Processor Queue Length"
            ]
          }
        },
        "append_dimensions": {
          "ImageId": "${aws:ImageId}",
          "InstanceId": "${aws:InstanceId}",
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"],
["d1"],[]]
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
```

```
            {
              "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\amazon-cloudwatch-agent.log",
              "log_group_name": "amazon-cloudwatch-agent.log",
              "timezone": "UTC"
            },
            {
              "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\test.log",
              "log_group_name": "test.log",
              "timezone": "Local"
            }
          ]
        },
        "windows_events": {
          "collect_list": [
            {
              "event_name": "System",
              "event_levels": [
                "INFORMATION",
                "ERROR"
              ],
              "log_group_name": "System",
              "log_stream_name": "System",
              "event_format": "xml"
            },
            {
              "event_name": "CustomizedName",
              "event_levels": [
                "WARNING",
                "ERROR"
              ],
              "log_group_name": "CustomizedLogGroup",
              "log_stream_name": "CustomizedLogStream",
              "event_format": "xml"
            }
          ]
        }
      },
      "log_stream_name": "example_log_stream_name"
    }
  }
```

Salvar o arquivo de configuração do atendente do CloudWatch manualmente

Se criar ou editar o arquivo de configuração do atendente do CloudWatch manualmente, você poderá atribuir qualquer nome a ele. Para simplificar a solução de problemas, recomendamos que você nomeie-o `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` em um servidor Linux e `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json` nos servidores que executam o Windows Server. Depois de criar o arquivo, copie-o em outros servidores em que você deseja executar o atendente.

Carregar o arquivo de configuração do atendente do CloudWatch no Systems Manager Parameter Store

Se você planeja usar o SSM Agent para instalar o atendente do CloudWatch em servidores, depois de editar manualmente o arquivo de configuração do atendente do CloudWatch, carregue-o para o Systems Manager Parameter Store. Para fazer isso, use o comando `put-parameter` do Systems Manager.

Para poder armazenar o arquivo no Parameter Store, você deve usar uma função do IAM com permissões suficientes. Para ter mais informações, consulte Criar funções e usuários do IAM para uso com o atendente do CloudWatch.

Use o comando a seguir, em que o *`nome do parâmetro`* é o nome a ser usado para esse arquivo no Parameter Store e *`configuration_file_pathname`* é o caminho e o nome do arquivo de configuração que você editou.

```
aws ssm put-parameter --name "parameter name" --type "String" --value
 file://configuration_file_pathname
```

Habilitar o CloudWatch Application Signals

Use o CloudWatch Application Signals para instrumentar as aplicações de forma automática na AWS para que você possa acompanhar a performance das aplicações em relação aos seus objetivos de negócios. O Application Signals fornece uma visualização unificada e centrada em aplicações para as aplicações em Java, as dependências e as bordas. Para ter mais informações, consulte Application Signals.

O CloudWatch Application Signals utiliza o agente do CloudWatch para receber métricas e rastreamentos das aplicações instrumentadas de forma automática, aplicar regras para reduzir a

alta cardinalidade, como opção, e, em seguida, publicar a telemetria processada no CloudWatch. É possível fornecer uma configuração personalizada ao agente do CloudWatch, especificamente para o Application Signals, ao usar o arquivo de configuração do agente. Para começar, a existência de uma seção `app_signals` na seção `metrics_collected` dentro da seção `logs` do arquivo de configuração do agente especifica que o agente do CloudWatch receberá métricas das aplicações instrumentadas de forma automática. De maneira semelhante, a existência de uma seção `app_signals` na seção `traces_collected` dentro da seção `traces` do arquivo de configuração do agente especifica que o agente do CloudWatch está habilitado para receber rastreamentos das aplicações instrumentadas de forma automática. Além disso, como opção, é possível aprovar as regras de configuração personalizadas para reduzir a publicação de telemetria de alta cardinalidade, conforme descrito nesta seção.

- Para os clusters do Amazon EKS, quando você instala o complemento Amazon CloudWatch Observability do EKS, por padrão, o agente do CloudWatch é habilitado para receber métricas e rastreamentos das aplicações instrumentadas de forma automática. Caso deseje, opcionalmente, aprovar regras de configuração personalizadas, você poderá fazê-lo ao aprovar uma configuração do agente personalizada para o complemento do Amazon EKS ao criá-lo ou atualizá-lo usando a configuração adicional, conforme descrito em (Opcional) Configuração adicional.
- Para as outras plataformas compatíveis, incluindo o Amazon EC2, é necessário iniciar o agente do CloudWatch com uma configuração do agente que habilite o Application Signals ao especificar as seções `app_signals` e, opcionalmente, quaisquer regras de configuração personalizadas, conforme descrito posteriormente nesta seção.

Veja a seguir uma visão geral dos campos que estão relacionados ao CloudWatch Application Signals no arquivo de configuração do agente do CloudWatch.

- `logs`
  - `metrics_collected`: este campo pode conter seções que especificam que o agente deve coletar logs para habilitar casos de uso, como o CloudWatch Application Signals e o Container Insights, com uma observabilidade aprimorada para o Amazon EKS.

> **Note**
>
> Anteriormente, essa seção também era usada para especificar que o agente deveria coletar logs que estivessem no formato de métrica incorporada. Essas configurações não são mais necessárias.

- `app_signals` (Opcional) Especifica que você deseja habilitar que o CloudWatch Application Signals receba métricas das aplicações instrumentadas de forma automática para viabilizar o CloudWatch Application Signals.
  - `rules` (Opcional) Uma matriz de regras para selecionar condicionalmente as métricas e os rastreamentos e aplicar ações para tratar cenários de alta cardinalidade. Cada regra pode conter os seguintes campos:
    - `rule_name` (Opcional) O nome da regra.
    - `selectors` (Opcional) Uma matriz de agentes de correspondências de dimensão para as métricas e para os rastreamentos. Cada seletor deve fornecer os seguintes campos:
      - `dimension`: obrigatório, se `selectors` não for um campo vazio. Esse campo especifica a dimensão para as métricas e para os rastreamentos a serem usados como filtros.
      - `match`: obrigatório, se `selectors` não for um campo vazio. Um padrão curinga usado para realizar a correspondência de valores da dimensão especificada.
    - `action` (Opcional) A ação a ser aplicada às métricas e aos rastreamentos que correspondem aos seletores especificados. O valor de `action` deve ser uma das seguintes palavras-chave:
      - `keep`: especifica somente o envio de métricas e de rastreamentos para o CloudWatch, se eles forem correspondidos pelos `selectors`.
      - `drop`: especifica o descarte da métrica e dos rastreamentos que correspondem aos `selectors`.
      - `replace`: especifica a substituição das dimensões das métricas e dos rastreamentos que correspondem aos `selectors`. A substituição ocorre de acordo com a seção `replacements`.
    - `replacements` Necessário se `action` for `replace`. Uma matriz de pares de dimensões e valores que serão aplicados às métricas e aos rastreamentos que correspondem aos `selectors` especificados quando a `action` for `replace`. Cada substituição deve fornecer os seguintes campos:
      - `target_dimension`: obrigatório, se `replacements` não for um campo vazio. Especifica a dimensão que precisa ser substituída.
      - `value`: obrigatório, se `replacements` não for um campo vazio. O valor que substituirá o valor original por `target_dimension`.

- `limiter` (Opcional) Use esta seção para limitar quantas métricas e dimensões o Application Signals enviará ao CloudWatch com a finalidade de otimizar seus custos.
  - `disabled` (Opcional) Se `true`, o recurso de limitação de métricas estará desabilitado. O padrão é `false`
  - `drop_threshold` (Opcional) O número máximo de métricas distintas por serviço em um intervalo de rotação que podem ser exportadas por um agente do CloudWatch. O padrão é 500.
  - `rotation_interval` (Opcional) O intervalo no qual o limitador redefine os registros de métricas para contagem de distinção. O intervalo é expresso como uma string com uma sequência de números e um sufixo de unidade. As frações são compatíveis. Os sufixos de unidades compatíveis são `s`, `m`, `h`, `ms`, `us` e `ns`.

    O padrão é `1h` para uma hora.
  - `log_dropped_metrics` (Opcional) Especifica se o agente deve gravar os registros em log nos logs do agente do CloudWatch quando as métricas do Application Signals são descartadas. O padrão é `false`.

    > **Note**
    >
    > Para ativar esse registro log, o parâmetro `debug` na seção `agent` também deve ser configurado como `true`.

- `traces`
  - `traces_collected`
    - `app_signals`: opcional. Especifique isso para habilitar que o agente do CloudWatch receba rastreamentos das aplicações instrumentadas de forma automática para viabilizar o CloudWatch Application Signals.

> **Note**
>
> Embora as regras `app_signals` personalizadas sejam especificadas na seção `metrics_collected`, que está contida na seção `logs`, elas também se aplicam de forma implícita à seção `traces_collected`. O mesmo conjunto de regras se aplicará às métricas e aos rastreamentos.

Quando existem múltiplas regras com ações diferentes, elas se aplicam na seguinte sequência: `keep`, `drop` e `replace`.

Veja a seguir um exemplo de um arquivo de configuração completo para o agente do CloudWatch que aplica as regras personalizadas.

```
{
  "logs": {
    "metrics_collected": {
      "app_signals": {
        "rules": [
          {
            "rule_name": "keep01",
            "selectors": [
              {
                "dimension": "Service",
                "match": "pet-clinic-frontend"
              },
              {
                "dimension": "RemoteService",
                "match": "customers-service"
              }
            ],
            "action": "keep"
          },
          {
            "rule_name": "drop01",
            "selectors": [
              {
                "dimension": "Operation",
                "match": "GET /api/customer/owners/*"
              }
            ],
            "action": "drop"
          },
          {
            "rule_name": "replace01",
            "selectors": [
              {
                "dimension": "Operation",
                "match": "PUT /api/customer/owners/*/pets/*"
              },
              {
```

```
                "dimension": "RemoteOperation",
                "match": "PUT /owners"
              }
            ],
            "replacements": [
              {
                "target_dimension": "Operation",
                "value": "PUT /api/customer/owners/{ownerId}/pets{petId}"
              }
            ],
            "action": "replace"
          }
        ]
      }
    }
  },
  "traces": {
    "traces_collected": {
      "app_signals": {}
    }
  }
}
```

Para o arquivo de configuração no exemplo anterior, as `rules` são processadas da seguinte forma:

1. A regra `keep01` garante que quaisquer métricas e rastreamentos com a dimensão `Service` como `pet-clinic-frontend` e a dimensão `RemoteService` como `customers-service` sejam mantidos.
2. Para as métricas e os rastreamentos processados após a aplicação de `keep01`, a regra `drop01` garante que as métricas e os rastreamentos com a dimensão `Operation` como `GET /api/customer/owners/*` sejam descartados.
3. Para as métricas e os rastreamentos processados após a aplicação de `drop01`, a regra `replace01` atualiza as métricas e os rastreamentos que têm a dimensão `Operation` como `PUT /api/customer/owners/*/pets/*` e a dimensão `RemoteOperation` como `PUT /owners`, de modo que a dimensão `Operation` passe a ser substituída por `PUT /api/customer/owners/{ownerId}/pets{petId}`.

Veja a seguir um exemplo completo de um arquivo de configuração do CloudWatch que gerencia a cardinalidade no Application Signals ao alterar o limite de métricas para 100, ao habilitar o registro em log de métricas descartadas e ao definir o intervalo de rotação para duas horas.

```
{
    "logs": {
        "metrics_collected": {
            "app_signals": {
                "limiter": {
                    "disabled": false,
                    "drop_threshold": 100,
                    "rotation_interval": "2h",
                    "log_dropped_metrics": true
                }
            }
        },
        "traces": {
            "traces_collected": {
                "app_signals": {}
            }
        }
    }
}
```

Coletar métricas de performance da rede

Instâncias do EC2 em execução no Linux que usam o Elastic Network Adapter (ENA) publicam métricas de performance da rede. A versão 1.246396.0 e posteriores do atendente do CloudWatch permitem importar essas métricas de performance de rede para o CloudWatch. Quando você importa essas métricas de performance de rede para o CloudWatch, elas são cobradas como métricas personalizadas do CloudWatch.

Para obter mais informações sobre o driver do ENA, consulte Habilitar redes avançadas com o Elastic Network Adapter (ENA) em instâncias do Linux e Habilitar redes avançadas com o Elastic Network Adapter (ENA) em instâncias do Windows.

A forma de configurar a coleção de métricas de performance de redes é diferente nos servidores Linux e Windows.

A tabela a seguir lista essas métricas de performance de rede habilitadas pelo adaptador ENA. Quando o atendente do CloudWatch importa essas métricas para o CloudWatch a partir de instâncias do Linux, ele inroduz `ethtool_` no início de cada um desses nomes da métrica.

Métrica	Descrição
Nome em servidores Linux: **bw_in_allowance_exceeded** Nome em servidores Windows: **Aggregate inbound BW allowance exceeded**	Número de pacotes na fila e/ou descartados porque a largura de banda agregada de entrada excedeu o máximo para a instância. Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. Para mais informações, consulte Coletar métricas de performance da rede Unidade: nenhuma
Nome em servidores Linux: **bw_out_allowance_exceeded** Nome em servidores Windows: **Aggregate outbound BW allowance exceeded**	Número de pacotes na fila e/ou descartados porque a largura de banda agregada de saída excedeu o máximo para a instância. Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. Para mais informações, consulte Coletar métricas de performance da rede Unidade: nenhuma
Nome em servidores Linux: **conntrack_allowance_available** Nome em servidores Windows: **Available connection tracking allowance**	Relata o número de conexões rastreadas que podem ser estabelecidas pela instância antes de atingir a cota de conexões rastreadas desse tipo de instância. Esta métrica está disponível somente em instâncias do EC2 baseadas em Nitro usando o driver Linux para o Adaptador de Rede Elástica (ENA) a partir da versão 2.8.1 e em computadores usando o driver do Windows para o Adaptador de Rede Elástica (ENA) a partir da versão 2.6.0. Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_c`

Métrica	Descrição
	`ollected` do arquivo de configuração do atendente do CloudWatch. Para mais informações, consulte Coletar métricas de performance da rede Unidade: nenhuma
Nome em servidores Linux: **`ena_srd_mode`** Nome em servidores Windows: **`ena srd mode`**	Descreve quais recursos do ENA Express estão habilitados. Para obter mais informações sobre o ENA Express, consulte Melhorar a performance da rede com o ENA Express em instâncias Linux. Os valores são os seguintes: • `0` = ENA Express desativado, UDP desativado • `1` = ENA Express ativado, UDP desativado • `2` = ENA Express desativado, UDP ativado **Note** Isso só acontece quando o ENA Express foi originalmente habilitado e o UDP foi configurado para usá-lo. O valor anterior é retido para tráfego UDP. • `3` = ENA Express ativado, UDP ativado

Métrica	Descrição
Nome em servidores Linux: **ena_srd_eligible_tx_pkts** Nome em servidores Windows: **ena srd eligible tx pkts**	O número de pacotes de rede enviados em um determinado período que atendem aos requisitos de elegibilidade do Scalable Reliable Datagram (SRD) da AWS, como se segue: • Os tipos de instâncias de envio e de recebimento são compatíveis. • As instâncias de envio e de recebimento devem ter o ENA Express configurado. • As instâncias de envio e recebimento devem estar na mesma sub-rede. • O caminho da rede entre as instâncias não deve incluir caixas de middleware. No momento, o ENA Express não é compatível com caixas de middleware.
Nome em servidores Linux: **ena_srd_tx_pkts** Nome em servidores Windows: **ena srd tx pkts**	O número de pacotes de SRD transmitidos em um determinado período.
Nome em servidores Linux: **ena_srd_rx_pkts** Nome em servidores Windows: **ena srd rx pkts**	O número de pacotes de SRD recebidos em um determinado período.
Nome em servidores Linux: **ena_srd_resource_utilization** Nome em servidores Windows: **ena srd resource utilization**	A porcentagem da utilização da memória máxima permitida para conexões por SRD simultâneas que a instância consumiu.

Métrica	Descrição
Nome em servidores Linux: **`linklocal_allowance_exceeded`** Nome em servidores Windows: **`Link local packet rate allowance exceeded`**	Número de pacotes descartados porque o PPS do tráfego para os serviços de proxy local excedeu o máximo para a interface da rede. Isso afeta o tráfego para o serviço de DNS, o Instance Metadata Service e o Amazon Time Sync Service. Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. Para mais informações, consulte Coletar métricas de performance da rede Unidade: nenhuma
Nome em servidores Linux: **`linklocal_allowance_exceeded`** Nome em servidores Windows: **`Link local packet rate allowance exceeded`**	Número de pacotes descartados porque o PPS do tráfego para os serviços de proxy local excedeu o máximo para a interface da rede. Isso afeta o tráfego para o serviço de DNS, o Instance Metadata Service e o Amazon Time Sync Service. Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. Para mais informações, consulte Coletar métricas de performance da rede Unidade: nenhuma

Métrica	Descrição
Nome em servidores Linux: **pps_allowance_exceeded** Nome em servidores Windows: **PPS allowance exceeded**	Número de pacotes na fila e/ou descartados porque o PPS bidirecional excedeu o máximo para a instância. Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. Para mais informações, consulte Coletar métricas de performance da rede Unidade: nenhuma

Configuração do Linux

Em servidores Linux, o plugin ethtool permite importar as métricas de performance de rede para o CloudWatch.

O ethtool é um utilitário Linux padrão que pode coletar estatísticas sobre dispositivos Ethernet em servidores Linux. As estatísticas coletadas dependem do dispositivo de rede e do driver. Exemplos dessas estatísticas incluem `tx_cnt`, `rx_bytes`, `tx_errors` e `align_errors`. Ao usar o plugin ethtool com o atendente do CloudWatch, também é possível importar essas estatísticas para o CloudWatch, juntamente com as métricas de performance de rede do EC2 listadas anteriormente nesta seção.

> **Tip**
>
> Para encontrar as estatísticas disponíveis em nosso sistema operacional e dispositivo de rede, use o comando `ethtool -S`.

Quando o atendente do CloudWatch importa métricas para o CloudWatch, ele adiciona um prefixo `ethtool_` para os nomes de todas as métricas importadas. Assim, a estatística ethtool padrão `rx_bytes` é chamada `ethtool_rx_bytes` no CloudWatch, e a métrica de performance de rede do EC2 `bw_in_allowance_exceeded` é chamado `ethtool_bw_in_allowance_exceeded` no CloudWatch.

Em servidores Linux, para importar métricas do ethtool, adicione uma seção `ethtool` à seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. A seção `ethtool` pode incluir as seguintes subseções:

- interface_include: incluir essa seção faz com que o atendente colete métricas somente das interfaces cujo nome está listado nessa seção. Se você omitir essa seção, as métricas serão coletadas de todas as interfaces ethernet que não estão listadas em `interface_exclude`.

  A interface ethernet padrão é `eth0`.
- interface_exclude: se você incluir essa seção, liste as interfaces ethernet cujas métricas não deseja coletar.

  O plugin ethtool sempre ignora interfaces de loopback.
- metrics_include: essa seção lista as métricas a serem importadas para o CloudWatch. Pode incluir estatísticas padrão coletadas pelo ethtool e métricas de rede de alta resolução do Amazon EC2.

O exemplo a seguir exibe parte do arquivo de configuração do atendente do CloudWatch. Essa configuração coleta as métricas padrão do ethtool `rx_packets` e `tx_packets`, e as métricas de performance de rede do Amazon EC2 apenas da interface `eth1`.

Para obter mais informações sobre o arquivo de configuração do atendente do CloudWatch, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

```
"metrics": {
    "append_dimensions": {
      "InstanceId": "${aws:InstanceId}"
    },
    "metrics_collected": {
      "ethtool": {
        "interface_include": [
          "eth1"
        ],
        "metrics_include": [
          "rx_packets",
          "tx_packets",
          "bw_in_allowance_exceeded",
          "bw_out_allowance_exceeded",
          "conntrack_allowance_exceeded",
          "linklocal_allowance_exceeded",
          "pps_allowance_exceeded"
```

```
            ]
        }
    }
}
```

Configuração do Windows

Nos servidores Windows, as métricas de performance da rede estão disponíveis por meio dos contadores de performance do Windows, dos quais o agente CloudWatch já coleta métricas. Assim sendo, você não precisa de um plug-in para coletar essas métricas dos servidores Windows.

Veja a seguir um exemplo de arquivo de configuração para coletar métricas de performance de rede do Windows. Para obter mais informações sobre como editar as configurações no arquivo de configuração do agente do CloudWatch, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

```
{
    "metrics": {
        "append_dimensions": {
            "InstanceId": "${aws:InstanceId}"
        },
        "metrics_collected": {
            "ENA Packets Shaping": {
                "measurement": [
                    "Aggregate inbound BW allowance exceeded",
                    "Aggregate outbound BW allowance exceeded",
                    "Connection tracking allowance exceeded",
                    "Link local packet rate allowance exceeded",
                    "PPS allowance exceeded"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            }
        }
    }
}
```

VIsualizar métricas de performance da rede

Depois de importar métricas de performance de rede para o CloudWatch, é possível visualizar essas métricas como gráficos de séries temporais e criar alarmes que podem observar essas métricas e notificar você, se elas violarem um limite especificado. O procedimento a seguir mostra como visualizar métricas do ethtool como um gráfico de séries temporais. Para obter mais informações sobre configuração de alarmes, consulte Usar alarmes do Amazon CloudWatch.

Como todas essas métricas são contadores agregados, é possível usar funções de matemática métrica do CloudWatch, como `RATE(METRICS())`, para calcular a taxa dessas métricas em gráficos ou usá-las para definir alarmes. Para mais informações sobre funções matemáticas de métrica, consulte Usar matemática de métricas.

Para visualizar métricas de performance de rede no console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Escolha o namespace para as métricas coletadas pelo atendente. Por padrão, é CWAgent, mas você pode ter especificado um namespace diferente no arquivo de configuração do atendente do CloudWatch.
4. Escolha uma dimensão de métrica; por exemplo, Per-Instance Metrics (Métricas por instância).
5. A guia All metrics (Todas as métricas) exibe todas as métricas dessa dimensão no namespace. Você pode fazer o seguinte:

    a. Para criar um gráfico de uma métrica, marque a caixa de seleção ao lado da métrica. Para selecionar todas as métricas, marque a caixa de seleção na linha de cabeçalho da tabela.
    b. Para classificar a tabela, use o cabeçalho da coluna.
    c. Para filtrar por recurso, escolha o ID do recurso e Add to search (Adicionar à pesquisa).
    d. Para filtrar por métrica, escolha o nome da métrica e Add to search (Adicionar à pesquisa).
6. (Opcional) Para adicionar esse gráfico a um painel do CloudWatch, escolha Actions (Ações) e Add to dashboard (Adicionar ao painel).

Colete métricas de GPU NVIDIA

Você pode usar o atendente do CloudWatch para coletar métricas de GPU NVIDIA de servidores Linux. Para configurar, adicione uma seção `nvidia_gpu` à seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. Para ter mais informações, consulte Seção Linux.

Além disso, a instância deve ter um driver NVIDIA instalado. Os drivers NVIDIA estão pré-instalados em algumas imagens de máquina da Amazon (AMIs). Caso contrário, é possível instalar o driver manualmente. Para obter mais informações, consulte Instalação de drivers NVIDIA em instâncias Linux.

As seguintes métricas podem ser coletadas. Todas essas métricas são coletadas sem uma `Unit` do CloudWatch, mas você pode especificar uma unidade para cada métrica adicionando um parâmetro ao arquivo de configuração do atendente CloudWatch. Para ter mais informações, consulte Seção Linux.

Métrica	Nome da métrica no CloudWatch	Descrição
`utilization_gpu`	`nvidia_smi_utilization_gpu`	A porcentagem de tempo do período amostral anterior durante a qual um ou mais kernals na GPU estavam sendo executados.
`temperature_gpu`	`nvidia_smi_temperature_gpu`	A temperatura principal da GPU em graus Celsius.
`power_draw`	`nvidia_smi_power_draw`	O último consumo de energia medido para toda a placa, em watts.
`utilization_memory`	`nvidia_smi_utilization_memory`	A porcentagem de tempo do período de amostra anterior durante a qual a memória global (dispositivo) estava sendo lida ou gravada.
`fan_speed`	`nvidia_smi_fan_speed`	A porcentagem da velocidade máxima do ventilador em que o ventilador do dispositivo deve funcionar atualmente.
`memory_total`	`nvidia_smi_memory_total`	Memória total reportada, em MB.
`memory_used`	`nvidia_smi_memory_used`	Memória utilizada, em MB.
`memory_free`	`nvidia_smi_memory_free`	Memória livre, em MB.

Métrica	Nome da métrica no CloudWatch	Descrição
pcie_link_gen_current	nvidia_smi_pcie_link_gen_current	A geração de links atual.
pcie_link_width_current	nvidia_smi_pcie_link_width_current	A largura do link atual.
encoder_stats_session_count	nvidia_smi_encoder_stats_session_count	Número atual de sessões de codificador.
encoder_stats_average_fps	nvidia_smi_encoder_stats_average_fps	A média móvel dos quadros de codificação por segundo.
encoder_stats_average_latency	nvidia_smi_encoder_stats_average_latency	A média móvel da latência de codificação em microssegundos.
clocks_current_graphics	nvidia_smi_clocks_current_graphics	A frequência atual do relógio gráfico (sombreador).
clocks_current_sm	nvidia_smi_clocks_current_sm	A frequência atual do relógio Streaming Multiprocessor (SM – Multiprocessador de transmissão).
clocks_current_memory	nvidia_smi_clocks_current_memory	A frequência atual do relógio de memória.

Métrica	Nome da métrica no CloudWatch	Descrição
`clocks_current_video`	`nvidia_smi_clocks_current_video`	A frequência atual dos relógios de vídeo (codificador e decodificador).

Todas essas métricas são coletadas com as seguintes dimensões:

Dimensão	Descrição
`index`	Um identificador exclusivo da GPU neste servidor. Representa o índice NVIDIA Management Library (NVML – Biblioteca de gerenciamento NVIDIA) do dispositivo.
`name`	O tipo de GPU. Por exemplo, `NVIDIA Tesla A100`
`host`	Nome do host do servidor.

Coletar métricas de processo com o plugin procstat

O plugin procstat permite coletar métricas de processos individuais. É compatível com servidores Linux e com servidores que executam versões compatíveis do Windows Server.

Tópicos

Configurar o atendente do CloudWatch para procstat

Para usar o plugin procstat, adicione uma seção `procstat` à seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. Existem três maneiras de especificar os processos a serem monitorados. Use apenas um desses métodos, embora possa usar esse método para especificar um ou mais processos a serem monitorados.

- `pid_file`: seleciona processos pelos nomes dos arquivos de Process Identification Number (PID – Número de identificação do processo) criados.
- `exe`: seleciona os processos que tenham nomes de processos correspondentes à string especificada usando regras de correspondência de expressão regular. A correspondência é uma correspondência do tipo “contém”, o que significa que, se você especificar `agent` como o termo a ser correspondido, os processos com nomes como `cloudwatchagent` corresponderão ao termo. Para obter mais informações, consulte Sintaxe.
- `pattern`: seleciona processos pelas linhas de comando usadas para iniciar os processos. Todos os processos são selecionados que tenham linhas de comando correspondentes à string especificada usando regras de correspondência de expressão regular. Toda a linha de comando é verificada, inclusive opções e parâmetros usados com o comando.

  A correspondência é uma correspondência do tipo “contém”, o que significa que, se você especificar `-c` como o termo a ser correspondido, os processos com parâmetros como `-config` corresponderão ao termo.
- `drop_original_metrics`: optional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.

O atendente do CloudWatch só usa um desses métodos, mesmo se você incluir mais de uma das seções acima. Se você especificar mais de uma seção, o atendente do CloudWatch usará a seção `pid_file` se estiver presente. Do contrário, ele usa a seção `exe`.

Em servidores Linux, as strings especificadas em uma seção `pattern` ou `exe` são avaliadas como expressões regulares. Em servidores nos quais o Windows Server esteja em execução, essas strings são avaliadas como consultas WMI. Um exemplo seria `pattern: "%apache%"`. Para obter mais informações, consulte Operador LIKE.

Independentemente do método, inclua um parâmetro `metrics_collection_interval` opcional, que especifica a frequência em segundos para coletar essas métricas. Se você omitir esse parâmetro, o valor padrão de 60 segundos será usado.

Nos exemplos nas seções a seguir, a seção `procstat` é a única seção incluída na seção `metrics_collected` do arquivo de configuração do atendente. Os arquivos de configuração também podem incluir outras seções em `metrics_collected`. Para ter mais informações, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

Configurar com pid_file

A seção `procstat` do exemplo a seguir monitora os processos que criam os arquivos PID `example1.pid` e `example2.pid`. As métricas diferentes são coletadas de cada processo. As métricas coletadas do processo que cria `example2.pid` são coletadas a cada 10 segundos, e as métricas coletadas do processo `example1.pid` são coletadas a cada 60 segundos, o valor padrão.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "pid_file": "/var/run/example1.pid",
                    "measurement": [
                        "cpu_usage",
                        "memory_rss"
                    ]
                },
                {
                    "pid_file": "/var/run/example2.pid",
                    "measurement": [
                        "read_bytes",
                        "read_count",
```

```
                        "write_bytes"
                    ],
                    "metrics_collection_interval": 10
                }
            ]
        }
    }
}
```

Configurar com exe

A seção `procstat` do exemplo a seguir monitora todos os processos com nomes correspondentes às strings `agent` ou `plugin`. As mesmas métricas são coletadas de cada processo.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "exe": "agent",
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                },
                {
                    "exe": "plugin",
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                }
            ]
        }
    }
}
```

Configurar com padrão

A seção `procstat` do exemplo a seguir monitora todos os processos com linhas de comando correspondentes às strings `config` ou `-c`. As mesmas métricas são coletadas de cada processo.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "pattern": "config",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                },
                {
                    "pattern": "-c",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                }
            ]
        }
    }
}
```

Métricas coletadas pelo procstat

A tabela a seguir lista as métricas que é possível coletar com o plugin `procstat`

O atendente do CloudWatch adiciona `procstat` ao início dos nomes de métrica a seguir. Há uma sintaxe diferente dependendo de ter sido coletada de um servidor Linux ou de um servidor no qual o Windows Server esteja em execução. Por exemplo, a métrica `cpu_time` é exibida como `procstat_cpu_time` quando coletada do Linux e como `procstat cpu_time` quando coletada do Windows Server.

Nome da métrica	Disponível em	Descrição
`cpu_time`	Linux	O tempo em que o processo

Nome da métrica	Disponível em	Descrição
		usa a CPU. Essa métrica é medida em centésimos de segundo. Unidade: Contagem
`cpu_time_guest`	Linux	A quantidade de tempo em que o processo permanece em modo de usuário. Essa métrica é medida em centésimos de segundo. Tipo: float Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
`cpu_time_guest_nice`	Linux	A quantidade de tempo em que o processo está sendo executado em um niced guest. Essa métrica é medida em centésimos de segundo. Tipo: float Unidade: nenhuma
`cpu_time_idle`	Linux	A quantidade de tempo em que o processo permanece em modo ocioso. Essa métrica é medida em centésimos de segundo. Tipo: float Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
`cpu_time_iowait`	Linux	A quantidade de tempo em que o processo está aguardando a conclusão de operações de entrada/saída. Essa métrica é medida em centésimos de segundo. Tipo: float Unidade: nenhuma
`cpu_time_irq`	Linux	A quantidade de tempo em que o processo está atendendo a interrupções. Essa métrica é medida em centésimos de segundo. Tipo: float Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
`cpu_time_nice`	Linux	A quantidade de tempo em que o processo permanece em modo nice. Essa métrica é medida em centésimos de segundo. Tipo: float Unidade: nenhuma
`cpu_time_soft_irq`	Linux	A quantidade de tempo em que o processo está atendendo a interrupções de software. Essa métrica é medida em centésimos de segundo. Tipo: float Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
`cpu_time_steal`	Linux	A quantidade de tempo gasto executando em outros sistemas operacionais quando executado em um ambiente virtualizado. Essa métrica é medida em centésimos de segundo. Tipo: float Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
`cpu_time_stolen`	Linux, Windows Server	A quantidade de tempo em que o processo está em tempo roubado, que é o tempo gasto em outros sistemas operacionais em um ambiente virtualizado. Essa métrica é medida em centésimos de segundo. Tipo: float Unidade: nenhuma
`cpu_time_system`	Linux, Windows Server, macOS	O tempo em que o processo permanece em modo de sistema. Essa métrica é medida em centésimos de segundo. Tipo: float Unidade: Contagem

Nome da métrica	Disponível em	Descrição
`cpu_time_user`	Linux, Windows Server, macOS	O tempo em que o processo permanece em modo de usuário. Essa métrica é medida em centésimos de segundo. Unidade: Contagem
`cpu_usage`	Linux, Windows Server, macOS	A porcentagem de tempo em que o processo permanece ativo em qualquer capacidade. Unidade: Percentual
`memory_data`	Linux, macOS	A quantidade de memória usada pelo processo em dados. Unidade: bytes

Nome da métrica	Disponível em	Descrição
`memory_locked`	Linux, macOS	A quantidade de memória bloqueada pelo processo. Unidade: bytes
`memory_rss`	Linux, Windows Server, macOS	A quantidade de memória real (conjunto residente) que o processo está usando. Unidade: bytes
`memory_stack`	Linux, macOS	A quantidade de memória em pilha usada pelo processo. Unidade: bytes
`memory_swap`	Linux, macOS	A quantidade de memória swap usada pelo processo. Unidade: bytes
`memory_vms`	Linux, Windows Server, macOS	A quantidade de memória virtual usada pelo processo. Unidade: bytes

Nome da métrica	Disponível em	Descrição
`num_fds`	Linux	O número de descritores de arquivo abertos por esse processo. Unidade: nenhuma
`num_threads`	Linux, Windows, macOS	O número de threads neste processo. Unidade: nenhuma
`pid`	Linux, Windows Server, macOS	Process Identifier (ID – Identificador de processo). Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
`pid_count`	Linux, Windows Server, macOS	O número de IDs de processo associados ao processo. Em servidores Linux e computadores macOC, o nome completo dessa métrica é `procstat_lookup_pid_count` e no Windows Server é `procstat_lookup pid_count` . Unidade: nenhuma
`read_bytes`	Linux, Windows Server	O número de bytes lidos de discos pelo processo. Unidade: bytes

Nome da métrica	Disponível em	Descrição
`write_bytes`	Linux, Windows Server	O número de bytes gravados em discos pelo processo. Unidade: bytes
`read_count`	Linux, Windows Server	O número de operações de leitura em disco executadas pelo processo. Unidade: nenhuma
`rlimit_realtime_priority_hard`	Linux	O limite rígido da prioridade em tempo real que pode ser definido para este processo. Unidade: nenhuma
`rlimit_realtime_priority_soft`	Linux	O limite flexível da prioridade em tempo real que pode ser definido para este processo. Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
`rlimit_signals_pending_hard`	Linux	O limite rígido do número máximo de sinais que podem ser enfileirados por este processo. Unidade: nenhuma
`rlimit_signals_pending_soft`	Linux	O limite flexível do número máximo de sinais que podem ser enfileirados por este processo. Unidade: nenhuma
`rlimit_nice_priority_hard`	Linux	O limite rígido da prioridade de nice que pode ser definido para este processo. Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
`rlimit_nice_priority_soft`	Linux	O limite flexível da prioridade de nice que pode ser definido para este processo. Unidade: nenhuma
`rlimit_num_fds_hard`	Linux	O limite rígido para o número máximo de descritores de arquivo que este processo pode ter em aberto. Unidade: nenhuma
`rlimit_num_fds_soft`	Linux	O limite flexível para o número máximo de descritores de arquivo que este processo pode ter em aberto. Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
write_count	Linux, Windows Server	O número de operações de gravação em disco executadas pelo processo. Unidade: nenhuma
involuntary_context_switches	Linux	O número de vezes em que o contexto do processo foi alterado involuntariamente. Unidade: nenhuma
voluntary_context_switches	Linux	O número de vezes em que o contexto do processo foi alterado voluntariamente. Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
`realtime_priority`	Linux	O uso atual da prioridade em tempo real para o processo. Unidade: nenhuma
`nice_priority`	Linux	O uso atual da prioridade boa para o processo. Unidade: nenhuma
`signals_pending`	Linux	O número de sinais pendentes a serem processados pelo processo. Unidade: nenhuma
`rlimit_cpu_time_hard`	Linux	O limite de recursos do tempo de CPU fixo para o processo. Unidade: nenhuma

Nome da métrica	Disponível em	Descrição
`rlimit_cpu_time_soft`	Linux	O limite de recursos do tempo de CPU flexível para o processo. Unidade: nenhuma
`rlimit_file_locks_hard`	Linux	O limite de recursos de bloqueios de arquivo fixo para o processo. Unidade: nenhuma
`rlimit_file_locks_soft`	Linux	O limite de recursos de bloqueios de arquivo flexível para o processo. Unidade: nenhuma
`rlimit_memory_data_hard`	Linux	O limite de recursos fixo no processo de memória usada em dados. Unidade: bytes

Nome da métrica	Disponível em	Descrição
`rlimit_memory_data_soft`	Linux	O limite de recursos flexível no processo de memória usada em dados. Unidade: bytes
`rlimit_memory_locked_hard`	Linux	O limite de recursos fixo no processo de memória bloqueada. Unidade: bytes
`rlimit_memory_locked_soft`	Linux	O limite de recursos flexível no processo de memória bloqueada. Unidade: bytes
`rlimit_memory_rss_hard`	Linux	O limite de recursos fixo no processo de memória física. Unidade: bytes

Nome da métrica	Disponível em	Descrição
`rlimit_memory_rss_soft`	Linux	O limite de recursos flexível no processo de memória física. Unidade: bytes
`rlimit_memory_stack_hard`	Linux	O limite de recursos fixo na pilha de processos. Unidade: bytes
`rlimit_memory_stack_soft`	Linux	O limite de recursos flexível na pilha de processos. Unidade: bytes
`rlimit_memory_vms_hard`	Linux	O limite de recursos fixo no processo de memória virtual. Unidade: bytes

Nome da métrica	Disponível em	Descrição
`rlimit_memory_vms_soft`	Linux	O limite de recursos flexível no processo de memória virtual. Unidade: bytes

Visualizar métricas de processo importadas pelo atendente do CloudWatch

Depois de importar métricas de processo para o CloudWatch, é possível visualizar essas métricas como gráficos de séries temporais e criar alarmes que podem observar essas métricas e notificar você, se elas violarem um limite especificado. O procedimento a seguir mostra como visualizar métricas de processo como um gráfico de séries temporais. Para obter mais informações sobre configuração de alarmes, consulte Usar alarmes do Amazon CloudWatch.

Para exibir métricas de processo no console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Escolha o namespace para as métricas coletadas pelo atendente. Por padrão, é CWAgent, mas você pode ter especificado um namespace diferente no arquivo de configuração do atendente do CloudWatch.
4. Escolha uma dimensão de métrica; por exemplo, Per-Instance Metrics (Métricas por instância).
5. A guia All metrics (Todas as métricas) exibe todas as métricas dessa dimensão no namespace. Você pode fazer o seguinte:

    a. Para criar um gráfico de uma métrica, marque a caixa de seleção ao lado da métrica. Para selecionar todas as métricas, marque a caixa de seleção na linha de cabeçalho da tabela.

    b. Para classificar a tabela, use o cabeçalho da coluna.

    c. Para filtrar por recurso, escolha o ID do recurso e, em seguida, escolha Adicionar à pesquisa.

   d. Para filtrar por métrica, selecione o nome da métrica e, em seguida, escolha Adicionar à pesquisa.

6. (Opcional) Para adicionar esse gráfico a um painel do CloudWatch, escolha Actions (Ações), Add to dashboard (Adicionar ao painel).

Recuperar métricas personalizadas com o StatsD

É possível recuperar métricas personalizadas de suas aplicações ou seus serviços usando o atendente do CloudWatch com o protocolo `StatsD`. StatsD é uma solução de código aberto popular que pode coletar métricas de uma ampla variedade de aplicações. O StatsD é especialmente útil para instrumentar suas próprias métricas. Para obter um exemplo de uso do atendente do CloudWatch e do StatsD juntos, consulte Como melhorar o monitoramento de suas métricas de aplicação personalizadas usando o atendente do Amazon CloudWatch.

O `StatsD` é compatível em servidores Linux e servidores em que o Windows Server esteja em execução. O CloudWatch oferece suporte ao seguinte formato do `StatsD`:

```
MetricName:value|type|@sample_rate|#tag1:
  value,tag1...
```

- `MetricName`: uma string sem dois pontos, barras, caracteres # ou caracteres @.
- `value`: pode ser inteiro ou flutuante.
- `type`: especifique `c` para contador, `g` para indicador, `ms` para temporizador, `h` para histograma ou `s` para conjunto.
- `sample_rate`: (opcional) um flutuante entre 0 e 1, inclusive. Use somente para métricas de contador, histograma e temporizador. O valor padrão é 1 (amostragem de 100% do tempo).
- `tags`: (opcional) uma lista separada por vírgulas das etiquetas. As etiquetas do `StatsD` são semelhantes às dimensões no CloudWatch. Use dois pontos para tags de chave/valor, como `env:prod`.

Você pode usar qualquer cliente `StatsD` que siga esse formato para enviar as métricas ao atendente do CloudWatch. Para obter mais informações sobre alguns dos clientes `StatsD` disponíveis, consulte a página do cliente StatsD no GitHub.

Para coletar essas métricas personalizadas, adicione uma linha `"statsd": {}` à seção `metrics_collected` do arquivo de configuração do atendente. Você pode adicionar essa linha

manualmente. Se você usa o assistente para criar o arquivo de configuração, isso é feito para você. Para ter mais informações, consulte Criar o arquivo de configuração do atendente do CloudWatch.

A configuração padrão do `StatsD` funciona para a maioria dos usuários. Há campos opcionais que podem ser adicionados, conforme necessário, à seção statsd do arquivo de configuração do agente:

- `service_address`: o endereço de serviço que o atendente do CloudWatch deve ouvir. O formato é *`ip`*`:`*`port`*. Se você omitir o endereço IP, o atendente escutará todas as interfaces disponíveis. Somente o formato UDP é compatível, portanto, você não precisa especificar um prefixo UDP.

  O valor padrão é `:8125`.
- `metrics_collection_interval`: com que frequência em segundos o plugin `StatsD` é executado e coleta métricas. O valor de padrão é de 10 segundos. O intervalo é de 1 a 172.000.
- `metrics_aggregation_interval`: com que frequência em segundos o CloudWatch agrega métricas em pontos de dados únicos. O valor padrão é de 60 segundos.

  Por exemplo, se `metrics_collection_interval` for 10 e `metrics_aggregation_interval` for 60, o CloudWatch coletará dados a cada 10 segundos. Após cada minuto, as seis leituras de dados realizadas nesse minuto são agregadas em um único ponto de dados, que é enviado ao CloudWatch.

  O intervalo é de 0 a 172.000. Definir `metrics_aggregation_interval` como 0 desabilita a agregação de métricas do `StatsD`.
- `allowed_pending_messages`: o número de mensagens UDP que podem ser enfileiradas. Quando a fila está cheia, o servidor StatsD começa a descartar pacotes. O valor padrão é 10000.
- `drop_original_metrics`: optional. Se você estiver usando o campo `aggregation_dimensions` na seção `metrics` para agrupar métricas em resultados agregados, por padrão, o agente enviará as métricas agregadas e as métricas originais que são separadas para cada valor da dimensão. Se você não quiser que as métricas originais sejam enviadas ao CloudWatch, é possível especificar esse parâmetro com uma lista de métricas. As métricas especificadas junto a esse parâmetro não têm suas métricas por dimensão relatadas ao CloudWatch. Em vez disso, somente as métricas agregadas são relatadas. Isso reduz o número de métricas que o agente coleta, reduzindo seus custos.

Veja a seguir um exemplo da seção statsd do arquivo de configuração do atendente, usando a porta padrão e intervalos de coleta e agregação personalizados.

```
{
   "metrics":{
      "metrics_collected":{
         "statsd":{
            "service_address":":8125",
            "metrics_collection_interval":60,
            "metrics_aggregation_interval":300
         }
      }
   }
}
```

Visualizar métricas do StatsD importadas pelo atendente do CloudWatch

Depois de importar métricas do StatsD para o CloudWatch, é possível visualizar essas métricas como gráficos de séries temporais e criar alarmes que podem observar essas métricas e notificar você, se elas violarem um limite especificado. O procedimento a seguir mostra como visualizar métricas do StatsD como um gráfico de séries temporais. Para obter mais informações sobre configuração de alarmes, consulte Usar alarmes do Amazon CloudWatch.

Para exibir métricas do StatsD no console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Escolha o namespace para as métricas coletadas pelo atendente. Por padrão, é CWAgent, mas você pode ter especificado um namespace diferente no arquivo de configuração do atendente do CloudWatch.
4. Escolha uma dimensão de métrica; por exemplo, Per-Instance Metrics (Métricas por instância).
5. A guia All metrics (Todas as métricas) exibe todas as métricas dessa dimensão no namespace. Você pode fazer o seguinte:

    a. Para criar um gráfico de uma métrica, marque a caixa de seleção ao lado da métrica. Para selecionar todas as métricas, marque a caixa de seleção na linha de cabeçalho da tabela.
    b. Para classificar a tabela, use o cabeçalho da coluna.
    c. Para filtrar por recurso, escolha o ID do recurso e, em seguida, escolha Adicionar à pesquisa.
    d. Para filtrar por métrica, selecione o nome da métrica e, em seguida, escolha Adicionar à pesquisa.

6. (Opcional) Para adicionar esse gráfico a um painel do CloudWatch, escolha Actions (Ações), Add to dashboard (Adicionar ao painel).

Recuperar métricas personalizadas com o collectd

É possível recuperar métricas adicionais de suas aplicações ou seus serviços usando o atendente do CloudWatch com o protocolo collectd, que é compatível somente com servidores Linux. O collectd é uma solução de código aberto bastante usada com plugins que podem coletar estatísticas do sistema para uma ampla variedade de aplicações. Ao combinar as métricas do sistema que o atendente do CloudWatch já pode coletar com as métricas adicionais do collectd, é possível monitorar, analisar e solucionar problemas de seus sistemas e suas aplicações. Para obter mais informações sobre o collectd, consulte collectd - The system statistics collection daemon.

Use o software collectd para enviar as métricas ao atendente do CloudWatch. Para as métricas do collectd, o atendente do CloudWatch atua como o servidor, enquanto o plugin do collectd atua como o cliente.

O software collectd não é instalado automaticamente em todos os servidores. Em um servidor com o Amazon Linux 2 em execução, siga estas etapas para instalar o collectd

```
sudo amazon-linux-extras install collectd
```

Para obter informações sobre como instalar collectd em outros sistemas, consulte a página Download para collectd.

Para coletar essas métricas personalizadas, adicione uma linha "collectd": {} à seção metrics_collected do arquivo de configuração do atendente. Você pode adicionar essa linha manualmente. Se você usa o assistente para criar o arquivo de configuração, isso é feito para você. Para ter mais informações, consulte Criar o arquivo de configuração do atendente do CloudWatch.

Parâmetros opcionais também estão disponíveis. Se estiver usando o collectd e não usar `/etc/collectd/auth_file` como collectd_auth_file, você deverá definir algumas dessas opções.

- service_address: o endereço de serviço que o atendente do CloudWatch deve ouvir. O formato é "`udp://`*`ip`*`:`*`port`*. O padrão é `udp://127.0.0.1:25826`.
- name_prefix: um prefixo para anexar ao início do nome de cada métrica do collectd. O padrão é `collectd_`. O tamanho máximo é de 255 caracteres.

- collectd_security_level: define o nível de segurança para comunicação de rede. O padrão é Encrypt (Criptografia).

  Encrypt (Criptografar) especifica que apenas dados criptografados são aceitos. Sign (Assinar) especifica que apenas dados com assinatura e criptografados são aceitos. None (Nenhum) especifica que todos os dados são aceitos. Se você especificar um valor para collectd_auth_file, os dados criptografados serão descriptografados, se possível.

  Para obter mais informações, consulte Configuração do cliente e Interações possíveis na Wiki do collectd.

- collectd_auth_file Define um arquivo no qual os nomes de usuário são mapeados para as senhas. Essas senhas são usadas para verificar assinaturas e descriptografar pacotes de rede criptografados. Se fornecidos, os dados assinados são verificados e os pacotes criptografados são descriptografados. Caso contrário, os dados assinados são aceitos sem verificar a assinatura e os dados criptografados não podem ser descriptografados.

  O padrão é `/etc/collectd/auth_file`.

  Se collectd_security_level estiver definido como None (Nenhum), é opcional. Se definir collectd_security_level como `encrypt` ou Sign (Assinar), você deverá especificar collectd_auth_file.

  Para o formato do arquivo de autenticação, cada linha é um nome de usuário seguido por dois pontos e qualquer número de espaços seguido pela senha. Por exemplo:

  ```
  user1: user1_password

  user2: user2_password
  ```

- collectd_typesdb: uma lista de um ou mais arquivos que contêm as descrições de conjunto de dados. A lista deve estar entre colchetes, ainda que haja somente uma entrada na lista. Cada entrada na lista deve estar entre aspas duplas. Se houver várias entradas, separe-as por vírgulas. O padrão em servidores Linux é `["/usr/share/collectd/types.db"]`. O padrão em computadores macOS depende da versão do collectd. Por exemplo, `["/usr/local/Cellar/collectd/5.12.0/share/collectd/types.db"]`.

  Para ter mais informações, consulte https://www.collectd.org/documentation/manpages/types.db.html.

- metrics_aggregation_interval: com que frequência, em segundos, o CloudWatch agrega métricas aos pontos de dados únicos. O padrão é 60 segundos. O intervalo é de 0 a 172,000. Defini-lo como 0 desabilita a agregação de métricas do collectd.

A seguir, veja um exemplo da seção do collectd do arquivo de configuração do atendente.

```
{
   "metrics":{
      "metrics_collected":{
         "collectd":{
            "name_prefix":"My_collectd_metrics_",
            "metrics_aggregation_interval":120
         }
      }
   }
}
```

Visualizar métricas do collectd importadas pelo atendente do CloudWatch

Depois de importar métricas do collectd para o CloudWatch, é possível visualizar essas métricas como gráficos de séries temporais e criar alarmes que podem observar essas métricas e notificar você, se elas violarem um limite especificado. O procedimento a seguir mostra como visualizar métricas do collectd como um gráfico de séries temporais. Para obter mais informações sobre configuração de alarmes, consulte Usar alarmes do Amazon CloudWatch.

Para exibir métricas do collectd no console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Escolha o namespace para as métricas coletadas pelo atendente. Por padrão, é CWAgent, mas você pode ter especificado um namespace diferente no arquivo de configuração do atendente do CloudWatch.
4. Escolha uma dimensão de métrica; por exemplo, Per-Instance Metrics (Métricas por instância).
5. A guia All metrics (Todas as métricas) exibe todas as métricas dessa dimensão no namespace. Você pode fazer o seguinte:

   a. Para criar um gráfico de uma métrica, marque a caixa de seleção ao lado da métrica. Para selecionar todas as métricas, marque a caixa de seleção na linha de cabeçalho da tabela.

b. Para classificar a tabela, use o cabeçalho da coluna.

c. Para filtrar por recurso, escolha o ID do recurso e, em seguida, escolha Adicionar à pesquisa.

d. Para filtrar por métrica, selecione o nome da métrica e, em seguida, escolha Adicionar à pesquisa.

6. (Opcional) Para adicionar esse gráfico a um painel do CloudWatch, escolha Actions (Ações), Add to dashboard (Adicionar ao painel).

Instalar e configurar a coleta de métricas do Prometheus em instâncias do Amazon EC2

As seções a seguir explicam como instalar o atendente do CloudWatch com o monitoramento Prometheus em instâncias do EC2 e como configurar o atendente para extrair de outros destinos. Também fornece tutoriais para configurar workloads de amostra para usar testes com monitoramento Prometheus.

Para obter informações sobre os sistemas operacionais compatíveis com o atendente do CloudWatch, consulte Coletar métricas, logs e rastreamentos com o agente do CloudWatch

Requisitos para grupo de segurança de VPC

Se você estiver usando uma VPC, os seguintes requisitos se aplicam.

- As regras de entrada dos grupos de segurança para as workloads do Prometheus devem abrir as portas do Prometheus para o atendente do CloudWatch para extrair as métricas Prometheus pelo IP privado.
- As regras de saída do grupo de segurança do atendente do CloudWatch devem permitir que o atendente do CloudWatch se conecte à porta das workloads do Prometheus por IP privado.

Tópicos

Etapa 1: Instalar o atendente do CloudWatch

A primeira etapa consiste em instalar o atendente do CloudWatch na instância do EC2. Para obter instruções, consulte Instalação do atendente do CloudWatch.

Etapa 2: Extrair fontes do Prometheus e importar métricas

O atendente do CloudWatch com monitoramento Prometheus precisa de duas configurações para extrair as métricas do Prometheus. Uma serve para as configurações padrão do Prometheus, conforme documentado em <scrape_config> na documentação do Prometheus. A outra é para a configuração do atendente do CloudWatch.

Configuração de extração do Prometheus

O atendente do CloudWatch oferece suporte às configurações de extração padrão do Prometheus, conforme documentado em <scrape_config> na documentação do Prometheus. É possível editar essa seção para atualizar as configurações que já estão nesse arquivo e adicionar outros destinos de extração do Prometheus. Um arquivo de configuração de exemplo contém as seguintes linhas de configuração global:

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
- job_name: MY_JOB
  sample_limit: 10000
  file_sd_configs:
    - files: ["C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\prometheus_sd_1.yaml",
 "C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\prometheus_sd_2.yaml"]
```

A seção `global` especifica parâmetros que são válidos em todos os contextos de configuração. Eles também servem como padrão para outras seções de configuração. Contém os seguintes parâmetros:

- `scrape_interval`: define a frequência da adição de destinos de extração de conteúdo.
- `scrape_timeout`: define quanto tempo aguardar até a expiração de uma solicitação de extração de conteúdo.

A seção `scrape_configs` especifica um conjunto de destinos e parâmetros que definem como extraí-los. Contém os seguintes parâmetros:

- `job_name`: o nome do trabalho atribuído a métricas extraídas por padrão.
- `sample_limit`: limite por extração no número de amostras extraídas que serão aceitas.
- `file_sd_configs`: lista de configurações de detecção de serviço de arquivo. Lê um conjunto de arquivos contendo uma lista de zero ou mais configurações estáticas. A seção `file_sd_configs` contém um parâmetro `files` que define padrões para arquivos dos quais os grupos de destino são extraídos.

O atendente do CloudWatch é compatível com os seguintes tipos de configuração de detecção de serviço.

`static_config` Permite especificar uma lista de destinos e um conjunto de rótulos comuns para eles. É a maneira canônica de especificar destinod estáticos em uma configuração de extração.

Veja a seguir um exemplo de configuração estática para extrair métricas do Prometheus de um host local. As métricas também poderão ser extraídas de outros servidores, se a porta Prometheus estiver aberta para o servidor onde o atendente é executado.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_sd_1.yaml
- targets:
    - 127.0.0.1:9404
  labels:
    key1: value1
    key2: value2
```

Esse exemplo contém os seguintes parâmetros:

- `targets`: os destinos extraídos pela configuração estática.
- `labels`: rótulos atribuídos a todas as métricas que são raspadas dos destinos.

`ec2_sd_config` Permite recuperar destinos de extração de instâncias do Amazon EC2. Veja a seguir uma amostra `ec2_sd_config` para extrair métricas do Prometheus de uma lista de instâncias do EC2. As portas do Prometheus dessas instâncias devem abrir para o servidor onde o atendente do CloudWatch está em execução. A função do IAM para a instância do EC2 em que o atendente do CloudWatch é executado deve incluir a permissão `ec2:DescribeInstance`. Por exemplo, você pode anexar a política gerenciada AmazonEC2ReadOnlyAccess à instância em que o atendente do CloudWatch está em execução.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
```

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: MY_JOB
    sample_limit: 10000
    ec2_sd_configs:
      - region: us-east-1
        port: 9404
        filters:
          - name: instance-id
            values:
              - i-98765432109876543
              - i-12345678901234567
```

Esse exemplo contém os seguintes parâmetros:

- `region`: a região da AWS em que a instância do EC2 de destino está. Se estiver em branco, será usada a região dos metadados da instância.
- `port`: a porta de onde as métricas serão extraídas.
- `filters`: filtros opcionais a serem usados para filtrar a lista de instâncias. Este exemplo filtra com base em IDs de instância do EC2. Para obter mais critérios que você pode filtrar, consulte DescribeInstances.

Configuração do atendente do CloudWatch para o Prometheus

O arquivo de configuração do atendente do CloudWatch inclui as seções `prometheus` em `logs` e `metrics_collected`. Inclui os seguintes parâmetros.

- cluster_name: especifica o nome do cluster a ser adicionado como um rótulo no evento de log. Esse campo é opcional.
- log_group_name: especifica o nome do grupo de log para as métricas do Prometheus extraídas.
- prometheus_config_path: especifica o caminho do arquivo de configuração de extração do Prometheus.
- emf_processor: especifica a configuração do processador de formato de métrica incorporado. Para obter mais informações sobre o formato de métrica incorporado, consulte Incorporação de métricas em logs.

  A seção `emf_processor` pode contem estes parâmetros:

- metric_declaration_dedup: é definida como true, a função de eliminação de duplicação para as métricas de formato de métrica incorporado está habilitada.
- metric_namespace: especifica o namespace da métrica para as métricas emitidas do CloudWatch.
- metric_unit: especifica o nome métrica: mapa de unidade da métrica. Para obter informações sobre unidades de métrica compatíveis, consulte MetricDatum.
- metric_declaration: são seções que especificam a matriz de logs com formato de métrica incorporado a ser gerada. Há seções `metric_declaration` para cada destino do Prometheus do qual o atendente do CloudWatch importa por padrão. Essas seções incluem os seguintes campos:
  - `source_labels` especifica o valor dos rótulos verificados pela linha `label_matcher`.
  - `label_matcher` é uma expressão regular que confere o valor dos rótulos listados em `source_labels`. As métricas correspondentes são disponibilizadas para inclusão no formato de métrica incorporado enviado ao CloudWatch.
  - `metric_selectors` é uma expressão regular que especifica as métricas a serem coletadas e enviadas ao CloudWatch.
  - `dimensions` é a lista de rótulos a serem usados como dimensões do CloudWatch para cada métrica selecionada.

Veja a seguir um exemplo de configuração do atendente do CloudWatch para o Prometheus.

```
{
   "logs":{
      "metrics_collected":{
         "prometheus":{
            "cluster_name":"prometheus-cluster",
            "log_group_name":"Prometheus",
            "prometheus_config_path":"C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\prometheus.yaml",
            "emf_processor":{
               "metric_declaration_dedup":true,
               "metric_namespace":"CWAgent-Prometheus",
               "metric_unit":{
                  "jvm_threads_current": "Count",
                  "jvm_gc_collection_seconds_sum": "Milliseconds"
               },
               "metric_declaration":[
```

```
                {
                   "source_labels":[
                      "job", "key2"
                   ],
                   "label_matcher":"MY_JOB;^value2",
                   "dimensions":[
                      [
                         "key1", "key2"
                      ],
                      [
                         "key2"
                      ]
                   ],
                   "metric_selectors":[
                      "^jvm_threads_current$",
                      "^jvm_gc_collection_seconds_sum$"
                   ]
                }
             ]
          }
       }
    }
}
```

O exemplo anterior configura uma seção de formato de métrica incorporado a ser enviada como um evento de log, se as seguintes condições forem atendidas:

- O valor do rótulo `job` é `MY_JOB`
- O valor do rótulo `key2` é `value2`
- As métricas `jvm_threads_current` e `jvm_gc_collection_seconds_sum` do Prometheus contêm rótulos `job` e `key2`.

O evento de log enviado inclui a seção destacada a seguir.

```
{
    "CloudWatchMetrics": [
        {
            "Metrics": [
                {
                    "Unit": "Count",
```

```
                    "Name": "jvm_threads_current"
                },
                {
                    "Unit": "Milliseconds",
                    "Name": "jvm_gc_collection_seconds_sum"
                }
            ],
            "Dimensions": [
                [
                    "key1",
                    "key2"
                ],
                [
                    "key2"
                ]
            ],
            "Namespace": "CWAgent-Prometheus"
        }
    ],
    "ClusterName": "prometheus-cluster",
    "InstanceId": "i-0e45bd06f196096c8",
    "Timestamp": "1607966368109",
    "Version": "0",
    "host": "EC2AMAZ-PDDOIUM",
    "instance": "127.0.0.1:9404",
    "jvm_threads_current": 2,
    "jvm_gc_collection_seconds_sum": 0.006000000000000002,
    "prom_metric_type": "gauge",
    ...
}
```

Exemplo: configurar workloads de amostra Java/JMX para tested de métrica do Prometheus

O JMX Exporter é um exportador oficial do Prometheus que pode extrair conteúdo e expor mBeans da JMX como métricas do Prometheus. Para obter mais informações, consulte prometheus/jmx_exporter.

O atendente do CloudWatch pode coletar métricas predefinidas do Prometheus a partir da Java Virtual Machine (JVM), Hjava e Tomcat (Catalina) usando um JMX Exporter em instâncias do EC2.

Etapa 1: Instalar o atendente do CloudWatch

A primeira etapa consiste em instalar o atendente do CloudWatch na instância do EC2. Para obter instruções, consulte Instalação do atendente do CloudWatch.

Etapa 2: Iniciar a workload do Java/JMX

A próxima etapa é iniciar a workload do Java/JMX.

Primeiro, baixe o arquivo jar do JMX Exporter mais recente do seguinte local: prometheus/jmx_exporter.

Use o jar para sua aplicação de amostra

Os exemplos de comando nas seções a seguir usam `SampleJavaApplication-1.0-SNAPSHOT.jar` como o arquivo jar. Substitua essas partes dos comandos pelo jar de sua aplicação.

Preparar a configuração do JMX Exporter

O arquivo `config.yaml` é o arquivo de configuração do JMX Exporter. Para obter mais informações, consulte Configuration (Configuração) na documentação do JMX Exporter.

Veja a seguir um exemplo de arquivo de configuração para Java e Tomcat.

```
---
lowercaseOutputName: true
lowercaseOutputLabelNames: true

rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
```

```
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

Inicie a aplicação Java com o exportador do Prometheus

Inicie o exemplo de aplicação. Isso emitirá métricas do Prometheus para a porta 9404. Substitua o ponto de entrada `com.gubupt.sample.app.App` pelas informações corretas para seu exemplo de aplicação java.

No Linux, insira o comando a seguir.

```
$ nohup java -javaagent:./jmx_prometheus_javaagent-0.14.0.jar=9404:./config.yaml -cp
  ./SampleJavaApplication-1.0-SNAPSHOT.jar com.gubupt.sample.app.App &
```

No Windows, insira o comando a seguir.

```
PS C:\> java -javaagent:.\jmx_prometheus_javaagent-0.14.0.jar=9404:.\config.yaml -cp  .
\SampleJavaApplication-1.0-SNAPSHOT.jar com.gubupt.sample.app.App
```

Verificar a emissão de métricas do Prometheus

Verifique se as métricas do Prometheus estão sendo emitidas.

No Linux, insira o comando a seguir.

```
$ curl localhost:9404
```

No Windows, insira o comando a seguir.

```
PS C:\> curl  http://localhost:9404
```

Exemplo de saída no Linux:

```
StatusCode        : 200
StatusDescription : OK
Content           : # HELP jvm_classes_loaded The number of classes that are currently
 loaded in the JVM
                    # TYPE jvm_classes_loaded gauge
                    jvm_classes_loaded 2526.0
                    # HELP jvm_classes_loaded_total The total number of class...
RawContent        : HTTP/1.1 200 OK
                    Content-Length: 71908
                    Content-Type: text/plain; version=0.0.4; charset=utf-8
                    Date: Fri, 18 Dec 2020 16:38:10 GMT

                    # HELP jvm_classes_loaded The number of classes that are
 currentl...
Forms             : {}
Headers           : {[Content-Length, 71908], [Content-Type, text/plain; version=0.0.4;
 charset=utf-8], [Date, Fri, 18
                    Dec 2020 16:38:10 GMT]}
Images            : {}
```

```
InputFields       : {}
Links             : {}
ParsedHtml        : System.__ComObject
RawContentLength  : 71908
```

Etapa 3: Configurar o atendente do CloudWatch para extrair métricas do Prometheus

Em seguida, instale a configuração de extração do Prometheus no arquivo de configuração do atendente do CloudWatch.

Para instalar a configuração de extração do Prometheus para o exemplo de Java/JMX

1. Instale a configuração para `file_sd_config` e `static_config`.

   No Linux, insira o comando a seguir.

   ```
   $ cat /opt/aws/amazon-cloudwatch-agent/var/prometheus.yaml
   global:
     scrape_interval: 1m
     scrape_timeout: 10s
   scrape_configs:
     - job_name: jmx
       sample_limit: 10000
       file_sd_configs:
         - files: [ "/opt/aws/amazon-cloudwatch-agent/var/prometheus_file_sd.yaml" ]
   ```

   No Windows, insira o comando a seguir.

   ```
   PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
   global:
     scrape_interval: 1m
     scrape_timeout: 10s
   scrape_configs:
     - job_name: jmx
       sample_limit: 10000
       file_sd_configs:
         - files: [ "C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
   \prometheus_file_sd.yaml" ]
   ```

2. Defina a configuração dos destinos da extação.

   No Linux, insira o comando a seguir.

```
$ cat /opt/aws/amazon-cloudwatch-agent/var/prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: linux
```

No Windows, insira o comando a seguir.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: windows
```

3. Defina a configuração de extração do Prometheus como `ec2_sc_config`. Substitua *`your-ec2-instance-id`* pelo ID correto da instância do EC2.

   No Linux, insira o comando a seguir.

```
$ cat .\prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    ec2_sd_configs:
      - region: us-east-1
        port: 9404
        filters:
          - name: instance-id
            values:
              - your-ec2-instance-id
```

   No Windows, insira o comando a seguir.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
```

```
  labels:
    application: sample_java_app
    os: windows
```

4. Instalar a configuração do atendente do CloudWatch Primeiro, navegue até o diretório correto. No Linux, é `/opt/aws/amazon-cloudwatch-agent/var/cwagent-config.json`. No Windows, é `C:\ProgramData\Amazon\AmazonCloudWatchAgent\cwagent-config.json`.

   Veja a seguir um exemplo de configuração com métricas do Java/JHX Prometheus definidas. Substitua *`path-to-Prometheus-Scrape-Configuration-file`* pelo caminho correto.

```
{
  "agent": {
    "region": "us-east-1"
  },
  "logs": {
    "metrics_collected": {
      "prometheus": {
        "cluster_name": "my-cluster",
        "log_group_name": "prometheus-test",
        "prometheus_config_path": "path-to-Prometheus-Scrape-Configuration-file",
        "emf_processor": {
          "metric_declaration_dedup": true,
          "metric_namespace": "PrometheusTest",
          "metric_unit":{
            "jvm_threads_current": "Count",
            "jvm_classes_loaded": "Count",
            "java_lang_operatingsystem_freephysicalmemorysize": "Bytes",
            "catalina_manager_activesessions": "Count",
            "jvm_gc_collection_seconds_sum": "Seconds",
            "catalina_globalrequestprocessor_bytesreceived": "Bytes",
            "jvm_memory_bytes_used": "Bytes",
            "jvm_memory_pool_bytes_used": "Bytes"
          },
          "metric_declaration": [
            {
              "source_labels": ["job"],
              "label_matcher": "^jmx$",
              "dimensions": [["instance"]],
              "metric_selectors": [
                "^jvm_threads_current$",
                "^jvm_classes_loaded$",
```

```
                "^java_lang_operatingsystem_freephysicalmemorysize$",
                "^catalina_manager_activesessions$",
                "^jvm_gc_collection_seconds_sum$",
                "^catalina_globalrequestprocessor_bytesreceived$"
              ]
            },
            {
              "source_labels": ["job"],
              "label_matcher": "^jmx$",
              "dimensions": [["area"]],
              "metric_selectors": [
                "^jvm_memory_bytes_used$"
              ]
            },
            {
              "source_labels": ["job"],
              "label_matcher": "^jmx$",
              "dimensions": [["pool"]],
              "metric_selectors": [
                "^jvm_memory_pool_bytes_used$"
              ]
            }
          ]
        }
      }
    },
    "force_flush_interval": 5
  }
}
```

5. Insira um dos comandos a seguir para reiniciar o atendente do CloudWatch.

   No Linux, insira o comando a seguir.

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/var/cwagent-config.json
   ```

   No Windows, insira o comando a seguir.

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
    -a fetch-config -m ec2 -s -c file:C:\ProgramData\Amazon\AmazonCloudWatchAgent
   \cwagent-config.json
   ```

Visualizar as métricas e logs do Prometheus

Agora é possível visualizar as métricas do Java/JMX que estão sendo coletadas.

Para visualizar as métricas do exemplo de workload do Java/JMX

1. Abra o console CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Na região da em que o cluster está em execução, escolha Metrics (Métricas) no painel de navegação à esquerda. Encontre o namespace PrometheUSTEST para ver as métricas.
3. Para visualizar os eventos do CloudWatch Logs, escolha Log Groups (Grupos de logs) no painel de navegação. Os eventos estão no grupo de logs prometheus-test.

Instalar o agente do CloudWatch usando o complemento de observabilidade do EKS do Amazon CloudWatch

O complemento Amazon CloudWatch Observability do EKS instala o agente do CloudWatch e o agente do Fluent Bit em um cluster do Amazon EKS, com uma observabilidade aprimorada do Container Insights para o Amazon EKS e com o CloudWatch Application Signals habilitado por padrão. Ao usar o complemento, é possível coletar as métricas de infraestrutura, a telemetria de performance para as aplicações e os logs de contêiner do cluster do Amazon EKS.

Com o Container Insights com capacidade de observabilidade aprimorada para o Amazon EKS, as métricas do Container Insights são cobradas por observação, em vez de serem cobradas por métrica armazenada ou log ingerido. Para o Application Signals, o faturamento é baseado nas solicitações de entrada para as aplicações, nas solicitações de saída das aplicações e em cada objetivo de nível de serviço (SLO) configurado. Cada solicitação de entrada recebida gera um sinal de aplicação e cada solicitação de saída realizada gera um sinal de aplicação. Cada SLO cria dois sinais de aplicações por período de medição. Para obter mais informações sobre os preços do CloudWatch, consulte Preço do Amazon CloudWatch.

O complemento do Amazon EKS possibilita o uso do Container Insights em nós de processamento do Linux e do Windows no cluster do Amazon EKS. Para habilitar o Container Insights no Windows, é necessário usar a versão 1.5.0 ou versões posteriores do complemento do Amazon EKS. No momento, o Application Signals não é compatível com sistema Windows em clusters do Amazon EKS.

O complemento Amazon CloudWatch Observability do EKS é compatível com os clusters do Amazon EKS executados com a versão 1.23 ou com versões posteriores do Kubernetes.

Ao instalar o complemento, também é necessário conceder permissões do IAM para habilitar que o agente do CloudWatch envie métricas, logs e rastreamentos para o CloudWatch. Há duas maneiras de fazer isso:

- Anexe uma política à função do IAM dos nós de processamento. Essa opção concede permissões aos nós de processamento para enviar telemetria ao CloudWatch.
- Use um perfil do IAM para contas de serviço dos pods de agentes e anexe a política a esse perfil. Funciona somente para clusters do Amazon EKS. Essa opção dá ao CloudWatch acesso apenas aos pods de agente adequados.

Opção 1: instalar com permissões do IAM nos nós de processamento

Para usar esse método, primeiro anexe a política do IAM CloudWatchAgentServerPolicy aos nós de processamento, digitando o comando a seguir. Nesse comando, substitua *my-worker-node-role* pelo perfil do IAM usado por seus nós de processamento do Kubernetes.

```
aws iam attach-role-policy \
--role-name my-worker-node-role \
--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Em seguida, instale o complemento de observabilidade do EKS do Amazon CloudWatch. Para instalar o complemento, você pode usar a AWS CLI, o console, o AWS CloudFormation ou o Terraform.

AWS CLI

Como usar a AWS CLI para instalar o complemento Amazon CloudWatch Observability do EKS

Insira o comando da a seguir. Substitua o *my-cluster-name* pelo nome do cluster.

```
aws eks create-addon --addon-name amazon-cloudwatch-observability --cluster-name my-cluster-name
```

Amazon EKS console

Como usar o console do Amazon EKS para adicionar o complemento Amazon CloudWatch Observability do EKS

1. Abra o console do Amazon EKS em https://console.aws.amazon.com/eks/home#/clusters.

2. No painel de navegação à esquerda, escolha Clusters.
3. Escolha o nome do cluster para o qual você deseja configurar o complemento Amazon CloudWatch Observability do EKS.
4. Escolha a guia Add-ons (Complementos).
5. Escolha Obter mais complementos.
6. Na página Selecionar complementos, faça o seguinte:
    a. Na seção Amazon EKS-addons, marque a caixa de seleção Observabilidade do Amazon CloudWatch.
    b. Escolha Próximo.
7. Na página Definir as configurações dos complementos selecionados:
    a. Selecione a Versão que deseja usar.
    b. Em Selecionar perfil do IAM, selecione Herdar do nó
    c. (Opcional) Você pode expandir Definições de configuração opcionais. Se você selecionar Substituir como Método de resolução de conflitos, uma ou mais configurações do complemento existente poderão ser substituídas pelas configurações do complemento do Amazon EKS. Se você não habilitar esta opção e houver um conflito com suas configurações existentes, a operação falhará. É possível usar a mensagem de erro resultante para solucionar o conflito. Antes de selecionar essa opção, certifique-se de que o complemento do Amazon EKS não gerencie as configurações que você precisa autogerenciar.
    d. Escolha Próximo.
8. Na página Adicionar tags, escolha Criar. Depois que a instalação do complemento for concluída, você verá o complemento instalado.

AWS CloudFormation

Como usar o AWS CloudFormation para instalar o complemento Amazon CloudWatch Observability do EKS

Substitua o *my-cluster-name* pelo nome do cluster. Para obter mais informações, consulte AWS::EKS::Addon.

```
{
    "Resources": {
```

```
        "EKSAddOn": {
            "Type": "AWS::EKS::Addon",
            "Properties": {
                "AddonName": "amazon-cloudwatch-observability",
                "ClusterName": "my-cluster-name"
            }
        }
    }
}
```

Terraform

Como usar o Terraform para instalar o complemento Amazon CloudWatch Observability do EKS

Substitua o *my-cluster-name* pelo nome do cluster. Para obter mais informações, consulte Recurso: aws_eks_addon.

```
resource "aws_eks_addon" "example" {
  addon_name   = "amazon-cloudwatch-observability"
  cluster_name = "my-cluster-name"
}
```

Opção 2: instalar usando a função de conta de serviço do IAM

Antes de usar esse método, verifique os seguintes pré-requisitos:

- Você possui um cluster funcional do Amazon EKS com nós conectados em uma das Regiões da AWS que são compatíveis com o Container Insights. Para obter a lista de regiões compatíveis, consulte Container Insights.
- Você instalou o `kubectl` e configurou o cluster. Para obter mais informações, consulte Instalar o `kubectl` no Manual do usuário do Amazon EKS.
- Você tem o `eksctl` instalado. Para obter mais informações, consulte Instalação ou atualização do `eksctl` no Guia do usuário do Amazon EKS.

Instalar o complemento de observabilidade do EKS do Amazon CloudWatch usando o perfil de conta de serviço do IAM

1. Insira o comando seguir para criar um provedor do OpenID Connect (OIDC), se o cluster ainda não tiver um. Para obter mais informações, consulte Configuração de uma conta de serviço do Kubernetes para assumir um perfil do IAM no Guia do usuário do Amazon EKS.

```
eksctl utils associate-iam-oidc-provider --cluster my-cluster-name --approve
```

2. Insira o comando a seguir para criar o perfil do IAM com a política CloudWatchAgentServerPolicy anexada e configure a conta do serviço de agente para assumir esse perfil usando o OIDC. Substitua *my-cluster-name* pelo nome do seu cluster e substitua *my-service-account-role* pelo nome do perfil ao qual você deseja associar a conta de serviço. Se o perfil ainda não existir, o `eksctl` o criará para você.

```
eksctl create iamserviceaccount \
  --name cloudwatch-agent \
  --namespace amazon-cloudwatch --cluster my-cluster-name \
  --role-name my-service-account-role \
  --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
  --role-only \
  --approve
```

3. Instale o complemento inserindo o comando a seguir. Substitua *my-cluster-name* pelo nome do seu cluster, substitua *111122223333* pela ID da sua conta e substitua *my-service-account-role* pelo perfil do IAM criado na etapa anterior.

```
aws eks create-addon --addon-name amazon-cloudwatch-observability --cluster-
name my-cluster-name --service-account-role-arn arn:aws:iam::111122223333:role/my-
service-account-role
```

(Opcional) Configuração adicional

Opte por não coletar logs de contêineres

Por padrão, o complemento usa o Fluent Bit para coletar logs de contêineres de todos os pods e, em seguida, envia os logs para o CloudWatch Logs. Para obter informações sobre quais logs são coletados, consulte Configurar o Fluent Bit.

Para desativar a coleta de logs de contêineres, passe a seguinte opção ao criar ou atualizar o complemento:

```
--configuration-values '{ "containerLogs": { "enabled": false } }'
```

Opte por não participar da coleção de métricas de GPU NVIDIA

A partir da versão 1.300034.0 do agente CloudWatch, o Container Insights coleta métricas de GPU NVIDIA das workloads do EKS por padrão. Essas métricas estão listadas na tabela em Métricas da GPU NVIDIA.

Você pode optar por não coletar métricas de GPU NVIDIA definindo a opção `accelerated_compute_metrics` no arquivo de configuração do agente CloudWatch como `false`. Essa opção está na seção `kubernetes` da seção `metrics_collected` no arquivo de configuração do CloudWatch. Este é um exemplo de uma configuração de exclusão.

```
{
  "agent": {
    "region": "us-east-1"
  },
  "logs": {
    "metrics_collected": {
      "emf": {
      },
      "kubernetes": {
        "enhanced_container_insights": true,
        "accelerated_compute_metrics": false
      }
    },
    "force_flush_interval": 5,
  }
}
```

Como usar uma configuração personalizada do agente do CloudWatch

Para coletar métricas, logs ou rastreamentos adicionais usando o agente do CloudWatch, é possível especificar uma configuração personalizada e, ao mesmo tempo, manter o Container Insights e o CloudWatch Application Signals habilitados. Para fazê-lo, incorpore o arquivo de configuração do agente do CloudWatch na chave de configuração no âmbito da chave do agente da configuração avançada que você pode usar ao criar ou ao atualizar o complemento para o EKS. Veja a seguir

uma representação da configuração padrão do agente quando nenhuma configuração adicional é fornecida.

> **Important**
>
> Qualquer configuração personalizada fornecida usando as definições de configurações adicionais substitui a configuração padrão usada pelo agente. Tenha cuidado para não desabilitar acidentalmente as funcionalidades que são habilitadas por padrão, como o Container Insights com uma observabilidade aprimorada e o CloudWatch Application Signals. Diante do cenário em que é necessário fornecer uma configuração do agente personalizada, recomendamos usar a configuração padrão apresentada a seguir como linha de base e, em seguida, modificá-la com base nas suas necessidades.

```
--configuration-values '{
  "agent": {
    "config": {
      "logs": {
        "metrics_collected": {
          "app_signals": {},
          "kubernetes": {
            "enhanced_container_insights": true
          }
        }
      },
      "traces": {
        "traces_collected": {
          "app_signals": {}
        }
      }
    }

}'
```

O exemplo apresentado a seguir mostra a configuração padrão do agente do CloudWatch no Windows. O agente do CloudWatch no Windows não oferece suporte à configuração personalizada.

```
{
  "logs": {
    "metrics_collected": {
```

```
      "kubernetes": {
        "enhanced_container_insights": true
      },
    }
  }
}
```

Gerenciamento de certificados TLS para o webhook de admissão

O complemento Amazon CloudWatch Observability do EKS utiliza webhooks de admissão do Kubernetes para validar e alterar solicitações de recursos personalizados (CR) do `AmazonCloudWatchAgent` e de `Instrumentation` e, opcionalmente, solicitações de pod do Kubernetes no cluster, se o CloudWatch Application Signals estiver habilitado. No Kubernetes, os webhooks requerem um certificado TLS no qual o servidor de API esteja configurado para confiar, a fim de garantir uma comunicação segura.

Por padrão, o complemento Amazon CloudWatch Observability do EKS gera automaticamente uma CA autoassinada e um certificado TLS assinado por essa CA para proteger a comunicação entre o servidor de API e o servidor de webhook. O certificado gerado automaticamente tem uma expiração padrão de dez anos e não é renovado de forma automática após expirar. Além disso, o pacote da CA e o certificado são gerados novamente sempre que o complemento é atualizado ou reinstalado, redefinindo, assim, a expiração. Caso deseje alterar a expiração padrão do certificado gerado automaticamente, você poderá usar as configurações adicionais apresentadas a seguir ao criar ou atualizar o complemento. Substitua *`expiry-in-days`* pelo período de expiração desejado em dias.

```
--configuration-values '{ "admissionWebhooks": { "autoGenerateCert":
 { "expiryDays": expiry-in-days } } }'
```

Para obter uma solução mais segura e repleta de recursos da autoridade de certificação, o complemento tem suporte opcional para o cert-manager, uma solução amplamente adotada para o gerenciamento de certificados TLS no Kubernetes que simplifica o processo de obtenção, renovação, gerenciamento e uso desses certificados. A solução garante que os certificados sejam válidos e estejam atualizados, bem como busca renová-los em um momento configurado antes da expiração. Além disso, o cert-manager facilita a emissão de certificados de diversas fontes com suporte, incluindo o AWS Certificate Manager Private Certificate Authority.

Recomendamos analisar as práticas recomendadas para o gerenciamento de certificados TLS em seus clusters e aconselhamos a opção pelo cert-manager para ambientes de produção. Observe

que, se você optar por habilitar o cert-manager para gerenciar os certificados TLS para o webhook de admissão, será necessário instalar previamente o cert-manager no cluster do Amazon EKS antes de instalar o complemento Amazon CloudWatch Observability do EKS. Consulte a documentação do cert-manager para saber mais sobre as opções de instalação disponíveis. Após a instalação, é possível optar por usar o cert-manager para o gerenciamento dos certificados TLS para o webhook de admissão usando a configuração adicional apresentada a seguir ao criar ou atualizar o complemento.

```
--configuration-values '{ "admissionWebhooks": { "certManager": { "enabled":
 true } } }'
```

A configuração avançada debatida nesta seção usará, por padrão, um emissor SelfSigned.

Como coletar IDs de volume do Amazon EBS

Se quiser coletar IDs de volume do Amazon EBS nos logs de performance, será necessário adicionar outra política ao perfil do IAM que está anexado aos nós de processamento ou à conta de serviço. Adicione o seguinte como uma política em linha. Para obter mais informações, consulte Adicionar e remover permissões de identidade do IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Solução de problemas do complemento Amazon CloudWatch Observability do EKS

Use as informações apresentadas a seguir para ajudar a solucionar problemas relacionados ao complemento Amazon CloudWatch Observability do EKS.

Atualização e exclusão do complemento Amazon CloudWatch Observability do EKS

Para obter instruções sobre como atualizar ou excluir o complemento Amazon CloudWatch Observability do EKS, consulte Gerenciar complementos do Amazon EKS. Use `amazon-cloudwatch-observability` como o nome do complemento.

Verificação da versão do agente do CloudWatch usado pelo complemento Amazon CloudWatch Observability do EKS

O complemento Amazon CloudWatch Observability do EKS instala um recurso personalizado do tipo `AmazonCloudWatchAgent` que controla o comportamento do daemonset do agente do CloudWatch no cluster, incluindo a versão do agente do CloudWatch que está sendo usada. É possível obter uma lista de todos os recursos personalizados do tipo `AmazonCloudWatchAgent`, que estão instalados em seu cluster, ao inserir o seguinte comando:

```
kubectl get amazoncloudwatchagent -A
```

Na saída desse comando, você poderá verificar a versão do agente do CloudWatch. Como alternativa, também é possível descrever o recurso `amazoncloudwatchagent` ou um dos pods do `cloudwatch-agent-*` em execução no cluster para inspecionar a imagem que está sendo usada.

Tratamento de um ConfigurationConflict durante o gerenciamento do complemento

Ao instalar ou atualizar o complemento Amazon CloudWatch Observability do EKS, se você perceber uma falha causada por um `Health Issue` do tipo `ConfigurationConflict` com uma descrição que começa com `Conflicts found when trying to apply. Will not continue due to resolve conflicts mode`, é provável que você já tenha o agente do CloudWatch e os componentes associados, como o ServiceAccount, o ClusterRole e o ClusterRoleBinding instalados no cluster. Quando o complemento tentar instalar o agente do CloudWatch e os componentes associados, se ele detectar quaisquer alterações no conteúdo, por padrão, apresentará falhas na instalação ou na atualização para evitar a substituição do estado dos recursos no cluster.

Se você estiver tentando realizar a integração do complemento Amazon CloudWatch Observability do EKS e obter essa falha, recomendamos excluir uma configuração existente do agente do CloudWatch instalada anteriormente no cluster e, em seguida, instalar o complemento do EKS. Certifique-se de fazer backup de quaisquer personalizações que você possa ter executado na configuração original do agente do CloudWatch, como uma configuração do agente personalizada, e fornecê-las ao complemento Amazon CloudWatch Observability do EKS na próxima instalação

ou atualização. Se você realizou a instalação do agente do CloudWatch para a integração com o Container Insights, consulte Exclusão do agente do CloudWatch e do Fluent Bit para o Container Insights para obter mais informações.

Como alternativa, o complemento oferece suporte a uma opção de configuração de resolução de conflitos que tem a funcionalidade de especificar `OVERWRITE`. É possível usar essa opção para prosseguir com a instalação ou a atualização do complemento ao substituir os conflitos no cluster. Se você estiver usando o console do Amazon EKS, encontrará o Método de resolução de conflitos ao escolher as Definições de configuração opcionais na criação ou na atualização do complemento. Caso esteja usando a AWS CLI, você poderá fornecer o comando `--resolve-conflicts OVERWRITE` para criar ou atualizar o complemento.

Métricas coletadas pelo atendente do CloudWatch

É possível coletar métricas de servidores instalando o atendente do CloudWatch no servidor. Você pode instalar o atendente nas instâncias do Amazon EC2 e nos servidores on-premises, bem como nos servidores que executam o Linux, o Windows Server ou macOS. Se você instalar o atendente em uma instância do Amazon EC2, as métricas que ele coletar serão complementares às métricas habilitadas por padrão em instâncias do Amazon EC2.

Para obter informações sobre como instalar o atendente do CloudWatch em uma instância, consulte Coletar métricas, logs e rastreamentos com o agente do CloudWatch.

Todas as métricas discutidas nesta seção são coletadas diretamente pelo atendente do CloudWatch.

Métricas coletadas pelo atendente do CloudWatch em instâncias do Windows Server

Em um servidor que execute o Windows Server, instalar o atendente do CloudWatch permite coletar as métricas associadas ao contadores no Monitor de Performance do Windows. Os nomes de métrica do CloudWatch para esses contadores são criados colocando um espaço entre o nome do objeto e o nome do contador. Por exemplo, o contador `% Interrupt Time` do objeto `Processor` recebe o nome da métrica `Processor % Interrupt Time` no CloudWatch. Para obter mais informações sobre os contadores do Monitor de Performance do Windows, consulte a documentação do Microsoft Windows Server.

O namespace padrão para métricas coletadas pelo atendente do CloudWatch é `CWAgent`, embora seja possível especificar um namespace diferente quando você configura o atendente.

Métricas coletadas pelo atendente do CloudWatch em instâncias do Linux e macOS

A tabela a seguir relaciona as métricas coletadas pelo atendente do CloudWatch em servidores Linux e computadores macOS.

Métrica	Descrição
`cpu_time_active`	A quantidade de tempo que a CPU está ativa em qualquer capacidade. Essa métrica é medida em centésimos de segundo. Unidade: nenhuma
`cpu_time_guest`	A quantidade de tempo em que a CPU está executando uma CPU virtual para um sistema operacional convidado. Essa métrica é medida em centésimos de segundo. Unidade: nenhuma
`cpu_time_guest_nice`	O tempo em que a CPU está executando uma CPU virtual para um sistema operacional convidado, que é de baixa prioridade e pode ser interrompida por outros processos. Essa métrica é medida em centésimos de segundo. Unidade: nenhuma
`cpu_time_idle`	A quantidade de tempo em que a CPU está ociosa. Essa métrica é medida em centésimos de segundo. Unidade: nenhuma
`cpu_time_iowait`	A quantidade de tempo em que a CPU está aguardando a conclusão de operações de entrada/saída. Essa métrica é medida em centésimos de segundo.

Métrica	Descrição
	Unidade: nenhuma
`cpu_time_irq`	A quantidade de tempo em que a CPU está atendendo a interrupções. Essa métrica é medida em centésimos de segundo. Unidade: nenhuma
`cpu_time_nice`	O tempo em que a CPU permanece em modo de usuário com processos de baixa prioridade que podem ser facilmente interrompidos por processos de prioridade mais alta. Essa métrica é medida em centésimos de segundo. Unidade: nenhuma
`cpu_time_softirq`	A quantidade de tempo em que a CPU está atendendo a interrupções de software. Essa métrica é medida em centésimos de segundo. Unidade: nenhuma
`cpu_time_steal`	A quantidade de tempo em que a CPU está no tempo roubado, que é o tempo gasto em outros sistemas operacionais em um ambiente virtualizado. Essa métrica é medida em centésimos de segundo. Unidade: nenhuma
`cpu_time_system`	A quantidade de tempo em que a CPU está no modo de sistema. Essa métrica é medida em centésimos de segundo. Unidade: nenhuma

Métrica	Descrição
cpu_time_user	A quantidade de tempo em que a CPU está no modo de usuário. Essa métrica é medida em centésimos de segundo. Unidade: nenhuma
cpu_usage_active	A porcentagem de tempo que a CPU está ativa em qualquer capacidade. Unidade: Percentual
cpu_usage_guest	O percentual de tempo que a CPU está executando uma CPU virtual para um sistema operacional convidado. Unidade: Percentual
cpu_usage_guest_nice	A porcentagem de tempo em que a CPU está executando uma CPU virtual para um sistema operacional convidado que é de baixa prioridade e pode ser interrompida por outros processos. Unidade: Percentual
cpu_usage_idle	O percentual de tempo em que a CPU está ociosa. Unidade: Percentual
cpu_usage_iowait	O percentual de tempo que a CPU está aguardando a conclusão de operações de entrada/saída. Unidade: Percentual
cpu_usage_irq	O percentual de tempo que a CPU está atendendo a interrupções. Unidade: Percentual

Métrica	Descrição
cpu_usage_nice	A porcentagem de tempo que a CPU está em modo de usuário com processos de baixa prioridade que podem ser facilmente interrompidos por processos de prioridade mais alta. Unidade: Percentual
cpu_usage_softirq	O percentual de tempo que a CPU está atendendo a interrupções de software. Unidade: Percentual
cpu_usage_steal	O percentual de tempo que a CPU está no tempo roubado, ou o tempo gasto em outros sistemas operacionais em um ambiente virtualizado. Unidade: Percentual
cpu_usage_system	O percentual de tempo que a CPU está no modo de sistema. Unidade: Percentual
cpu_usage_user	O percentual de tempo que a CPU está no modo de usuário. Unidade: Percentual
disk_free	O espaço livre nos discos. Unidade: bytes
disk_inodes_free	O número de nós de índice disponíveis no disco. Unidade: Contagem

Métrica	Descrição
`disk_inodes_total`	O número total de nós de índice reservados no disco. Unidade: Contagem
`disk_inodes_used`	O número de nós de índice usados no disco. Unidade: Contagem
`disk_total`	Total de espaço nos discos, incluindo usado e gratuito. Unidade: bytes
`disk_used`	O espaço usado nos discos. Unidade: bytes
`disk_used_percent`	O percentual do total de espaço em disco que é usado. Unidade: Percentual
`diskio_iops_in_progress`	O número de solicitações de E/S que foram emitidas para o driver de dispositivo, mas ainda não foram concluídas. Unidade: Contagem
`diskio_io_time`	A quantidade de tempo que o disco tem solicitações de E/S na fila. Unidade: milissegundos A única estatística que deve ser usada para essa métrica é `Sum`. Não use `Average`.

Métrica	Descrição
`diskio_reads`	O número de operações de leitura de disco. Unidade: Contagem A única estatística que deve ser usada para essa métrica é `Sum`. Não use `Average`.
`diskio_read_bytes`	O número de bytes lidos dos discos. Unidade: bytes A única estatística que deve ser usada para essa métrica é `Sum`. Não use `Average`.
`diskio_read_time`	A quantidade de tempo que solicitações de leitura aguardaram nos discos. Várias solicitações de leitura em espera ao mesmo tempo que aumentam em número. Por exemplo, se 5 solicitações aguardare m uma média de 100 milissegundos, 500 serão relatadas. Unidade: milissegundos A única estatística que deve ser usada para essa métrica é `Sum`. Não use `Average`.
`diskio_writes`	O número de operações de gravação de disco. Unidade: Contagem A única estatística que deve ser usada para essa métrica é `Sum`. Não use `Average`.
`diskio_write_bytes`	O número de bytes gravados nos discos. Unidade: bytes A única estatística que deve ser usada para essa métrica é `Sum`. Não use `Average`.

Métrica	Descrição
`diskio_write_time`	A quantidade de tempo que solicitações de gravação aguardaram nos discos. Várias solicitações de gravação em espera ao mesmo tempo que aumentam em número. Por exemplo, se 8 solicitaç ões aguardarem uma média de 1000 milissegundos, 8000 serão relatadas. Unidade: milissegundos A única estatística que deve ser usada para essa métrica é `Sum`. Não use `Average`.
`ethtool_bw_in_allowance_exc eeded`	Número de pacotes na fila e/ou descartados porque a largura de banda agregada de entrada excedeu o máximo para a instância. Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_c ollected` do arquivo de configuração do atendente do CloudWatch. Para mais informações, consulte Coletar métricas de performance da rede Unidade: nenhuma
`ethtool_bw_out_allowance_ex ceeded`	Número de pacotes na fila e/ou descartados porque a largura de banda agregada de saída excedeu o máximo para a instância. Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_c ollected` do arquivo de configuração do atendente do CloudWatch. Para mais informações, consulte Coletar métricas de performance da rede Unidade: nenhuma

Métrica	Descrição
`ethtool_conntrack_allowance_exceeded`	Número de pacotes descartados porque o monitoramento da conexão excedeu o máximo para a instância e não foi possível estabelecer novas conexões. Isso pode resultar em perda de pacotes para tráfego indo para a instância ou vindo da instância Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. Para mais informações, consulte Coletar métricas de performance da rede Unidade: nenhuma
`ethtool_linklocal_allowance_exceeded`	Número de pacotes descartados porque o PPS do tráfego para os serviços de proxy local excedeu o máximo para a interface da rede. Isso afeta o tráfego para o serviço de DNS, o Instance Metadata Service e o Amazon Time Sync Service. Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. Para mais informações, consulte Coletar métricas de performance da rede Unidade: nenhuma

Métrica	Descrição
`ethtool_pps_allowance_exceeded`	Número de pacotes na fila e/ou descartados porque o PPS bidirecional excedeu o máximo para a instância. Essa métrica será coletada somente se você listá-la na subseção `ethtool` da seção `metrics_collected` do arquivo de configuração do atendente do CloudWatch. Para ter mais informações, consulte Coletar métricas de performance da rede. Unidade: nenhuma
`mem_active`	A quantidade de memória que foi usada de alguma maneira durante o último período de amostra. Unidade: bytes
`mem_available`	A quantidade de memória que está disponível e pode ser fornecida instantaneamente para os processos. Unidade: bytes
`mem_available_percent`	O percentual de memória que está disponível e pode ser fornecido instantaneamente para os processos. Unidade: Percentual
`mem_buffered`	A quantidade de memória que está sendo usada para buffers. Unidade: bytes
`mem_cached`	A quantidade de memória que está sendo usada para caches de arquivo. Unidade: bytes

Métrica	Descrição
`mem_free`	A quantidade de memória que não está sendo usada. Unidade: bytes
`mem_inactive`	A quantidade de memória que não foi usada de alguma maneira durante o último período de amostra Unidade: bytes
`mem_total`	A quantidade total de memória. Unidade: bytes
`mem_used`	A quantidade de memória em uso no momento. Unidade: bytes
`mem_used_percent`	O percentual de memória em uso no momento. Unidade: Percentual
`net_bytes_recv`	O número de bytes recebidos pela interface de rede. Unidade: bytes A única estatística que deve ser usada para essa métrica é `Sum`. Não use `Average`.
`net_bytes_sent`	O número de bytes enviados pela interface de rede. Unidade: bytes A única estatística que deve ser usada para essa métrica é `Sum`. Não use `Average`.

Métrica	Descrição
net_drop_in	O número de pacotes recebidos por essa interface de rede que foram descartados. Unidade: Contagem A única estatística que deve ser usada para essa métrica é Sum. Não use Average.
net_drop_out	O número de pacotes transmitidos por essa interface de rede que foram descartados. Unidade: Contagem A única estatística que deve ser usada para essa métrica é Sum. Não use Average.
net_err_in	O número de erros de recepção detectados por essa interface de rede. Unidade: Contagem A única estatística que deve ser usada para essa métrica é Sum. Não use Average.
net_err_out	O número de erros de transmissão detectados por essa interface de rede. Unidade: Contagem A única estatística que deve ser usada para essa métrica é Sum. Não use Average.
net_packets_sent	O número de pacotes enviados por essa interface de rede. Unidade: Contagem A única estatística que deve ser usada para essa métrica é Sum. Não use Average.

Métrica	Descrição
`net_packets_recv`	O número de pacotes recebidos por essa interface de rede. Unidade: Contagem A única estatística que deve ser usada para essa métrica é `Sum`. Não use `Average`.
`netstat_tcp_close`	O número de conexões de TCP sem estado. Unidade: Contagem
`netstat_tcp_close_wait`	O número de conexões de TCP esperando por uma solicitação de encerramento do cliente. Unidade: Contagem
`netstat_tcp_closing`	O número de conexões de TCP que estão esperando por uma solicitação de encerramento com reconhecimento do cliente. Unidade: Contagem
`netstat_tcp_established`	O número de conexões de TCP estabelecidas. Unidade: Contagem
`netstat_tcp_fin_wait1`	O número de conexões de TCP no estado `FIN_WAIT1` durante o processo de encerramento de uma conexão. Unidade: Contagem
`netstat_tcp_fin_wait2`	O número de conexões de TCP no estado `FIN_WAIT2` durante o processo de encerramento de uma conexão. Unidade: Contagem

Métrica	Descrição
`netstat_tcp_last_ack`	O número de conexões de TCP esperando o cliente para enviar a confirmação da mensagem de encerramento da conexão. Este é o último estado logo antes que a conexão seja encerrada. Unidade: Contagem
`netstat_tcp_listen`	O número de portas TCP atualmente escutando uma solicitação de conexão. Unidade: Contagem
`netstat_tcp_none`	O número de conexões de TCP com clientes inativos. Unidade: Contagem
`netstat_tcp_syn_sent`	O número de conexões de TCP esperando por uma solicitação de conexão correspondente após enviar uma solicitação de conexão. Unidade: Contagem
`netstat_tcp_syn_recv`	O número de conexões de TCP esperando por uma confirmação de solicitação de conexão após enviar e receber uma solicitação de conexão. Unidade: Contagem
`netstat_tcp_time_wait`	O número de conexões de TCP atualmente esperando para garantir que o cliente tenha recebido a confirmação da solicitação de encerramento da conexão. Unidade: Contagem

Métrica	Descrição
`netstat_udp_socket`	O número de conexões de UDP atuais. Unidade: Contagem
`processes_blocked`	O número de processos que estão bloqueados. Unidade: Contagem
`processes_dead`	O número de processos que estão inativos, o que é indicado pelo código de estado `X` no Linux. Essa métrica não é coletada em computadores macOS. Unidade: Contagem
`processes_idle`	O número de processos que estão ociosos (em suspensão por mais de 20 segundos). Disponível apenas em instâncias do FreeBSD. Unidade: Contagem
`processes_paging`	O número de processos que estão em paginação, o que é indicado pelo código de estado `W` no Linux. Essa métrica não é coletada em computadores macOS. Unidade: Contagem
`processes_running`	O número de processos que estão em execução, indicado pelo código de estado `R`. Unidade: Contagem
`processes_sleeping`	O número de processos que estão em suspensão, indicado pelo código de estado `S`. Unidade: Contagem

Métrica	Descrição
processes_stopped	O número de processos que estão interrompidos, indicado pelo código de estado T. Unidade: Contagem
processes_total	O número total de processos na instância. Unidade: Contagem
processes_total_threads	O número total de threads que compõem os processos. Essa métrica está disponível apenas em instâncias do Linux. Essa métrica não é coletada em computadores macOS. Unidade: Contagem
processes_wait	O número de processos que estão em paginação, o que é indicado pelo código de estado W em instâncias do FreeBSD. Essa métrica está disponível apenas em instâncias do FreeBSD e não está disponível em instâncias do Linux, Windows Server ou macOS. Unidade: Contagem
processes_zombies	O número de processos zumbi, indicado pelo código de estado Z. Unidade: Contagem
swap_free	A quantidade de espaço de troca que não está em uso. Unidade: bytes

Métrica	Descrição
`swap_used`	A quantidade de espaço de troca em uso no momento. Unidade: bytes
`swap_used_percent`	O percentual de espaço de troca em uso no momento. Unidade: Percentual

Definições de métricas de memória coletadas pelo agente do CloudWatch

Quando o agente do CloudWatch coleta métricas de memória, a origem é o subsistema de gerenciamento de memória do host. Por exemplo, o kernel do Linux expõe dados mantidos pelo sistema operacional em `/proc`. Para a memória, os dados estão em `/proc/meminfo`.

Cada sistema operacional e arquitetura diferente tem cálculos diferentes dos recursos usados pelos processos. Para obter mais informações, consulte as seções a seguir.

Durante cada intervalo de coleta, o agente do CloudWatch em cada instância coleta os recursos da instância e calcula os recursos que estão sendo usados por todos os processos que estão sendo executados nessa instância. Essas informações são relatadas de volta para as métricas do CloudWatch. É possível configurar a duração do intervalo de coleta no arquivo de configuração do agente do CloudWatch. Para ter mais informações, consulte Arquivo de configuração do atendente do CloudWatch: seção do atendente.

A lista a seguir explica como as métricas de memória que o agente do CloudWatch coleta são definidas.

- Memória ativa: memória que está sendo usada por um processo. Em outras palavras, a memória usada pelas aplicações em execução no momento.
- Memória disponível: a memória que pode ser fornecida instantaneamente aos processos sem que o sistema entre em swap (também conhecida como memória virtual).
- Memória em buffer: a área de dados compartilhada por dispositivos de hardware ou processos de programas que operam em velocidades e prioridades diferentes.

- Memória em cache: armazena instruções e dados do programa que são usados repetidamente na operação dos programas que a CPU provavelmente precisará em seguida.
- Memória livre: memória que não está sendo usada e está prontamente disponível. É totalmente livre para que o sistema a use quando necessário.
- Memória inativa: páginas que não foram acessadas "recentemente".
- Memória total: o tamanho real da memória física RAM.
- Memória usada: memória que está sendo usada atualmente por programas e processos.

Tópicos

Linux: métricas coletadas e cálculos usados

Métricas coletadas e unidades:

- Ativa (bytes)
- Disponível (bytes)
- Percentual disponível (percentual)
- Em buffer (bytes)
- Em cache (bytes)
- Livre (bytes)
- Inativa (bytes)
- Total (bytes)
- Usada (bytes)
- Percentual usado (percentual)

Memória usada = Memória total - Memória livre - Memória em cache - Memória em buffer

Memória total = Memória usada + Memória livre + Memória em cache + Memória em buffer

macOS: métricas coletadas e cálculos usados

Métricas coletadas e unidades:

- Ativa (bytes)
- Disponível (bytes)
- Percentual disponível (percentual)
- Livre (bytes)
- Inativa (bytes)
- Total (bytes)
- Usada (bytes)
- Percentual usado (percentual)

Memória disponível = Memória livre + Memória inativa

Memória usada = Memória total - Memória disponível

Memória total = Memória disponível + Memória usada

Windows: métricas coletadas

As métricas coletadas nos hosts Windows estão listadas abaixo. Todas essas métricas têm `None` para `Unit`.

- Bytes disponíveis
- Falhas de cache/seg
- Falhas de página/seg
- Páginas/seg

Não há cálculos usados para métricas do Windows porque o agente do CloudWatch analisa eventos usando contadores de performance.

Exemplo: cálculo de métricas de memória no Linux

Como exemplo, suponha que a inserção do comando **`cat /proc/meminfo`** em um host Linux mostre os resultados a seguir:

```
MemTotal:       3824388 kB
MemFree:         462704 kB
MemAvailable:   2157328 kB
Buffers:         126268 kB
Cached:         1560520 kB
SReclaimable:    289080 kB>
```

Neste exemplo, o agente do CloudWatch coletará os valores a seguir. Todos os valores que o agente do CloudWatch coleta e reporta estão em bytes.

- `mem_total`: 3916173312 bytes
- `mem_available`: 2209103872 bytes (MemFree+ em cache)
- `mem_free`: 473808896 bytes
- `mem_cached`: 1893990400 bytes (`cached` + `SReclaimable`
- `mem_used`: 1419075584 bytes (`MemTotal` – (`MemFree` + `Buffers` + (`Cached` + `SReclaimable`)))
- `mem_buffered`: 129667072 bytes
- `mem_available_percent`: 56,41%
- `mem_used_percent`: 36,24% (`mem_used` / `mem_total`) * 100

Cenários comuns com o atendente do CloudWatch

As seções a seguir descrevem como realizar tarefas comuns de configuração e personalização do agente do CloudWatch.

Tópicos

- Diferenças de carimbo de data/hora entre o atendente unificado do CloudWatch e o atendente mais antigo do CloudWatch Logs

Executar o atendente do CloudWatch como um usuário diferente

Em servidores Linux, o CloudWatch é executado como o usuário raiz por padrão. Para fazer com que o atendente seja executado como um usuário diferente, use o parâmetro `run_as_user` na seção do `agent` no arquivo de configuração do atendente do CloudWatch. Essa opção está disponível apenas em servidores Linux.

Se você já estiver executando o atendente com o usuário raiz e desejar passar a usar um usuário diferente, use um dos procedimentos a seguir.

Para executar o atendente do CloudWatch como um usuário diferente em uma instância do EC2 que executa o Linux

1. Baixe e instale um novo pacote do atendente do CloudWatch. Para ter mais informações, consulte Baixar o pacote do atendente do CloudWatch.
2. Crie um novo usuário do Linux ou utilize o usuário padrão chamado `cwagent` que o arquivo RPM ou DEB criou.
3. Forneça as credenciais para esse usuário de uma das seguintes maneiras:
   - Se o arquivo `.aws/credentials` existir no diretório base do usuário raiz, seránecessário criar um arquivo de credenciais para o usuário que será usado para executar o atendente do CloudWatch. Esse arquivo de credenciais será `/home/`*`username`*`/.aws/credentials`. Depois defina o valor do parâmetro `shared_credential_file` no `common-config.toml` como o nome do caminho do arquivo de credenciais. Para ter mais informações, consulte (Opcional) Modificar a configuração comum para informações de proxy ou região.
   - Se o arquivo`.aws/credentials` não existir no diretório home do usuário raiz, você poderá seguir um destes procedimentos:
     - Crie um arquivo de credenciais para o usuário que será usado para executar o atendente do CloudWatch. Esse arquivo de credenciais será `/home/`*`username`*`/.aws/credentials`. Depois defina o valor do parâmetro `shared_credential_file` no `common-config.toml` como o nome do caminho do arquivo de credenciais. Para ter mais informações, consulte (Opcional) Modificar a configuração comum para informações de proxy ou região.

- Em vez de criar um arquivo de credenciais, anexe uma função do IAM à instância. O atendente usa essa função como o provedor de credenciais.

4. No arquivo de configuração do atendente do CloudWatch, adicione a seguinte linha à seção `agent`:

```
"run_as_user": "username"
```

   Faça outras modificações no arquivo de configuração conforme necessário. Para ter mais informações, consulte Criar o arquivo de configuração do atendente do CloudWatch.

5. Dê ao usuário as permissões necessárias. O usuário deve ter permissões de Leitura (r) para que os arquivos de log sejam coletados e ter permissão de Executar (x) em cada diretório no caminho dos arquivos de log.

6. Inicie o atendente com o arquivo de configuração que você acabou de modificar.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

Para executar o atendente do CloudWatch como um usuário diferente em um servidor on-premises que executa o Linux

1. Baixe e instale um novo pacote do atendente do CloudWatch. Para ter mais informações, consulte Baixar o pacote do atendente do CloudWatch.

2. Crie um novo usuário do Linux ou utilize o usuário padrão chamado `cwagent` que o arquivo RPM ou DEB criou.

3. Armazene as credenciais desse usuário em um caminho que o usuário possa acessar, como `/home/username/.aws/credentials`.

4. Defina o valor do parâmetro `shared_credential_file` em `common-config.toml` como o nome do caminho do arquivo de credenciais. Para ter mais informações, consulte (Opcional) Modificar a configuração comum para informações de proxy ou região.

5. No arquivo de configuração do atendente do CloudWatch, adicione a seguinte linha à seção `agent`:

```
"run_as_user": "username"
```

Faça outras modificações no arquivo de configuração conforme necessário. Para ter mais informações, consulte Criar o arquivo de configuração do atendente do CloudWatch.

6. Dê ao usuário permissões necessárias. O usuário deve ter permissões de Leitura (r) para que os arquivos de log sejam coletados e ter permissão de Executar (x) em cada diretório no caminho dos arquivos de log.
7. Inicie o atendente com o arquivo de configuração que você acabou de modificar.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

Como o atendente do CloudWatch lida com arquivos de log esparsos

Arquivos esparsos são arquivos com blocos vazios e conteúdo real. Um arquivo esparso usa espaço em disco de forma mais eficiente, escrevendo informações breves que representam os blocos vazios no disco em vez dos bytes nulos reais que compõem o bloco. Isso torna o tamanho real de um arquivo esparso geralmente muito menor do que seu tamanho aparente.

No entanto, o atendente do CloudWatch não trata arquivos esparsos de forma diferente do que trata arquivos normais. Quando o atendente lê um arquivo esparso, os blocos vazios são tratados como blocos “reais” preenchidos com bytes nulos. Por isso, o atendente do CloudWatch publica tantos bytes quanto o tamanho aparente de um arquivo esparso no CloudWatch.

Configurar o atendente do CloudWatch para publicar um arquivo esparso pode causar um aumento inesperado de custos do CloudWatch, então recomendamos não fazer isso. Por exemplo, `/var/logs/lastlog` no Linux geralmente é um arquivo muito esparso, e recomendamos não publicá-lo no CloudWatch.

Adicionar dimensões personalizadas a métricas coletadas pelo atendente do CloudWatch

Para adicionar dimensões personalizadas, como tags para métricas coletadas pelo atendente, adicione o campo `append_dimensions` à seção do arquivo de configuração do atendente que lista essas métricas.

Por exemplo, a seguinte seção de exemplo do arquivo de configuração adiciona uma dimensão personalizada chamada `stackName` com um valor de `Prod` para as métricas de `cpu` e `disk` coletadas pelo atendente.

```
"cpu":{
  "resources":[
    "*"
  ],
  "measurement":[
    "cpu_usage_guest",
    "cpu_usage_nice",
    "cpu_usage_idle"
  ],
  "totalcpu":false,
  "append_dimensions":{
    "stackName":"Prod"
  }
},
"disk":{
  "resources":[
    "/",
    "/tmp"
  ],
  "measurement":[
    "total",
    "used"
  ],
  "append_dimensions":{
    "stackName":"Prod"
  }
}
```

Lembre-se de que, sempre que você alterar o arquivo de configuração do atendente, deverá reiniciar o atendente para que as alterações entrem em vigor.

Vários arquivos de configuração do atendente CloudWatch

É possível configurar o agente do CloudWatch para usar vários arquivos de configuração em servidores Linux e Windows. Por exemplo, é possível usar um arquivo de configuração comum que coleta um conjunto de métricas, logs e rastreamentos que você sempre deseja coletar de todos os servidores em sua infraestrutura. Depois, você pode usar arquivos de configuração adicionais que coletam métricas de determinadas aplicações ou em certas situações.

Para configurar isso, primeiro crie os arquivos de configuração que você deseja usar. Todos os arquivos de configuração que serão usados em conjunto no mesmo servidor devem ter diferentes

nomes de arquivos. É possível armazenar os arquivos de configuração em servidores ou no Parameter Store.

Inicie o atendente do CloudWatch usando a opção `fetch-config` e especifique o primeiro arquivo de configuração. Para anexar o segundo arquivo de configuração para o atendente em execução, use o mesmo comando, mas com a opção `append-config`. Todas as métricas, logs e rastreamentos listados no arquivo de configuração serão coletados. No exemplo a seguir, os comandos ilustram esse cenário usando armazenamentos de configurações como arquivos. A primeira linha inicia o atendente usando o arquivo de configuração de `infrastructure.json`, e a segunda linha acrescenta o arquivo de configuração `app.json`.

Use os comandos a seguir para Linux.

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2
 -s -c file:/tmp/infrastructure.json
```

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a append-config -m
 ec2 -s -c file:/tmp/app.json
```

Use os comandos a seguir para Windows Server.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a fetch-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\infrastructure.json"
```

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a append-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\app.json"
```

No exemplo a seguir, arquivos de configuração ilustram um uso para esse recurso. O primeiro arquivo de configuração é usado para todos os servidores na infraestrutura, e o segundo recolhe apenas logs de um determinada aplicação e é anexado a servidores que executam essa aplicação.

infrastructure.json

```
{
  "metrics": {
    "metrics_collected": {
      "cpu": {
```

```
        "resources": [
          "*"
        ],
        "measurement": [
          "usage_active"
        ],
        "totalcpu": true
      },
      "mem": {
         "measurement": [
           "used_percent"
        ]
      }
    }
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
            "log_group_name": "amazon-cloudwatch-agent.log"
          },
          {
            "file_path": "/var/log/messages",
            "log_group_name": "/var/log/messages"
          }
        ]
      }
    }
  }
}
```

app.json

```
{
    "logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "/app/app.log*",
```

```
                        "log_group_name": "/app/app.log"
                    }
                ]
            }
        }
    }
}
```

Todos os arquivos de configuração anexados à configuração devem ter nomes de arquivos que sejam diferentes entre si e do arquivo de configuração inicial. Se você usar `append-config` com um arquivo de configuração com o mesmo nome do arquivo que um arquivo de configuração que o atendente já está usando, o comando append substituirá as informações do primeiro arquivo de configuração, em vez de anexar a ele. Isso é verdade mesmo se dois arquivos de configuração com o mesmo nome de arquivo estejam em diferentes caminhos de arquivo.

O exemplo anterior mostra o uso de dois arquivos de configuração, mas não há um limite para o número de arquivos de configuração que você pode anexar à configuração do atendente. Também é possível combinar o uso de arquivos de configuração localizados em servidores e configurações localizadas no Parameter Store.

Agregar ou acumular métricas coletadas pelo atendente do CloudWatch

Para agregar ou acumular métricas coletadas pelo atendente, adicione um campo `aggregation_dimensions` à seção para essa métrica no arquivo de configuração do atendente.

Por exemplo, o seguinte trecho do arquivo de configuração acumula métricas na dimensão de `AutoScalingGroupName`. As métricas de todas as instâncias em cada grupo do Auto Scaling são agregadas e podem ser visualizadas como um todo.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"]]
}
```

Para acumular junto à combinação de cada dimensão `InstanceId` e `InstanceType`, além de acumular no nome do grupo do Auto Scaling, adicione o seguinte.

```
"metrics": {
```

```
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId", "InstanceType"]]
}
```

Para acumular métricas em apenas uma coleção, use `[]`.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [[]]
}
```

Lembre-se de que, sempre que você alterar o arquivo de configuração do atendente, deverá reiniciar o atendente para que as alterações entrem em vigor.

Coletar métricas de alta resolução com o atendente do CloudWatch

O campo `metrics_collection_interval` especifica o intervalo de tempo para as métricas coletadas, em segundos. Ao especificar um valor menor do que 60 para esse campo, as métricas são coletadas como métricas de alta resolução.

Por exemplo, se todas as métricas devem ser de alta resolução e coletadas a cada 10 segundos, especifique 10 como o valor de `metrics_collection_interval` na seção `agent` como um intervalo de coleta de métricas global.

```
"agent": {
  "metrics_collection_interval": 10
}
```

Como alternativa, o exemplo a seguir define as métricas de `cpu` a serem coletadas a cada segundo, e todas as outras métricas são coletadas a cada minuto.

```
"agent":{
  "metrics_collection_interval": 60
},
"metrics":{
  "metrics_collected":{
    "cpu":{
      "resources":[
```

```
        "*"
      ],
      "measurement":[
        "cpu_usage_guest"
      ],
      "totalcpu":false,
      "metrics_collection_interval": 1
    },
    "disk":{
      "resources":[
        "/",
        "/tmp"
      ],
      "measurement":[
        "total",
        "used"
      ]
    }
  }
}
```

Lembre-se de que, sempre que você alterar o arquivo de configuração do atendente, deverá reiniciar o atendente para que as alterações entrem em vigor.

Envio de métricas, logs e rastreamentos a uma conta diferente

Para que o agente do CloudWatch envie as métricas, logs ou rastreamentos a uma conta diferente, especifique um parâmetro `role_arn` no arquivo de configuração do agente no servidor de envio. O valor `role_arn` especifica uma função do IAM na conta de destino que o atendente utiliza ao enviar dados à conta de destino. Essa função permite que a conta de envio assuma uma função correspondente na conta de destino ao entregar as métricas ou logs para a conta de destino.

Também é possível especificar strings `role_arn` separadas no arquivo de configuração do agente: uma para usar ao enviar métricas, outra para enviar logs e uma outra para enviar rastreamentos.

O exemplo a seguir de parte da seção `agent` do arquivo de configuração define o agente para usar `CrossAccountAgentRole` ao enviar dados para uma conta diferente.

```
{
  "agent": {
    "credentials": {
```

```
      "role_arn": "arn:aws:iam::123456789012:role/CrossAccountAgentRole"
    }
  },
  .....
}
```

Como alternativa, o exemplo a seguir define perfis diferentes para a conta de envio a ser usada para o envio de métricas, logs e rastreamentos:

```
"metrics": {
    "credentials": {
     "role_arn": "RoleToSendMetrics"
    },
    "metrics_collected": {....
```

```
"logs": {
    "credentials": {
    "role_arn": "RoleToSendLogs"
    },
    ....
```

Políticas necessárias

Ao especificar um `role_arn` no arquivo de configuração do atendente, você também deve garantir que as funções do IAM de contas de envio e de destino tenham determinadas políticas. As funções nas contas de envio e de destino devem ter `CloudWatchAgentServerPolicy`. Para obter mais informações sobre como atribuir essa política a uma função, consulte Criar funções do IAM a serem usadas com o atendente do CloudWatch em instâncias do Amazon EC2.

A função na conta de envio também deverá incluir a seguinte política. Você adiciona essa política na guia Permissions (Permissões) do console do IAM ao editar a função.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::target-account-ID:role/agent-role-in-target-account"
            ]
        }
    ]
}
```

A função na conta de destino deverá incluir a política a seguir para que ela reconheça a função do IAM usada pela conta de envio. Você inclui essa política na guia Trust relationships (Relações de confiança) do console do IAM ao editar a função. A função na conta de destino em que você adiciona essa política é a função que você criou em Criar funções e usuários do IAM para uso com o atendente do CloudWatch. Essa função é a função especificada em *agent-role-in-target-account* na política usada pela conta de envio.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::sending-account-ID:role/role-in-sender-account"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Diferenças de carimbo de data/hora entre o atendente unificado do CloudWatch e o atendente mais antigo do CloudWatch Logs

O atendente do CloudWatch oferece suporte a um conjunto diferente de símbolos para formatos de carimbo de data/hora, em comparação com o atendente do CloudWatch Logs mais antigo. Essas diferenças são mostradas na tabela a seguir.

Símbolos compatíveis com os dois atendentes	Símbolos compatíveis somente com o atendente unificado do CloudWatch	Símbolos compatíveis somente com o atendente mais antigo do CloudWatch Logs
%A, %a, %b, %B, %d, %f, %H, %l, %m, %M, %p, %S, %y, %Y, %Z, %z	%-d, %-l, %-m, %-M, %-S	%c,%j, %U, %W, %w

Para obter mais informações sobre os significados dos símbolos compatíveis com o novo atendente do CloudWatch, consulte Arquivo de configuração do atendente do CloudWatch: seção Logs no Manual do usuário do Amazon CloudWatch. Para obter informações sobre os significados dos símbolos compatíveis com o atendente do CloudWatch Logs, consulte Arquivo de configuração do atendente no Manual do usuário do Amazon CloudWatch Logs.

Solucionar problemas do atendente do CloudWatch

Use as seguintes informações para ajudar a solucionar problemas com o atendente do CloudWatch.

Tópicos

- Logs gerados pelo atendente do CloudWatch
- Interromper e reiniciar o atendente do CloudWatch

Parâmetros de linha de comando do atendente do CloudWatch

Para ver a lista completa de parâmetros compatíveis com o atendente do CloudWatch, insira o seguinte na linha de comando em um computador onde ele estiver instalado:

```
amazon-cloudwatch-agent-ctl -help
```

Falha ao instalar o atendente do CloudWatch usando o Run Command

Para instalar o atendente do CloudWatch usando o Systems Manager Run Command, o SSM Agent no servidor de destino deve ter a versão 2.2.93.0 ou posterior. Caso seu SSM Agent não esteja na versão correta, você poderá encontrar erros que incluem as seguintes mensagens:

```
no latest version found for package AmazonCloudWatchAgent on platform linux
```

```
failed to download installation package reliably
```

Para obter informações sobre como instalar ou atualizar a versão do SSM Agent, consulte Instalar e configurar o SSM Agent no Manual do usuário do AWS Systems Manager.

O atendente do CloudWatch não inicia

Se o atendente do CloudWatch falhar ao iniciar, pode haver um problema em sua configuração. As informações de configuração são registradas no arquivo `configuration-validation.log`. O arquivo está localizado em `/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` nos servidores Linux e em `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log` nos servidores que executam o Windows Server.

Verificar se o atendente do CloudWatch está em execução

Você pode consultar o atendente do CloudWatch para saber ele se está em execução ou se foi interrompido. Você pode usar o AWS Systems Manager para fazer isso remotamente. Também pode usar linha de comando, mas apenas para verificar um servidor local.

Para consultar o status do atendente do CloudWatch usando o Run Command

1. Abra o console do Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Executar comando.

   - ou -

   Se a página inicial do AWS Systems Manager for exibida, role para baixo e escolha Explore Run Command.
3. Selecione Run command.
4. Na lista Documento de comando, selecione o botão ao lado de AmazonCloudWatch-ManageAgent.
5. Na lista Ação, escolha status.
6. Em Origem de configuração opcional escolha padrão e mantenha Local de configuração opcional em branco.
7. Na área Destino, selecione a instância a ser verificada.
8. Escolha Executar.

Se o atendente estiver em execução, a saída poderá ser semelhante ao seguinte.

```
{
       "status": "running",
       "starttime": "2017-12-12T18:41:18",
       "version": "1.73.4"
}
```

Se o atendente for interrompido, o campo `"status"` exibirá `"stopped"`.

Para consultar o status do atendente do CloudWatch localmente usando a linha de comando

- Em um servidor Linux, digite o seguinte:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a
   status
  ```

  Em um servidor que executa o Windows Server, digite o seguinte em PowerShell como administrador:

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m
 ec2 -a status
```

O atendente do CloudWatch não é iniciado e o erro menciona uma região do Amazon EC2

Se o atendente não for iniciado e a mensagem de erro mencionar um endpoint da região do Amazon EC2, talvez você tenha configurado o atendente de maneira que ele precise acessar o endpoint do Amazon EC2 sem conceder esse acesso.

Por exemplo, se você especificar um valor para o parâmetro `append_dimensions` no arquivo de configuração do atendente que depende dos metadados do Amazon EC2 e usar proxies, será necessário verificar se o servidor pode acessar o endpoint do Amazon EC2. Para obter mais informações sobre esses endpoints, consulte Amazon Elastic Compute Cloud (Amazon EC2) no Referência geral da Amazon Web Services.

O atendente do CloudWatch não inicia no Windows Server

No Windows Server, você poderá ver o seguinte erro:

```
Start-Service : Service 'Amazon CloudWatch Agent (AmazonCloudWatchAgent)' cannot be
 started due to the following
error: Cannot start service AmazonCloudWatchAgent on computer '.'.
At C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1:113
 char:12
+     $svc | Start-Service
+            ~~~~~~~~~~~~~
    + CategoryInfo          : OpenError:
 (System.ServiceProcess.ServiceController:ServiceController) [Start-Service],
   ServiceCommandException
    + FullyQualifiedErrorId :
 CouldNotStartService,Microsoft.PowerShell.Commands.StartServiceCommand
```

Para corrigi-lo, primeiro verifique se o serviço do servidor está em execução. Esse erro pode ser visto se o atendente tentar iniciar quando o serviço do servidor não está em execução.

Se o serviço do servidor já estiver em execução, talvez o probema seja o descrito a seguir. Em algumas instalações do Windows Server, o atendente do CloudWatch leva mais de 30 segundos

para iniciar. Como o Windows Server, por padrão, permite apenas 30 segundos para iniciar serviços, isso faz com que o atendente falhe com um erro semelhante ao seguinte:

Para corrigir esse problea, aumente o valor de tempo limite do serviço. Para obter mais informações, consulte Um serviço não começa, e os eventos 7000 e 7011 estão registrados no log de eventos do Windows.

Onde estão as métricas?

Se o atendente do CloudWatch estiver em execução, mas não for possível encontrar métricas coletadas por ele no AWS Management Console ou na AWS CLI, confirme se você está usando o namespace correto. Por padrão, o namespace para métricas coletado pelo atendente é `CWAgent`. Você pode personalizar o namespace usando o campo `namespace` na seção `metrics` do arquivo de configuração do atendente. Se você não vir as métricas esperadas, verifique o arquivo de configuração para confirmar o namespace que está sendo usado.

Quando você baixa o pacote do atendente do CloudWatch pela primeira vez, o arquivo de configuração do atendente é `amazon-cloudwatch-agent.json`. Esse arquivo está no diretório em que você executou o assistente de configuração ou pode ter sido transferido para um diretório diferente. Se você usar o assistente de configuração, a saída do arquivo de configuração do atendente será chamada de `config.json`. Para obter mais informações sobre o arquivo de configuração, incluindo o campo `namespace`, consulte Arquivo de configuração do atendente do CloudWatch: seção de métricas.

O atendente do CloudWatch leva muito tempo para ser executado em um contêiner ou registra um erro de limite de salto

Se você executar o atendente do CloudWatch como um serviço de contêiner e quiser adicionar dimensões métricas do Amazon EC2 a todas as métricas coletadas pelo atendente, poderá ver os seguintes erros na versão v1.247354.0 do atendente:

```
2022-06-07T03:36:11Z E! [processors.ec2tagger] ec2tagger: Unable to retrieve Instance
 Metadata Tags. This plugin must only be used on an EC2 instance.
2022-06-07T03:36:11Z E! [processors.ec2tagger] ec2tagger: Please increase hop limit
 to 2 by following this document https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/
configuring-instance-metadata-options.html#configuring-IMDS-existing-instances.
2022-06-07T03:36:11Z E! [telegraf] Error running agent: could not initialize processor
 ec2tagger: EC2MetadataRequestError: failed to get EC2 instance identity document
caused by: EC2MetadataError: failed to make EC2Metadata request
        status code: 401, request id:
```

```
caused by:
```

Você poderá ver esse erro se o atendente tentar obter metadados de IMDSv2 dentro de um contêiner sem um limite de salto apropriado. Nas versões do atendente anteriores à v1.247354.0, você pode ter esse problema sem ver a mensagem de log.

Para resolver isso, aumente o limite de salto para 2 seguindo as instruções em Configurar as opções de metadados da instância.

Atualizei a configuração de meu atendente, mas não vejo as novas métricas ou logs no console do CloudWatch

Se você atualizar o arquivo de configuração do atendente do CloudWatch, na próxima vez que iniciar o atendente, deverá usar a opção **`fetch-config`**. Por exemplo, se você armazenou o arquivo atualizado no computador local, insira o seguinte comando:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -s -m ec2 -c file:configuration-file-path
```

Arquivos e locais do atendente do CloudWatch

A tabela a seguir lista os arquivos instalados e usados pelo atendente do CloudWatch, bem como seus locais nos servidores que executam o Linux ou o Windows Server.

Arquivo	Local do Linux	Local do Windows Server
O script de controle que controla o início, a interrupção e a reinicialização do atendente.	`/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl` ou `/usr/bin/amazon-cloudwatch-agent-ctl`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1`
O arquivo de log no qual o atendente grava. Talvez você precise anexá-lo ao entrar em contato com AWS Support.	`/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` ou `/var/log/amazon/amazon-`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log`

Arquivo	Local do Linux	Local do Windows Server
	`cloudwatch-agent/amazon-cloudwatch-agent.log`	
Arquivo de validação de configuração do atendente.	`/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` ou `/var/log/amazon/amazon-cloudwatch-agent/configuration-validation.log`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log`
O arquivo JSON usado para configurar o atendente, imediatamente após o assistente criá-lo. Para ter mais informações, consulte Criar o arquivo de configuração do atendente do CloudWatch.	`/opt/aws/amazon-cloudwatch-agent/bin/config.json`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\config.json`
O arquivo JSON usado para configurar o atendente, se esse arquivo de configuração tiver sido baixado do Parameter Store.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` ou `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json`

Arquivo	Local do Linux	Local do Windows Server
O arquivo TOML usado para especificar informações de região e credenciais a serem usadas pelo agente, substituindo os padrões do sistema.	`/opt/aws/amazon-cloudwatch-agent/etc/common-config.toml` ou `/etc/amazon/amazon-cloudwatch-agent/common-config.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\common-config.toml`
O arquivo TOML que contém o conteúdo convertido do arquivo de configuração JSON. O script `amazon-cloudwatch-agent-ctl` gera esse arquivo. Os usuários não devem modificar diretamente esse arquivo. Isso pode ser útil para verificar se a tradução de JSON para TOML foi bem-sucedida.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml` ou `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.toml`
O arquivo YAML que contém o conteúdo convertido do arquivo de configuração JSON. O script `amazon-cloudwatch-agent-ctl` gera esse arquivo. Você não deve modificar diretamente este arquivo. Este arquivo pode ser útil para verificar se a tradução de JSON para TOML foi bem-sucedida.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.yaml or /etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.yaml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.yaml`

Encontrar informações sobre versões do atendente do CloudWatch

Para localizar o número de versão do atendente do CloudWatch em um servidor Linux, insira o seguinte comando:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status
```

Para localizar o número de versão do atendente do CloudWatch no Windows Server, insira o seguinte comando:

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m ec2
 -a status
```

> **Note**
>
> Usar esse comando é a maneira correta de localizar a versão do atendente do CloudWatch. Se você usar Programs and Features (Programas e recursos) no painel de controle, verá um número de versão incorreto.

Também é possível baixar um arquivo README sobre as alterações mais recentes no atendente e um arquivo que indica o número da versão que está disponível atualmente para baixar. Esses arquivos estão nestes locais:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/RELEASE_NOTES` ou `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/RELEASE_NOTES`
- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/CWAGENT_VERSION` ou `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/CWAGENT_VERSION`

Logs gerados pelo atendente do CloudWatch

O atendente gera um log enquanto é executado. Esse log inclui informações de solução de problemas. Esse log é o arquivo `amazon-cloudwatch-agent.log`. O arquivo está localizado em `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` nos

servidores Linux e em `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log` nos servidores que executam o Windows Server.

Você pode configurar o atendente para registrar detalhes adicionais no arquivo `amazon-cloudwatch-agent.log`. No arquivo de configuração do atendente, na seção `agent`, defina o campo `debug` como `true`. Depois, reconfigure e reinicie o atendente do CloudWatch. Para desativar o registro dessas informações adicionais no log, defina o campo `debug` como `false`. Em seguida, reconfigure e reinicie o atendente. Para ter mais informações, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

Nas versões 1.247350.0 e posteriores do atendente do CloudWatch, você pode definir opcionalmente o campo `aws_sdk_log_level` na seção `agent` do arquivo de configuração do atendente com uma ou mais das opções a seguir. Separe várias opções com o caractere `|`.

- `LogDebug`
- `LogDebugWithSigning`
- `LogDebugWithHTTPBody`
- `LogDebugRequestRetries`
- `LogDebugWithEventStreamBody`

Para mais informações sobre essas opções, consulte LogLevelType.

Interromper e reiniciar o atendente do CloudWatch

Você pode interromper o atendente do CloudWatch manualmente usando o AWS Systems Manager ou a linha de comando.

Para interromper o atendente do CloudWatch usando o Run Command

1. Abra o console do Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, selecione Executar comando.

   - ou -

   Se a página inicial do AWS Systems Manager for exibida, role para baixo e escolha Explore Run Command.
3. Selecione Run command.
4. Na lista Documento do comando, escolha AmazonCloudWatch-ManageAgent.

5. Na área Targets (Destinos), escolha a instância onde você instalou o atendente do CloudWatch.
6. Na lista Ação, escolha interromper.
7. Mantenha Optional Configuration Source (Origem de configuração opcional) e Optional Configuration Location (Local de configuração opcional) em branco.
8. Escolha Executar.

Para interromper o atendente do CloudWatch localmente usando a linha de comando

- Em um servidor Linux, digite o seguinte:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a
   stop
  ```

  Em um servidor que executa o Windows Server, digite o seguinte em PowerShell como administrador:

  ```
  & $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m
   ec2 -a stop
  ```

Para reiniciar o atendente, siga as instruções em Iniciar o atendente do CloudWatch.

Incorporação de métricas em logs

O formato de métricas incorporadas do CloudWatch permite gerar métricas personalizadas de forma assíncrona na forma de logs gravados no CloudWatch Logs. É possível incorporar métricas personalizadas com dados detalhados de eventos de log, e o CloudWatch automaticamente extrairá as métricas personalizadas para que você possa visualizar e criar alarmes para elas visando a detecção de incidentes em tempo real. Além disso, os eventos de log detalhados associados às métricas extraídas podem ser consultados usando o CloudWatch Logs Insights para fornecer insights precisos sobre as causas raízes de eventos operacionais.

O formato de métricas incorporadas ajuda a gerar métricas personalizadas acionáveis de recursos efêmeros, como funções e contêineres do Lambda. Agora, ao usar o formato de métricas incorporadas para enviar logs desses recursos efêmeros, é possível criar métricas personalizadas facilmente, sem precisar instrumentar ou manter código separado, ao mesmo tempo que obtém recursos analíticos poderosos nos dados de log.

Nenhuma configuração é necessária para usar o formato de métricas incorporadas. Estruture seus logs seguindo a Especificação do formato de métricas incorporadas ou gere-os usando nossas bibliotecas de cliente e envie-os para o CloudWatch Logs usando a API PutLogEvents ou o agente do CloudWatch.

Há cobranças para a ingestão e o arquivamento de logs e para as métricas personalizadas que são geradas. Para obter mais informações, consulte Amazon CloudWatch Pricing (Preços do Amazon CloudWatch).

> **Note**
>
> Tenha cuidado ao configurar a extração de métricas, já que isso impacta o uso da métrica personalizada e o faturamento correspondente. Se você criar acidentalmente métricas baseadas em dimensões de alta cardinalidade (como `requestId`), o formato de métricas incorporadas criará por design uma métrica personalizada correspondente a cada combinação de dimensão exclusiva. Para obter mais informações, consulte Dimensões.

Tópicos

- Configuração de alarmes em métricas criadas com o formato de métricas incorporadas

Publicação de logs com o formato de métricas incorporadas

É possível gerar logs de formato de métricas incorporadas com os seguintes métodos:

- Gera e envie os logs usando as bibliotecas de cliente de código aberto.
- Gere manualmente os logs usando a Especificação do formato de métricas incorporadas, e use em seguida o agente do CloudWatch ou a API PutLogEvents para enviar os logs.

Tópicos

- Criação de logs no formato de métricas incorporadas usando as bibliotecas clientes
- Especificação: formato de métricas incorporadas
- Usar a API PutLogEvents para enviar logs de formato de métricas incorporadas criados manualmente
- Usar o atendente do CloudWatch para enviar logs de formato de métricas incorporadas
- Explica como usar o formato de métrica incorporado com o AWS Distro for OpenTelemetry

Criação de logs no formato de métricas incorporadas usando as bibliotecas clientes

A Amazon fornece bibliotecas de cliente de código aberto, que podem ser usadas para criar logs de formato de métricas incorporadas. No momento, essas bibliotecas estão disponíveis nos idiomas da lista a seguir. Exemplos completos de diferentes configurações podem ser encontrados em nossas bibliotecas de clientes em /examples.

As bibliotecas e as instruções de como usá-las estão localizadas no GitHub. Use os links a seguir.

- Node.Js

> **Note**
>
> Para o Node.js, as versões 4.1.1+, 3.0.2+, 2.0.7+ são necessárias para uso com o formato de log JSON do Lambda. O uso de versões anteriores em tais ambientes do Lambda levará à perda de métricas.

Para obter mais informações, consulte Acessar o Amazon CloudWatch Logs para o AWS Lambda.

- Python
- Java
- C#

As bibliotecas de clientes devem funcionar imediatamente com o agente do CloudWatch. Os logs de formato de métrica incorporada gerados são enviados ao agente do CloudWatch, que os agrega e publica no CloudWatch Logs para você.

> **Note**
>
> Ao usar o Lambda, nenhum agente precisa enviar os logs para o CloudWatch. Qualquer coisa registrada em log em STDOUT é enviada ao CloudWatch Logs por meio do agente de logs do Lambda.

Especificação: formato de métricas incorporadas

O formato de métrica incorporado do CloudWatch é uma especificação JSON usada para instruir o CloudWatch Logs a extrair automaticamente valores de métrica incorporados em eventos de log estruturados. É possível usar o CloudWatch para criar gráficos e alarmes com relação aos valores de métrica extraídos.

Convenções de especificação do formato de métricas incorporadas

As palavras-chave "DEVE", "NÃO DEVE", "OBRIGATÓRIO" "RECOMENDADO", "PODE" e "OPCIONAL" nessa especificação de formato devem ser interpretadas conforme descrito em Key Words RFC2119.

Os termos "JSON", "texto JSON", "valor JSON", "membro", "elemento", "objeto", "matriz", "número", "string", "booliano", "verdadeiro", "falso" e "nulo" nessa especificação de formato devem ser interpretados conforme definido em JavaScript Object Notation RFC8259.

> **Note**
>
> Se você planeja criar alarmes em métricas criadas usando o formato de métricas incorporadas, consulte Configuração de alarmes em métricas criadas com o formato de métricas incorporadas para obter as recomendações.

Estrutura de documento de formato de métrica incorporado

Esta seção descreve a estrutura de um documento de formato de métricas incorporadas. Os documentos de formato de métricas incorporadas estão definidos em JavaScript Object Notation RFC8259.

Salvo indicação em contrário, os objetos definidos por essa especificação NÃO DEVEM conter nenhum membro adicional. Os membros não reconhecidos por essa especificação DEVEM ser ignorados. Os membros definidos nessa especificação diferenciam letras maiúsculas de minúsculas.

O formato de métrica incorporado está sujeito aos mesmos limites que os eventos padrão do CloudWatch Logs e está limitado ao tamanho máximo de 256 KB.

Com o formato de métrica incorporada, você pode acompanhar o processamento de seus logs de EMF por métricas publicadas no namespace `AWS/Logs` da conta. Podem ser usados para rastrear falhas na geração de métricas do EMF, bem como se as falhas ocorrem devido à análise ou validação. Para obter mais informações, consulte Monitorar com métricas do CloudWatch.

Nó raiz

A mensagem de LogEvent DEVE ser um objeto JSON válido sem dados adicionais no início ou no final da string da mensagem de LogEvent. Para obter mais informações sobre a estrutura de LogEvent, consulte InputLogEvent.

Os documentos de formato de métricas incorporadas DEVEM conter o membro de nível superior a seguir no nó raiz. Isso é um objeto Objeto de metadados.

```
{
 "_aws": {
    "CloudWatchMetrics": [ ... ]
  }
}
```

O nó raiz DEVE conter todos os membros Membros de destino definidos pelas referências no Objeto MetricDirective.

O nó raiz PODE conter qualquer outro membro que não esteja incluído nos requisitos acima. Os valores desses membros DEVEM ser tipos JSON válidos.

Objeto de metadados

O membro `_aws` pode ser usado para representar metadados sobre a carga útil que informa os serviços downstream como eles devem processar o LogEvent. O valor DEVE ser um objeto e DEVE conter os seguintes membros:

- CloudWatchMetrics: uma matriz de Objeto MetricDirective usada para instruir o CloudWatch a extrair métricas do nó raiz de LogEvent.

```
{
  "_aws": {
    "CloudWatchMetrics": [ ... ]
  }
}
```

- Timestamp: número que representa o carimbo de data/hora usado para métricas extraídas do evento. Os valores DEVEM ser expressos como o número de milissegundos após 1º de janeiro de 1970 00:00:00 UTC.

```
{
  "_aws": {
    "Timestamp": 1559748430481
  }
}
```

Objeto MetricDirective

O objeto MetricDirective instrui serviços downstream que o LogEvent contém métricas que serão extraídas e publicadas no CloudWatch. MetricDirectives DEVE conter os seguintes membros:

- Namespace: uma string que representa o namespace do CloudWatch da métrica.
- Dimensions: um Matriz DimensionSet.
- Metrics: uma matriz de objetos MetricDefinition. Essa matriz NÃO DEVE conter mais de 100 objetos MetricDefinition.

Matriz DimensionSet

Um DimensionSet é uma matriz de strings que contém as chaves de dimensão que serão aplicadas a todas as métricas no documento. Os valores dentro dessa matriz também DEVEM ser membros no nó raiz – referido como o Membros de destino

Um DimensionSet NÃO DEVE conter mais de 30 chaves de dimensão. O DimensionSet PODE estar vazio.

O membro de destino DEVE ter um valor de string. Este valor NÃO DEVE conter mais de 1.024 caracteres. O membro de destino define uma dimensão que será publicada como parte da identidade da métrica. Cada DimensionSet usado cria uma métrica no CloudWatch. Para obter mais informações sobre dimensões, consulte Dimensão e Dimensões

```
{
 "_aws": {
   "CloudWatchMetrics": [
     {
       "Dimensions": [ [ "functionVersion" ] ],
       ...
     }
   ]
 },
 "functionVersion": "$LATEST"
}
```

> **Note**
>
> Tenha cuidado ao configurar a extração de métricas, já que isso impacta o uso da métrica personalizada e o faturamento correspondente. Se você criar acidentalmente métricas baseadas em dimensões de alta cardinalidade (como `requestId`), o formato de métricas incorporadas criará por design uma métrica personalizada correspondente a cada combinação de dimensão exclusiva. Para obter mais informações, consulte Dimensões.

Objeto MetricDefinition

Um MetricDefinition é um objeto que DEVE conter o seguinte membro:

- Name (Nome): uma string Valores de referência para uma métrica Membros de destino. Os destinos métricos DEVEM ser um valor numérico ou uma matriz de valores numéricos.

Um objeto MetricDefinition PODE conter os seguintes membros:

- Unit (Unit): um valor de string OPCIONAL que representa a unidade de medida da métrica correspondente. Os valores DEVEM ser unidades métricas válidas do CloudWatch. Para obter informações sobre unidades válidas, consulte MetricDatum. Se um valor não for fornecido, então um valor padrão de NONE (NENHUM) será assumido.
- StorageResolution: um valor inteiro OPCIONAL que representa a resolução de armazenamento da métrica correspondente. Definir isso como 1 especifica essa métrica como uma métrica de alta resolução, de forma que o CloudWatch armazene a métrica com resolução de menos de um minuto de até um segundo. Definir isso como 60 especifica essa métrica como resolução padrão, que o CloudWatch armazena com resolução de 1 minuto. Os valores DEVEM ser resoluções válidas do CloudWatch: 1 ou 60. Se um valor não for fornecido, então um valor padrão de 60 será assumido.

  Para obter mais informações sobre as métricas de alta resolução, consulte Métricas de alta resolução.

> **Note**
>
> Se você planeja criar alarmes em métricas criadas usando o formato de métricas incorporadas, consulte Configuração de alarmes em métricas criadas com o formato de métricas incorporadas para obter as recomendações.

```
{
  "_aws": {
    "CloudWatchMetrics": [
      {
        "Metrics": [
          {
            "Name": "Time",
            "Unit": "Milliseconds",
            "StorageResolution": 60
          }
        ],
        ...
      }
    ]
  },
```

```
  "Time": 1
}
```

Valores de referência

Os valores de referência são valores de string que fazem referência a membros Membros de destino no nó raiz. Essas referências NÃO devem ser confundidas com os ponteiros JSON descritos em RFC6901. Os valores de destino não podem ser aninhados.

Membros de destino

Os destinos válidos DEVEM ser membros no nó raiz e não podem ser objetos aninhados. Por exemplo, um valor _reference_ de `"A.a"` DEVE corresponder ao seguinte membro:

```
{ "A.a" }
```

Ele NÃO DEVE corresponder ao membro aninhado:

```
{ "A": { "a" } }
```

Os valores válidos dos membros de destino dependem do que está fazendo referência a eles. Um destino métrico DEVE ser um valor numérico ou uma matriz de valores numéricos. Os destinos da métrica da matriz numérica NÃO PODEM ter mais de 100 membros. Um destino de dimensão DEVE ter um valor de string.

Exemplo de formato de métricas incorporadas e esquema JSON

Veja a seguir um exemplo válido de formato de métricas incorporadas.

```
{
  "_aws": {
    "Timestamp": 1574109732004,
    "CloudWatchMetrics": [
      {
        "Namespace": "lambda-function-metrics",
        "Dimensions": [["functionVersion"]],
        "Metrics": [
          {
            "Name": "time",
            "Unit": "Milliseconds",
            "StorageResolution": 60
```

```
          }
        ]
      }
    ]
  },
  "functionVersion": "$LATEST",
  "time": 100,
  "requestId": "989ffbf8-9ace-4817-a57c-e4dd734019ee"
}
```

É possível usar o esquema a seguir para validar documentos de formato de métricas incorporadas.

```
{
    "type": "object",
    "title": "Root Node",
    "required": [
        "_aws"
    ],
    "properties": {
        "_aws": {
            "$id": "#/properties/_aws",
            "type": "object",
            "title": "Metadata",
            "required": [
                "Timestamp",
                "CloudWatchMetrics"
            ],
            "properties": {
                "Timestamp": {
                    "$id": "#/properties/_aws/properties/Timestamp",
                    "type": "integer",
                    "title": "The Timestamp Schema",
                    "examples": [
                        1565375354953
                    ]
                },
                "CloudWatchMetrics": {
                    "$id": "#/properties/_aws/properties/CloudWatchMetrics",
                    "type": "array",
                    "title": "MetricDirectives",
                    "items": {
                        "$id": "#/properties/_aws/properties/CloudWatchMetrics/items",
                        "type": "object",
```

```
                        "title": "MetricDirective",
                        "required": [
                            "Namespace",
                            "Dimensions",
                            "Metrics"
                        ],
                        "properties": {
                            "Namespace": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Namespace",
                                "type": "string",
                                "title": "CloudWatch Metrics Namespace",
                                "examples": [
                                    "MyApp"
                                ],
                                "pattern": "^(.*)$",
                                "minLength": 1,
                                "maxLength": 1024
                            },
                            "Dimensions": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Dimensions",
                                "type": "array",
                                "title": "The Dimensions Schema",
                                "minItems": 1,
                                "items": {
                                    "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items",
                                    "type": "array",
                                    "title": "DimensionSet",
                                    "minItems": 0,
                                    "maxItems": 30,
                                    "items": {
                                        "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items/items",
                                        "type": "string",
                                        "title": "DimensionReference",
                                        "examples": [
                                            "Operation"
                                        ],
                                        "pattern": "^(.*)$",
                                        "minLength": 1,
                                        "maxLength": 250
}
```

```
                                }
                            },
                            "Metrics": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Metrics",
                                "type": "array",
                                "title": "MetricDefinitions",
                                "items": {
                                    "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items",
                                    "type": "object",
                                    "title": "MetricDefinition",
                                    "required": [
                                        "Name"
                                    ],
                                    "properties": {
                                        "Name": {
                                            "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Name",
                                            "type": "string",
                                            "title": "MetricName",
                                            "examples": [
                                                "ProcessingLatency"
                                            ],
                                            "pattern": "^(.*)$",
                                            "minLength": 1,
                                            "maxLength": 1024
                                        },
                                        "Unit": {
                                            "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Unit",
                                            "type": "string",
                                            "title": "MetricUnit",
                                            "examples": [
                                                "Milliseconds"
                                            ],
                                            "pattern": "^(Seconds|Microseconds|
Milliseconds|Bytes|Kilobytes|Megabytes|Gigabytes|Terabytes|Bits|Kilobits|Megabits|
Gigabits|Terabits|Percent|Count|Bytes\\/Second|Kilobytes\\/Second|Megabytes\\/Second|
Gigabytes\\/Second|Terabytes\\/Second|Bits\\/Second|Kilobits\\/Second|Megabits\\/
Second|Gigabits\\/Second|Terabits\\/Second|Count\\/Second|None)$"
                                        },
                                        "StorageResolution": {
```

```
                                            "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/StorageResolution",
                                            "type": "integer",
                                            "title": "StorageResolution",
                                            "examples": [
                                                60
                                            ]
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Usar a API PutLogEvents para enviar logs de formato de métricas incorporadas criados manualmente

É possível enviar logs de formato de métrica incorporado ao CloudWatch Logs usando a API PutLogEvents do CloudWatch Logs. Ao chamar PutLogEvents, opcionalmente, você pode incluir o seguinte cabeçalho HTTP para instruir o CloudWatch Logs de que as métricas devem ser extraídas, mas isso não é mais necessário.

```
x-amzn-logs-format: json/emf
```

Veja a seguir um exemplo completo usando o AWS SDK for Java 2.x:

```
package org.example.basicapp;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsRequest;
import software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsResponse;
import software.amazon.awssdk.services.cloudwatchlogs.model.InputLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.PutLogEventsRequest;

import java.util.Collections;
```

```
public class EmbeddedMetricsExample {
        public static void main(String[] args) {

                final String usage = "To run this example, supply a Region code (eg.
 us-east-1), log group, and stream name as command line arguments"
                                + "Ex: PutLogEvents <region-id> <log-group-name>
 <stream-name>";

                if (args.length != 3) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String regionId = args[0];
                String logGroupName = args[1];
                String logStreamName = args[2];

                CloudWatchLogsClient logsClient =
 CloudWatchLogsClient.builder().region(Region.of(regionId)).build();

                // Build a JSON log using the EmbeddedMetricFormat.
                long timestamp = System.currentTimeMillis();
                String message = "{" +
                                "  \"_aws\": {" +
                                "    \"Timestamp\": " + timestamp  + "," +
                                "    \"CloudWatchMetrics\": [" +
                                "      {" +
                                "        \"Namespace\": \"MyApp\"," +
                                "        \"Dimensions\": [[\"Operation\"], [\"Operation
\", \"Cell\"]]," +
                                "        \"Metrics\": [{ \"Name\": \"ProcessingLatency
\", \"Unit\": \"Milliseconds\", \"StorageResolution\": 60 }]" +
                                "      }" +
                                "    ]" +
                                "  }," +
                                "  \"Operation\": \"Aggregator\"," +
                                "  \"Cell\": \"001\"," +
                                "  \"ProcessingLatency\": 100" +
                                "}";
                InputLogEvent inputLogEvent = InputLogEvent.builder()
                        .message(message)
                        .timestamp(timestamp)
                        .build();
```

```
                // Specify the request parameters.
                PutLogEventsRequest putLogEventsRequest = PutLogEventsRequest.builder()
                        .logEvents(Collections.singletonList(inputLogEvent))
                        .logGroupName(logGroupName)
                        .logStreamName(logStreamName)
                        .build();

                logsClient.putLogEvents(putLogEventsRequest);

                System.out.println("Successfully put CloudWatch log event");
        }

}
```

> **Note**
>
> Com o formato de métrica incorporada, você pode acompanhar o processamento de seus logs de EMF por métricas publicadas no namespace `AWS/Logs` da conta. Podem ser usados para rastrear falhas na geração de métricas do EMF, bem como se as falhas ocorrem devido à análise ou validação. Para obter mais informações, consulte Monitorar com métricas do CloudWatch.

Usar o atendente do CloudWatch para enviar logs de formato de métricas incorporadas

Para usar esse método, primeiro instale o atendente do CloudWatch para os serviços dos quais deseja enviar logs de formato de métrica incorporado e, então, você poderá começar a enviar os eventos.

O atendente do CloudWatch deve ser da versão 1.230621.0 ou posterior.

> **Note**
>
> Não é necessário instalar o atendente do CloudWatch para enviar logs de funções do Lambda.

Os tempos limite da função do Lambda não são processados automaticamente. Isso significa que se a função atingir o tempo limite antes de as métricas serem liberadas, as métricas dessa invocação não serão capturadas.

Instalação do atendente do CloudWatch

Instale o atendente do CloudWatch para cada serviço que ´deve enviar logs de formato de métrica incorporado.

Instalar o atendente do CloudWatch no EC2

Primeiro, instale o atendente do CloudWatch na instância. Para obter mais informações, consulte Instalação do atendente do CloudWatch.

Após instalar o atendente, configure-o para escutar em uma porta UDP ou TCP para os logs de formato de métricas incorporadas. Veja a seguir um exemplo dessa configuração que escuta no soquete padrão `tcp:25888`. Para obter mais informações sobre a configuração do atendente, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Instalar o atendente do CloudWatch no Amazon ECS

A maneira mais fácil de implantar o atendente do CloudWatch no Amazon ECS é executá-lo como um arquivo associado, definindo-o na mesma definição de tarefa da aplicação.

Criar um arquivo de configuração do atendente

Crie o arquivo de configuração do atendente do CloudWatch localmente. Neste exemplo, o caminho do arquivo relativo será `amazon-cloudwatch-agent.json`.

Para obter mais informações sobre a configuração do atendente, consulte Criar ou editar manualmente o arquivo de configuração do atendente do CloudWatch.

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Enviar configuração para o SSM Parameter Store

Insira o comando a seguir para enviar o arquivo de configuração do atendente do CloudWatch ao AWS Systems Manager (SSM) Parameter Store.

```
aws ssm put-parameter \
    --name "cwagentconfig" \
    --type "String" \
    --value "`cat amazon-cloudwatch-agent.json`" \
    --region "{{region}}"
```

Configurar a definição da tarefa

Configure a definição de tarefa para usar o atendente do CloudWatch e expor a porta TCP ou UDP. A definição de tarefa de amostra que você deve usar depende do modo de rede.

Observe que o `webapp` especifica a variável de ambiente `AWS_EMF_AGENT_ENDPOINT`. Isso é usado pela biblioteca e deve apontar para o endpoint no qual o atendente está escutando. Além disso, o `cwagent` especifica o `CW_CONFIG_CONTENT` como um parâmetro “valueFrom” que aponta para a configuração do SSM criada na etapa anterior.

Esta seção contém um exemplo para o modo ponte e um exemplo para o modo host ou awsvpc. Para obter mais exemplos de como configurar o atendente do CloudWatch no Amazon ECS, consulte o Repositório de exemplos do Github

Veja a seguir um exemplo do modo de ponte. Quando o modo de ponte de redes está habilitado, o atendente precisa estar vinculado à aplicação usando o parâmetro `links` e deve ser abordado usando o nome do contêiner.

```
{
  "containerDefinitions": [
          {
```

```
            "name": "webapp",
            "links": [ "cwagent" ],
            "image": "my-org/web-app:latest",
            "memory": 256,
            "cpu": 256,
            "environment": [{
              "name": "AWS_EMF_AGENT_ENDPOINT",
              "value": "tcp://cwagent:25888"
            }],
        },
        {
            "name": "cwagent",
            "mountPoints": [],
            "image": "public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest",
            "memory": 256,
            "cpu": 256,
            "portMappings": [{
                "protocol": "tcp",
                "containerPort": 25888
            }],
            "environment": [{
              "name": "CW_CONFIG_CONTENT",
              "valueFrom": "cwagentconfig"
            }],
        }
    ],
}
```

Veja a seguir um exemplo do modo de host ou modo awsvpc. Ao executar esses modos de rede, o atendente pode ser abordado como `localhost`.

```
{
  "containerDefinitions": [
        {
            "name": "webapp",
            "image": "my-org/web-app:latest",
            "memory": 256,
            "cpu": 256,
            "environment": [{
              "name": "AWS_EMF_AGENT_ENDPOINT",
              "value": "tcp://127.0.0.1:25888"
            }],
        },
```

```
            {
                "name": "cwagent",
                "mountPoints": [],
                "image": "public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest",
                "memory": 256,
                "cpu": 256,
                "portMappings": [{
                    "protocol": "tcp",
                    "containerPort": 25888
                }],
                "environment": [{
                  "name": "CW_CONFIG_CONTENT",
                  "valueFrom": "cwagentconfig"
                }],
            }
        ],
}
```

Note

No modo awsvpc, é necessário atribuir um endereço IP público à VPC (somente Fargate), configurar um gateway NAT ou definir um endpoint da VPC do CloudWatch Logs. Para obter mais informações sobre como configurar um NAT, consulte Gateways NAT. Para obter mais informações sobre como configurar um endpoint da VPC do CloudWatch Logs,consulte Usar o CloudWatch Logs com endpoints da VPC de interface.
Veja a seguir um exemplo de como atribuir um endereço IP público a uma tarefa que usa o tipo de inicialização do Fargate.

```
aws ecs run-task \
--cluster {{cluster-name}} \
--task-definition cwagent-fargate \
--region {{region}} \
--launch-type FARGATE \
--network-configuration
 "awsvpcConfiguration={subnets=[{{subnetId}}],securityGroups=[{{sgId}}],assignPublicIp=ENA
```

Verificar permissões

Verifique se a função do IAM que está executando as tarefas tem permissão para ler do SSM Parameter Store. É possível adicionar essa permissão anexando a política AmazonSSMReadOnlyAccess. Para fazer isso, insira o comando a seguir.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonSSMReadOnlyAccess
 \
--role-name CWAgentECSExecutionRole
```

Instalar o atendente do CloudWatch no Amazon EKS

Partes desse processo podem ser puladas caso o CloudWatch Container Insights já esteja instalado nesse cluster.

Permissões

Se você ainda não instalou o Container Insights, primeiro verifique se os nós do Amazon EKS têm as permissões do IAM apropriadas. Eles devem ter a CloudWatchAgentServerPolicy anexada. Para obter mais informações, consulte Verifique os pré-requisitos do .

Criar ConfigMap

Criar um ConfigMap para o atendente O ConfigMap também informa ao atendente para escutar em uma porta TCP ou UDP. Use o ConfigMap a seguir.

```
# cwagent-emf-configmap.yaml
apiVersion: v1
data:
  # Any changes here must not break the JSON format
  cwagentconfig.json: |
    {
      "agent": {
        "omit_hostname": true
      },
      "logs": {
        "metrics_collected": {
          "emf": { }
        }
      }
    }
kind: ConfigMap
metadata:
  name: cwagentemfconfig
```

```
  namespace: default
```

Se o Container Insights já estiver instalado, adicione a linha `"emf": { }` a seguir ao ConfigMap existente.

Aplicar o ConfigMap

Insira o comando a seguir para aplicar o ConfigMap.

```
kubectl apply -f cwagent-emf-configmap.yaml
```

Implantar o atendente

Para implantar o atendente do CloudWatch como um arquivo associado, adicione o atendente à definição do pod, conforme o exemplo a seguir.

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
  namespace: default
spec:
  containers:
    # Your container definitions go here
    - name: web-app
      image: my-org/web-app:latest
    # CloudWatch Agent configuration
    - name: cloudwatch-agent
      image: public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest
      imagePullPolicy: Always
      resources:
        limits:
          cpu: 200m
          memory: 100Mi
        requests:
          cpu: 200m
          memory: 100Mi
      volumeMounts:
        - name: cwagentconfig
          mountPath: /etc/cwagentconfig
      ports:
  # this should match the port configured in the ConfigMap
        - protocol: TCP
```

```
          hostPort: 25888
          containerPort: 25888
  volumes:
    - name: cwagentconfig
      configMap:
        name: cwagentemfconfig
```

Usar o atendente do CloudWatch para enviar logs de formato de métricas incorporadas

Após instalar e executar o atendente do CloudWatch, é possível enviar logs de formato de métrica incorporado por TCP ou UDP. Há dois requisitos ao enviar os logs pelo atendente:

- Os logs devem conter uma chave `LogGroupName` que informa ao atendente qual grupo de logs deve ser usado.
- Cada evento de log deve estar em uma única linha. Em outras palavras, um evento de log não pode conter o caractere de nova linha (\n).

Os eventos de log também devem seguir a especificação do formato de métricas incorporadas. Para obter mais informações, consulte Especificação: formato de métricas incorporadas .

Se você planeja criar alarmes em métricas criadas usando o formato de métricas incorporadas, consulte Configuração de alarmes em métricas criadas com o formato de métricas incorporadas para obter as recomendações.

Veja a seguir um exemplo de como enviar eventos de log manualmente de um shell bash do Linux. Em vez disso, é possível usar as interfaces de soquete UDP fornecidas pela linguagem de programação da sua escolha.

```
echo '{"_aws":{"Timestamp":1574109732004,"LogGroupName":"Foo","CloudWatchMetrics":
[{"Namespace":"MyApp","Dimensions":[["Operation"]],"Metrics":
[{"Name":"ProcessingLatency","Unit":"Milliseconds","StorageResolution":60}]}]},"Operation":"Agg
 \
> /dev/udp/0.0.0.0/25888
```

> **Note**
>
> Com o formato de métrica incorporada, você pode acompanhar o processamento de seus logs de EMF por métricas publicadas no namespace `AWS/Logs` da conta. Podem ser usados

para rastrear falhas na geração de métricas do EMF, bem como se as falhas ocorrem devido à análise ou validação. Para obter mais informações, consulte Monitorar com métricas do CloudWatch.

Explica como usar o formato de métrica incorporado com o AWS Distro for OpenTelemetry

É possível usar o formato de métrica incorporado como parte do projeto OpenTelemetry. O OpenTelemetry é uma iniciativa de código aberto que remove limites e restrições entre formatos específicos do fornecedor para rastreamento, logs e métricas, oferecendo um único conjunto de especificações e APIs. Para obter mais informações, consulte OpenTelemetry.

Usar o formato de métrica incorporado com o OpenTelemetry requer dois componentes: uma origem dos dados compatível com o OpenTelemetry e o AWS Distro para OpenTelemetry Collector habilitado para uso com logs de formato de métrica incorporado do CloudWatch.

Temos redistribuições pré-configuradas dos componentes do OpenTelemetry, mantidas pela AWS, para facilitar ao máximo a integração. Para obter mais informações sobre o uso do OpenTelemetry com formato de métrica incorporado, além de outros produtos da AWS, consulte AWS Distro for OpenTelemetry.

Para obter mais informações sobre compatibilidade de idiomas e uso, consulte Observabilidade da AWS no Github.

Visualizar métricas e logs no console

Após gerar logs de formato de métricas incorporadas que extraem métricas, é possível usar o console do CloudWatch para visualizar as métricas. As métricas incorporadas terão as dimensões especificadas ao gerar os logs. Além disso, as métricas incorporadas geradas usando as bibliotecas de cliente têm as seguintes dimensões padrão:

- ServiceType
- ServiceName
- LogGroup

Como visualizar métricas geradas de logs de formato de métricas incorporadas

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Metrics (Métricas).
3. Selecione um namespace especificado para as métricas incorporadas ao gerá-las. Se você usou bibliotecas de cliente para gerar as métricas e não especificou um namespace, selecione aws-embedded-metrics. Esse é o namespace padrão de métricas incorporadas geradas usando as bibliotecas de cliente.
4. Selecione uma dimensão de métrica (por exemplo, ServiceName).
5. A guia All metrics (Todas as métricas) exibe todas as métricas dessa dimensão no namespace. Você pode fazer o seguinte:

    a. Para classificar a tabela, use o cabeçalho da coluna.
    b. Para criar um gráfico de uma métrica, marque a caixa de seleção ao lado da métrica. Para selecionar todas as métricas, marque a caixa de seleção na linha de cabeçalho da tabela.
    c. Para filtrar por recurso, escolha o ID do recurso e, em seguida, escolha Add to search (Adicionar à pesquisa).
    d. Para filtrar por métrica, selecione o nome da métrica e, em seguida, escolha Add to search (Adicionar à pesquisa).

Consultar logs usando o CloudWatch Logs Insights

É possível consultar os eventos de log detalhados associados às métricas extraídas usando o CloudWatch Logs Insights para fornecer insights mais profundos sobre as causas raízes de eventos operacionais. Um dos benefícios da extração de métricas dos logs é a possibilidade de filtrar os logs posteriormente pela métrica (nome da métrica mais conjunto de dimensões exclusivo) e pelos valores de métrica exclusivos, a fim de obter contexto para os eventos que contribuíram para o valor da métrica agregada.

Por exemplo, para obter o ID de uma solicitação afetada ou o ID de um rastreamento do X-Ray, você pode executar a consulta a seguir no CloudWatch Logs Insights.

```
filter Latency > 1000 and Operation = "Aggregator"
| fields RequestId, TraceId
```

Também é possível executar a agregação de tempo de consulta em chaves de alta cardinalidade, como localizar os clientes afetados por um evento. Isso é ilustrado no exemplo a seguir.

```
filter Latency > 1000 and Operation = "Aggregator"
| stats count() by CustomerId
```

Para obter mais informações, consulte Analisar os dados de log com o CloudWatch Logs Insights

Configuração de alarmes em métricas criadas com o formato de métricas incorporadas

Em geral, a criação de alarmes em métricas geradas pelo formato de métricas incorporadas segue o mesmo padrão da criação de alarmes em qualquer outra métrica. Para obter mais informações, consulte Usar alarmes do Amazon CloudWatch.

A geração de métricas no formato de métricas incorporadas depende do seu fluxo de publicação de logs, pois os logs precisam ser processados pelo CloudWatch Logs para serem transformados em métricas. Portanto, é importante que você publique os logs em tempo hábil para que seus pontos de dados de métricas sejam criados dentro do período em que os alarmes são avaliados.

Se você planeja usar o formato de métricas incorporadas para enviar métricas de alta resolução e criar alarmes para sobre essas métricas, recomendamos que você transfira os logs para o CloudWatch Logs em um intervalo de 5 segundos ou menos para evitar a introdução de atrasos adicionais que possam causar alarmes em dados parciais ou ausentes. Se você estiver usando o agente do CloudWatch, poderá ajustar o intervalo de descarga definindo o parâmetro `force_flush_interval` no arquivo de configuração do agente do CloudWatch. O padrão para esse valor é de 5 segundos.

Se você estiver usando o Lambda em outras plataformas nas quais não seja possível controlar o intervalo de liberação dos logs, considere usar alarmes “M de N” para controlar o número de pontos de dados usados para acionar o alarme. Para obter mais informações, consulte Avaliar um alarme.

Produtos da AWS que publicam métricas do CloudWatch

Os seguintes produtos da AWS publicam métricas no CloudWatch. Para obter mais informações sobre as métricas e dimensões, consulte a documentação especificada.

Serviço	Namespace	Documentação
AWS Amplify	`AWS/AmplifyHosting`	Monitoramento
Amazon API Gateway	`AWS/ApiGateway`	Monitorar a execução da API com o Amazon CloudWatch
Amazon AppFlow	`AWS/AppFlow`	Monitorar o Amazon AppFlow com o Amazon CloudWatch
AWS Application Migration Service	`AWS/MGN`	Monitoramento do Application Migration Service com o Amazon CloudWatch
AWS App Runner	`AWS/AppRunner`	Visualizar métricas de serviço do App Runner relatadas ao CloudWatch
AppStream 2.0	`AWS/AppStream`	Monitoramento dos recursos do Amazon AppStream 2.0
AWS AppSync	`AWS/AppSync`	Métricas do CloudWatch
Amazon Athena	`AWS/Athena`	Monitorar consultas do Athena com métricas do CloudWatch
Amazon Aurora	`AWS/RDS`	Métricas do Amazon Aurora
AWS Backup	`AWS/Backup`	Monitorar métricas do AWS Backup com o CloudWatch
Amazon Bedrock	`AWS/Bedrock`	Monitoramento do Amazon Bedrock com o Amazon CloudWatch

Serviço	Namespace	Documentação
AWS Billing and Cost Management	`AWS/Billing`	Como monitorar as cobranças com alertas e notificações
Amazon Braket	`AWS/Braket/By Device`	Monitoramento do Amazon Braket com o Amazon CloudWatch
AWS Certificate Manager	`AWS/CertificateManager`	Métricas compatíveis do CloudWatch
CA privada da AWS	`AWS/ACMPrivateCA`	Métricas compatíveis do CloudWatch
AWS Chatbot	`AWS/Chatbot`	Monitoramento do AWS Chatbot com Amazon CloudWatch
Amazon Chime	`AWS/ChimeVoiceConnector`	Monitoramento do Amazon Chime com o Amazon CloudWatch
SDK do Amazon Chime	`AWS/ChimeSDK`	Métricas de serviço
AWS Client VPN	`AWS/ClientVPN`	Monitoramento com Amazon CloudWatch
Amazon CloudFront	`AWS/CloudFront`	Monitoramento de atividade do CloudFront usando o CloudWatch
AWS CloudHSM	`AWS/CloudHSM`	Obter métricas do CloudWatch
Amazon CloudSearch	`AWS/CloudSearch`	Monitoramento de um domínio do Amazon CloudSearch com o Amazon CloudWatch

Serviço	Namespace	Documentação
AWS CloudTrail	`AWS/CloudTrail`	Métricas do CloudWatch compatíveis
Agente do CloudWatch	`CWAgent` ou um namespace personalizado	Métricas coletadas pelo atendente do CloudWatch
Transmissões de métricas do CloudWatch	`AWS/CloudWatch/MetricStreams`	Monitorar seus fluxos de métrica com métricas do CloudWatch
CloudWatch RUM	`AWS/RUM`	Métricas do CloudWatch que você pode coletar com o CloudWatch RUM
CloudWatch Synthetics	`CloudWatchSynthetics`	Métricas do CloudWatch publicadas por canaries
Amazon CloudWatch Logs	`AWS/Logs`	Monitoramento do uso com métricas do CloudWatch
AWS CodeBuild	`AWS/CodeBuild`	Monitorar o AWS CodeBuild
Amazon CodeGuru Reviewer		Monitoraramento do CodeGuru Reviewer com o Amazon CloudWatch
Amazon Kendra		Monitoramento do Amazon Kendra com o Amazon CloudWatch
Amazon CodeWhisperer	`AWS/CodeWhisperer`	Monitoramento do Amazon CodeWhisperer com o Amazon CloudWatch
Amazon Cognito	`AWS/Cognito`	Monitoramento do Amazon Cognito

Serviço	Namespace	Documentação
Amazon Comprehend	`AWS/Comprehend`	Monitoramento de endpoints do Amazon Comprehend
AWS Config	`AWS/Config`	Métricas de uso e sucesso do AWS Config
Amazon Connect	`AWS/Connect`	Monitoramento do Amazon Connect com métricas do Amazon CloudWatch
Amazon Data Lifecycle Manager	`AWS/DataLifecycleManager`	Monitorar políticas usando o Amazon CloudWatch
AWS DataSync	`AWS/DataSync`	Monitoramento de sua tarefa
Amazon DataZone		Monitoramento do Amazon DataZone com o Amazon CloudWatch
Amazon DevOps Guru	`AWS/DevOps-Guru`	Monitoramento do Amazon DevOps Guru com o Amazon CloudWatch
AWS Database Migration Service	`AWS/DMS`	Monitoramento de tarefas do AWS DMS
AWS Direct Connect	`AWS/DX`	Monitoramento com Amazon CloudWatch
AWS Directory Service	`AWS/DirectoryService`	Use as métricas do Amazon CloudWatch para determinar quando adicionar controladores de domínio
Amazon DocumentDB	`AWS/DocDB`	Métricas do Amazon DocumentDB
Amazon DynamoDB	`AWS/DynamoDB`	Métricas e dimensões do DynamoDB

Serviço	Namespace	Documentação
DynamoDB Accelerator (DAX)	AWS/DAX	Visualizar métricas e dimensões do DAX
Amazon EC2	AWS/EC2	Monitoramento de suas instâncias usando o CloudWatch
Amazon EC2 Elastic Graphics	AWS/ElasticGPUs	Usar métricas do CloudWatch para monitorar o Elastic Graphics
Frota spot do Amazon EC2	AWS/EC2Spot	Métricas do CloudWatch para frota spot
Amazon EC2 Auto Scaling	AWS/AutoScaling	Monitorar seus grupos e instâncias do Auto Scaling usando o CloudWatch
AWS Elastic Beanstalk	AWS/ElasticBeanstalk	Publicação de métricas personalizadas do Amazon CloudWatch para um ambiente
Amazon Elastic Block Store	AWS/EBS	Métricas do Amazon CloudWatch para o Amazon EBS
Amazon Elastic Container Registry	AWS/ECR	Métricas do repositório do Amazon ECR
Amazon Elastic Container Service	AWS/ECS	Métricas do Amazon ECS no CloudWatch
Amazon ECS por meio do CloudWatch Container Insights	ECS/ContainerInsights	Métricas do Amazon ECS Container Insights

Serviço	Namespace	Documentação
Autoescalabilidade de clusters do Amazon ECS	`AWS/ECS/ManagedScaling`	Autoescalabilidade de cluster do Amazon ECS
AWS Elastic Disaster Recovery		Métricas do CloudWatch para DRS
Amazon Elastic File System	`AWS/EFS`	Monitoramento com CloudWatch
Amazon Elastic Inference	`AWS/ElasticInference`	Como usar métricas do CloudWatch para monitorar o Amazon Elastic Inference
Amazon EKS por meio do CloudWatch Container Insights	`ContainerInsights`	Métricas do Amazon EKS e do Kubernetes Container Insights
Elastic Load Balancing	`AWS/ApplicationELB`	Métricas do CloudWatch para seu Application Load Balancer
Elastic Load Balancing	`AWS/NetworkELB`	Métricas do CloudWatch para o Network Load Balancer
Elastic Load Balancing	`AWS/GatewayELB`	Métricas do CloudWatch para seu Gateway Load Balancer
Elastic Load Balancing	`AWS/ELB`	Métricas do CloudWatch para seu Classic Load Balancer
Amazon Elastic Transcoder	`AWS/ElasticTranscoder`	Monitoramento com Amazon CloudWatch

Serviço	Namespace	Documentação
Amazon ElastiCache para Memcached	`AWS/ElastiCache`	Monitoramento do uso com métricas do CloudWatch
Amazon ElastiCache para Redis	`AWS/ElastiCache`	Monitoramento do uso com métricas do CloudWatch
Amazon OpenSearch Service	`AWS/ES`	Monitorar métricas de cluster do OpenSearch com o Amazon CloudWatch
Amazon EMR	`AWS/ElasticMapReduce`	Monitorar métricas de armazenamento com o CloudWatch
AWS Elemental MediaConnect	`AWS/MediaConnect`	Monitorar o MediaConnect com o Amazon CloudWatch
AWS Elemental MediaConvert	`AWS/MediaConvert`	Como usar as métricas do CloudWatch para visualizar métricas para Recursos do AWS Elemental MediaConvert
AWS Elemental MediaLive	`AWS/MediaLive`	Monitorar atividades usando métricas do Amazon CloudWatch
AWS Elemental MediaPackage	`AWS/MediaPackage`	Monitorar o AWS Elemental MediaPackage com métricas do Amazon CloudWatch
AWS Elemental MediaStore	`AWS/MediaStore`	Monitorar o AWS Elemental MediaStore com métricas do Amazon CloudWatch
AWS Elemental MediaTailor	`AWS/MediaTailor`	Monitoramento do AWS Elemental MediaTailor com Amazon CloudWatch
Amazon EventBridge	`AWS/Events`	Monitorar o Amazon EventBridge

Serviço	Namespace	Documentação
Amazon FinSpace		Registro em logs e monitoramento
Amazon Forecast		Métricas do CloudWatch para o Amazon Forecast
Amazon Fraud Detector		Monitoramento do Amazon Fraud Detector com o Amazon CloudWatch
Amazon FSx para Lustre	`AWS/FSx`	Monitorar o Amazon FSx for Lustre
Amazon FSx for OpenZFS	`AWS/FSx`	Monitoramento com Amazon CloudWatch
Amazon FSx para Windows File Server	`AWS/FSx`	Monitorar o Amazon FSx for Windows File Server
Amazon FSx for NetApp ONTAP	`AWS/FSx`	Monitoramento com Amazon CloudWatch
Amazon FSx for OpenZFS	`AWS/FSx`	Monitoramento com Amazon CloudWatch
Amazon GameLift	`AWS/GameLift`	Monitorar o Amazon GameLift com o CloudWatch
AWS Global Accelerator	`AWS/GlobalAccelerator`	Usar métricas do Amazon CloudWatch com AWS Global Accelerator
AWS Glue	`Glue`	Monitorar o AWS Glue usando as métricas do CloudWatch
AWS Ground Station	`AWS/GroundStation`	Métricas usando o Amazon CloudWatch

Serviço	Namespace	Documentação
AWS HealthLake	`AWS/HealthLake`	Monitorar o HealthLake com o CloudWatch
Amazon Inspector	`AWS/Inspector`	Monitorar o Amazon Inspector usando o CloudWatch
Amazon Interactive Video Service	`AWS/IVS`	Monitoramento do Amazon IVS com o Amazon CloudWatch
Amazon Interactive Video Service Chat	`AWS/IVSChat`	Monitoramento do Amazon IVS com o Amazon CloudWatch
AWS IoT	`AWS/IoT`	Métricas e dimensões do AWS IoT
AWS IoT Analytics	`AWS/IoTAnalytics`	Namespaces, métricas e dimensões
AWS IoT FleetWise	`AWS/IoTFleetWise`	Monitorar o AWS IoT FleetWise com o Amazon CloudWatch
AWS IoT SiteWise	`AWS/IoTSiteWise`	Monitorar o AWS IoT SiteWise com métricas do Amazon CloudWatch
AWS IoT TwinMaker	`AWS/IoTTwinMaker`	Monitorar o AWS IoT TwinMaker com métricas do Amazon CloudWatch
AWS IoT com um clique		Monitoramento do AWS IoT 1-Click com o Amazon CloudWatch
AWS Key Management Service	`AWS/KMS`	Monitoramento com CloudWatch

Serviço	Namespace	Documentação
Amazon Keyspaces (para Apache Cassandra)	`AWS/Cassandra`	Métricas e dimensões do Amazon Keyspaces
Amazon Kendra		Monitoramento do Amazon Kendra com o Amazon CloudWatch
Amazon Managed Service for Apache Flink	`AWS/KinesisAnalytics`	Serviço gerenciado para Apache Flink para aplicações SQL: Monitoramento com o CloudWatch Serviço gerenciado para Apache Flink para Apache Flink: Visualização de métricas e dimensões do Amazon Managed Service for Apache Flink
Amazon Data Firehose	`AWS/Firehose`	Monitoring Firehose Using CloudWatch Metrics
Amazon Kinesis Data Streams	`AWS/Kinesis`	Monitorar o Amazon Kinesis Data Streams com o Amazon CloudWatch
Amazon Kinesis Video Streams	`AWS/KinesisVideo`	Monitorar métricas do Kinesis Video Streams com o CloudWatch
AWS Lambda	`AWS/Lambda`	Métricas do AWS Lambda
Amazon Lex	`AWS/Lex`	Monitorar o Amazon Lex com o Amazon CloudWatch

AWS License Manager	`AWSLicenseManager/licenseUsage`
	
	`AWS/LicenseManager/LinuxSubscriptions`
		Monitorar o uso de licenças com o Amazon CloudWatch
		
		Métricas de uso e alarmes do Amazon CloudWatch para assinaturas do Linux

Serviço	Namespace	Documentação
Amazon Location Service	`AWS/Location`	Métricas do Amazon Location Service exportadas para o Amazon CloudWatch
Amazon Lookout for Equipment	`AWS/lookoutequipment`	Monitoramento do Lookout for Equipment com o Amazon CloudWatch
Amazon Lookout for Metrics	`AWS/LookoutMetrics`	Monitoramento Lookout for Metrics com o Amazon CloudWatch
Amazon Lookout for Vision	`AWS/LookoutVision`	Monitoramento do Lookout for Metrics com o Amazon CloudWatch
AWS Mainframe Modernization		Monitoramento do AWS Mainframe Modernization com o Amazon CloudWatch
Amazon Machine Learning	`AWS/ML`	Monitorar o Amazon ML com métricas do CloudWatch
Amazon Managed Blockchain	`AWS/managedblockchain`	Use as métricas do Hyperledger Fabric Peer Node no Amazon Managed Blockchain
Amazon Managed Service para Prometheus	`AWS/Prometheus`	Métricas do Amazon CloudWatch
Amazon Managed Streaming for Apache Kafka	`AWS/Kafka`	Monitorar o Amazon MSK com o Amazon CloudWatch

Serviço	Namespace	Documentação
Amazon Managed Streaming for Apache Kafka	`AWS/Kafka Connect`	Monitoramento do MSK Connect
Amazon Managed Workflows for Apache Airflow	`AWS/MWAA`	Container, queue, and database metrics for Amazon MWAA
Amazon MemoryDB for Redis	`AWS/MemoryDB`	Monitoramento das métricas do CloudWatch
Amazon MQ	`AWS/AmazonMQ`	Monitorar atendentes do Amazon MQ usando o Amazon CloudWatch
Amazon Neptune	`AWS/Neptune`	Monitoramento do Neptune com o CloudWatch
AWS Network Firewall	`AWS/NetworkFirewall`	Métricas do AWS Network Firewall no Amazon CloudWatch
Gerenciador de rede AWS	`AWS/NetworkManager`	Métricas do CloudWatch para recursos on-premises
Amazon Nimble Studio	`AWS/NimbleStudio`	Monitoring Nimble Studio with Amazon CloudWatch (Monitoramento do Nimble Studio com o Amazon CloudWatch)
AWS HealthOmics	`AWS/Omics`	Monitoramento do AWS HealthOmics com o Amazon CloudWatch
AWS OpsWorks	`AWS/OpsWorks`	Monitorar pilhas usando o Amazon CloudWatch

Serviço	Namespace	Documentação
AWS Outposts	`AWS/Outposts`	Métricas do CloudWatch para AWS Outposts
AWS Panorama	`AWS/PanoramaDeviceMetrics`	Monitoramento de dispositivos e aplicações com o Amazon CloudWatch
Amazon Personalize	`AWS/Personalize`	Métricas do CloudWatch para o Amazon Personalize
Amazon Pinpoint	`AWS/Pinpoint`	Visualizar métricas do Amazon Pinpoint no CloudWatch
Amazon Polly	`AWS/Polly`	Integrar o CloudWatch ao Amazon Polly
AWS PrivateLink	`AWS/PrivateLinkEndpoints`	Métricas do CloudWatch para AWS PrivateLink
AWS PrivateLink	`AWS/PrivateLinkServices`	Métricas do CloudWatch para AWS PrivateLink
AWS Private 5G	`AWS/Private5G`	Métricas do Amazon CloudWatch
Amazon QLDB	`AWS/QLDB`	Monitorar dados no Amazon QuickSight
Amazon QuickSight	`AWS/QuickSight`	Monitoramento com Amazon CloudWatch
Amazon Redshift	`AWS/Redshift`	Dados de performance do Amazon Redshift

Serviço	Namespace	Documentação
Amazon Relational Database Service	AWS/RDS	Monitorar métricas do Amazon RDS com o Amazon CloudWatch
Amazon Rekognition	AWS/Rekognition	Monitoramento do Rekognition com o Amazon CloudWatch
AWS re:Post Privado	AWS/rePostPrivate	Monitorar o AWS re:Post Privado com o Amazon CloudWatch
AWS RoboMaker	AWS/Robomaker	Monitorar o AWS RoboMaker com o Amazon CloudWatch
Amazon Route 53	AWS/Route53	Monitorar o Amazon Route 53
Controlador de recuperação de aplicações do Route 53	AWS/Route53RecoveryReadiness	Usar o Amazon CloudWatch com o controlador de recuperação de aplicações
Amazon SageMaker	AWS/SageMaker	Monitorar o SageMaker com o CloudWatch
Pipelines de criação de modelos do Amazon SageMaker	AWS/SageMaker/ModelBuildingPipeline	Métricas de pipelines do SageMaker
AWS Secrets Manager	AWS/SecretsManager	Monitorar o Secrets Manager com o Amazon CloudWatch
Amazon Security Lake	AWS/SecurityLake	Métricas do CloudWatch para o Amazon Security Lake

Serviço	Namespace	Documentação
Service Catalog	`AWS/ServiceCatalog`	Métricas do CloudWatch para o Service Catalog
AWS Shield Advanced	`AWS/DDoSProtection`	Monitoramento com CloudWatch
Amazon Simple Email Service	`AWS/SES`	Recuperação de dados de eventos do Amazon SES a partir do CloudWatch
AWS SimSpace Weaver	`AWS/simspaceweaver`	Monitoramento do AWS SimSpace Weaver com o Amazon CloudWatch
Amazon Simple Notification Service	`AWS/SNS`	Monitoramento do Amazon SNS com o CloudWatch
Amazon Simple Queue Service	`AWS/SQS`	Monitoramento de filas do Amazon SQS usando o CloudWatch
Amazon S3	`AWS/S3`	Monitoramento de métricas com o Amazon CloudWatch
Lente de Armazenamento do S3	`AWS/S3/Storage-Lens`	Monitorar métricas do S3 Storage Lens no CloudWatch
Amazon Simple Workflow Service	`AWS/SWF`	Métricas do Amazon SWF para o CloudWatch
AWS Step Functions	`AWS/States`	Monitoramento do Step Functions usando o CloudWatch
AWS Storage Gateway	`AWS/StorageGateway`	Usar métricas do Amazon CloudWatch

Serviço	Namespace	Documentação
Run Command do AWS Systems Manager	`AWS/SSM-RunCommand`	Monitorar métricas do Run Command usando o CloudWatch
Amazon Textract	`AWS/Textract`	Métricas do CloudWatch para o Amazon Textract
Amazon Timestream	`AWS/Timestream`	Métricas e dimensões do Timestream
AWS Transfer for SFTP	`AWS/Transfer`	Métricas do CloudWatch do AWS SFTP
Amazon Transcribe	`AWS/Transcribe`	Monitoramento do Amazon Transcribe com o Amazon CloudWatch
Amazon Translate	`AWS/Translate`	Métricas e dimensões do CloudWatch para o Amazon Translate
AWS Trusted Advisor	`AWS/TrustedAdvisor`	Criação de alarmes do Trusted Advisor com o CloudWatch
Amazon VPC	`AWS/NATGateway`	Monitorar seu gateway NAT com o CloudWatch
Amazon VPC	`AWS/TransitGateway`	Métricas do CloudWatch para seus Transit Gateways
Amazon VPC	`AWS/VPN`	Monitoramento com CloudWatch
IP Address Manager da Amazon VPC	`AWS/IPAM`	Criar alarmes com o Amazon CloudWatch

Serviço	Namespace	Documentação
AWS WAF	`AWS/WAFV2` para recursos do AWS WAF `WAF` para recursos clássicos do AWS WAF	Monitoramento com CloudWatch
Amazon WorkMail	`AWS/WorkMail`	Monitoramento do Amazon WorkMail com o Amazon CloudWatch
Amazon WorkSpaces	`AWS/WorkSpaces`	Monitorar seu WorkSpaces usando as métricas do CloudWatch
Amazon WorkSpaces Web	`AWS/WorkSpacesWeb`	Monitorar o Amazon WorkSpaces Web com o Amazon CloudWatch

Métricas de uso do AWS

O CloudWatch coleta métricas que rastreiam o uso de alguns recursos e APIs da AWS. Essas métricas estão publicadas no namespace `AWS/Usage`. As métricas de uso no CloudWatch permitem gerenciar o uso visualizando proativamente métricas no console do CloudWatch, criando painéis personalizados, detectando alterações na atividade com a detecção de anomalias do CloudWatch e configurando alarmes que alerram quando o uso se aproxima de um limite.

Alguns produtos da AWS integram essas métricas de uso com o Service Quotas. Para esses serviços, é possível usar o CloudWatch para gerenciar suas cotas de serviço utilizadas por sua conta. Para ter mais informações, consulte Visualizar as Service Quotas e definir alarmes.

Tópicos

Visualizar as Service Quotas e definir alarmes

Em alguns produtos da AWS, é possível usar essas métricas para visualizar o uso do produto atual nos grafos e painéis do CloudWatch. É possível usar uma função matemática métrica do CloudWatch para exibir as cotas de serviço desses recursos nos gráficos. Também é possível configurar alarmes que alertam você quando o uso se aproxima de uma cota de serviço. Para obter mais informações sobre Service Quotas, consulte O que são cotas de serviço no Guia do usuário de cotas de serviço.

Se tiver feito login em uma conta que esteja configurada como uma conta de monitoramento na observabilidade entre contas do CloudWatch, será possível usar essa conta de monitoramento para visualizar as cotas de serviço e definir alarmes para métricas nas contas de origem vinculadas a essa conta de monitoramento. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

Atualmente, os seguintes produtos integram suas métricas de uso ao Service Quotas:

- AWS CloudHSM
- Amazon Chime SDK
- Amazon CloudWatch

- Amazon CloudWatch Logs
- Amazon DynamoDB
- Amazon EC2
- Amazon Elastic Container Registry
- Elastic Load Balancing
- AWS Fargate
- AWS Fault Injection Service
- AWS Interactive Video Service
- AWS Key Management Service
- Amazon Data Firehose
- Amazon Location Service
- Consulta ao Amazon Managed Blockchain (AMB)
- AWS RoboMaker
- Amazon SageMaker

Como visualizar uma cota de serviço e opcionalmente definir um alarme

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Na guia Todas as métricas, escolha Uso e Por recurso da AWS.

   A lista das métricas de uso da cota de serviço é exibida.
4. Marque a caixa de seleção ao lado de uma das métricas.

   O gráfico exibe o uso atual desse recurso da AWS.
5. Para adicionar a cota de serviço ao gráfico, faça o seguinte:

   a. Escolha a guia Graphed metrics (Métricas em gráfico).

   b. Selecione Math expression (Expressão matemática), Start with an empty expression (Começar com uma expressão vazia). Na nova linha, em Details (Detalhes), insira **`SERVICE_QUOTA(m1)`**.

      Uma nova linha é adicionada ao gráfico, exibindo a cota de serviço do recurso representado na métrica.

6. Para ver o uso atual como uma porcentagem da cota, adicione uma nova expressão ou altere a expressão SERVICE_QUOTA atual. A nova expressão a ser usada é **`m1/SERVICE_QUOTA(m1)*100`**.
7. (Opcional) Para definir um alarme que notifique se você caso se aproxime da cota de serviço, faça o seguinte:

   a. Na linha **`m1/SERVICE_QUOTA(m1)*100`**, em Actions (Ações), escolha o ícone de alarme. Ele se parece com um sino.

      A página de criação de alarmes é exibida.

   b. Em Conditions (Condições), verifique se o Threshold type (Tipo de limite) é Static (Estático) e se Whenever Expression1 is (Sempre que a Expression1 for) esteja definido como Greater (Maior). Em than (que), insira **`80`**. Isso cria um alarme que entrará no estado ALARM (ALARME) quando seu uso exceder 80% da cota.

   c. Escolha Próximo.

   d. Na próxima página, selecione um tópico do Amazon SNS ou crie um e escolha Next (Próximo). O tópico que você selecionar será notificado quando o alarme entrar no estado ALARM.

   e. Na próxima página, insira um nome e uma descrição para o alarme e selecione Next (Próximo).

   f. Selecione Criar alarme.

Métricas de uso de API da AWS

A maioria das APIs compatíveis com o registro do AWS CloudTrail também relata métricas de uso ao CloudWatch. As métricas de uso de API no CloudWatch permitem gerenciar o uso da API visualizando proativamente métricas no console do CloudWatch, criando painéis personalizados, detectando alterações na atividade com a detecção de anomalias do CloudWatch e configurando alarmes que alerram quando o uso se aproxima de um limite.

A tabela a seguir lista os produtos que relatam métricas de uso de API para o CloudWatch e o valor a ser usado para a dimensão `Service` para visualizar as métricas de uso desse produto.

Serviço	Valor da dimensão **`Service`**
AWS Identity and Access Management Access Analyzer	`Access Analyzer`
AWS Account Management	`Account Management`
Alexa for Business	`A4B`
Amazon API Gateway	`API Gateway`
AWS App Mesh	`App Mesh`
AWS AppConfig	`AWS AppConfig`
Amazon AppFlow	`AppFlow`
Application Auto Scaling	`Application Auto Scaling`
Application Discovery Service	`Application Discovery Service`
Amazon AppStream	`AppStream`
Image Builder do AppStream 2.0	`Image Builder`
Amazon Athena	`Athena`
AWS Audit Manager	`Audit Manager`
AWS Backup	`Backup`
AWS Batch	`Batch`
Amazon Braket	`Braket`
Orçamentos da AWS	`Budgets`
AWS Certificate Manager	`Certificate Manager`
Amazon Chime SDK	`ChimeSDK`

Serviço	Valor da dimensão **`Service`**
Amazon Cloud Directory	`Cloud Directory`
AWS Cloud Map	`Cloud Map`
AWS CloudFormation	`CloudFormation`
AWS CloudHSM	`CloudHSM`
Amazon CloudSearch	`CloudSearch`
AWS CloudShell	`CloudShell`
AWS CloudTrail	`CloudTrail`
Amazon CloudWatch	`CloudWatch`
Amazon CloudWatch Logs	`Logs`
Amazon CloudWatch Application Insights	`CloudWatch Application Insights`
AWS CodeBuild	`CodeBuild`
AWS CodeCommit	`CodeCommit`
Amazon CodeGuru Profiler	`CodeGuru Profiler`
AWS CodePipeline	`CodePipeline`
AWS CodeStar	`CodeStar`
AWS CodeStar Notifications	`CodeStar Notifications`
Conexões do AWS CodeStar	`CodeStar Connections`
Grupos de identidade do Amazon Cognito	`Cognito Identity Pools`
Amazon Cognito Sync	`Cognito Sync`
Amazon Comprehend	`Comprehend`

Serviço	Valor da dimensão **`Service`**
Amazon Comprehend Medical	`Comprehend Medical`
AWS Compute Optimizer	`ComputeOptimzier`
Amazon Connect	`Connect`
Amazon Connect Customer Profiles	`Customer Profiles`
AWS Cost and Usage Reports	`Cost and Usage Report`
AWS Cost Explorer	`Cost Explorer`
AWS Data Exchange	`Data Exchange`
AWS Data Lifecycle Manager	`Data Lifecycle Manager`
AWS Database Migration Service	`Database Migration Service`
AWS DataSync	`DataSync`
AWS DeepLens	`AWS DeepLens`
Amazon Detective	`Detective`
Device Advisor	`Device Advisor`
AWS Direct Connect	`Direct Connect`
AWS Directory Service	`Directory Service`
DynamoDB Accelerator	`DynamoDBAccelerator`
Amazon EC2	`EC2`
Ajuste de escala automático do EC2	`EC2 Auto Scaling`
Amazon Elastic Container Registry	`ECR Public`
Amazon Elastic Container Service	`ECS`

Serviço	Valor da dimensão **`Service`**
Amazon Elastic File System	`EFS`
Amazon Elastic Kubernetes Service	`EKS`
AWS Elastic Beanstalk	`Elastic Beanstalk`
Amazon Elastic Inference	`Elastic Inference`
Elastic Load Balancing	`Elastic Load Balancing`
Amazon EMR	`EMR Containers`
AWS Firewall Manager	`Firewall Manager`
Amazon FSx	`FSx`
Amazon GameLift	`GameLift`
AWS Glue DataBrew	`DataBrew`
Amazon Managed Grafana	`Grafana`
AWS IoT Greengrass	`Greengrass`
AWS Ground Station	`Ground Station`
APIs AWS Health e notificações	`AWS Health APIs And Notifications`
Amazon Interactive Video Service	`IVS`
AWS IoT Core	`IoT`
AWS IoT com um clique	`IoT 1-Click`
AWS IoT Events	`IoT Events`
AWS IoT RoboRunner	`IoT RoboRunner`
AWS IoT SiteWise	`IoT Sitewise`

Serviço	Valor da dimensão **`Service`**
AWS IoT Wireless	`IoT Wireless`
Amazon Kendra	`Kendra`
Amazon Keyspaces (para Apache Cassandra)	`Keyspaces`
Amazon Managed Service for Apache Flink	`Kinesis Analytics`
Amazon Data Firehose	`Firehose`
Kinesis Video Streams	`Kinesis Video Streams`
AWS Key Management Service	`KMS`
AWS Lambda	`Lambda`
AWS Launch Wizard	`Launch Wizard`
Amazon Lex	`Amazon Lex`
Amazon Lightsail	`Lightsail`
Amazon Location Service	`Location`
Amazon Lookout for Vision	`Lookout for Vision`
Amazon Machine Learning	`Amazon Machine Learning`
Amazon Macie	`Macie`
Consulta ao Amazon Managed Blockchain (AMB)	`Amazon Managed Blockchain Query`
AWS Managed Services	`AWS Managed Services`
AWS Marketplace Commerce Analytics	`Marketplace Analytics Service`
AWS Elemental MediaConnect	`MediaConnect`
AWS Elemental MediaConvert	`MediaConvert`

Serviço	Valor da dimensão **`Service`**
AWS Elemental MediaLive	`MediaLive`
AWS Elemental MediaStore	`Mediastore`
AWS Elemental MediaTailor	`MediaTailor`
AWS Mobile Hub	`Mobile Hub`
AWS Network Firewall	`Network Firewall`
AWS OpsWorks	`OpsWorks`
AWS OpsWorks para gerenciamento de configuração	`OPsWorks CM`
AWS Outposts	`Outposts`
AWS Organizations	`Organizations`
Insights de Performance do Amazon RDS	`Performance Insights`
Amazon Pinpoint	`Pinpoint`
AWS Private Certificate Authority	`Private Certificate Authority`
Amazon Managed Service para Prometheus	`Prometheus`
AWS Proton	`Proton`
Amazon Quantum Ledger Database (Amazon QLDB)	`QLDB`
Amazon RDS	`RDS`
Amazon Redshift	`Redshift Data API`
Amazon Rekognition	`Rekognition`
AWS Resource Access Manager	`Resource Access Manager`

Serviço	Valor da dimensão **`Service`**
AWS Resource Groups	`Resource Groups`
AWS Resource Groups Tagging API	`Resource Groups Tagging API`
AWS RoboMaker	`RoboMaker`
Domínios do Amazon Route 53	`Route 53 Domains`
Amazon Route 53 Resolver	`Route 53 Resolver`
Amazon S3	`S3`
Amazon S3 Glacier	`Amazon S3 Glacier`
Runtime do Amazon SageMaker	`Sagemaker`
Savings Plans	`Savings Plans`
AWS Secrets Manager	`Secrets Manager`
AWS Security Hub	`Security Hub`
AWS Server Migration Service	`AWS Server Migration Service`
AWS Service Catalog AppRegistry	`Service Catalog AppRegistry`
Service Quotas	`Service Quotas`
AWS Shield	`Shield`
AWS Signer	`Signer`
Amazon Simple Notification Service	`SNS`
Amazon Simple Email Service	`SES`
Amazon Simple Queue Service	`SQS`
Armazenamento de identidades	`Identity Store`

Serviço	Valor da dimensão **`Service`**
Storage Gateway	`Storage Gateway`
AWS Support	`Support`
Amazon Simple Workflow Service	`SWF`
Amazon Textract	`Textract`
AWS IoT Things Graph	`ThingsGraph`
Amazon Timestream	`Timestream`
Amazon Transcribe	`Transcribe`
Amazon Translate	`Translate`
Transcrição de transmissão do Amazon Transcribe	`Transcribe Streaming`
AWS Transfer Family	`Transfer`
AWS WAF	`WAF`
Amazon WorkDocs	`Amazon WorkDocs`
Amazon WorkLink	`WorkLink`
Amazon WorkMail	`Amazon WorkMail`
Amazon WorkSpaces	`Workspaces`
AWS X-Ray	`X-Ray`

Alguns serviços relatam métricas de uso para outras APIs também. Para ver se uma API relata métricas de uso ao CloudWatch, use o console do CloudWatch para ver as métricas relatadas por esse serviço no namespace `AWS/Usage`.

Para ver a lista de APIs de um serviço que relatam métricas de uso ao CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Na guia Todas as métricas, escolha Uso e Por recurso da AWS.
4. Na caixa de pesquisa próxima à lista de métricas, insira o nome do serviço. As métricas serão filtradas pelo serviço inserido.

Métricas de uso do CloudWatch

O CloudWatch coleta métricas que rastreiam o uso de alguns recursos da AWS. Essas métricas correspondem à service quotas da AWS. O rastreamento dessas métricas pode ajudar a gerenciar as cotas proativamente. Para obter mais informações, consulte Visualizar as Service Quotas e definir alarmes.

As métricas de uso da cota de serviço estão no namespace `AWS/Usage` e são coletadas a cada minuto.

No momento, o único nome da métrica nesse namespace que o CloudWatch publica é `CallCount`. Essa métrica é publicada com as dimensões `Resource`, `Service` e `Type`. A dimensão `Resource` especifica o nome da operação da API que está sendo acompanhada. Por exemplo, a métrica `CallCount` com as dimensões `"Service": "CloudWatch"`, `"Type": "API"` e `"Resource": "PutMetricData"` indica o número de vezes que a operação da API do CloudWatch `PutMetricData` foi chamada em sua conta.

A métrica `CallCount` não tem uma unidade especificada. A estatística mais útil para a métrica é `SUM`, que representa a contagem total de operações para o período de 1 minuto.

Métricas

Métrica	Descrição
`CallCount`	O número de operações especificadas executadas em sua conta.

Dimensões

Dimensão	Descrição
`Service`	O nome do serviço da AWS que contém o recurso. Para as métricas de uso do CloudWatch, o valor dessa dimensão é `CloudWatch` .
`Class`	A classe do recurso que está sendo acompanhado. As métricas de uso da API do CloudWatch usam essa dimensão com um valor de `None`.
`Type`	O tipo de recurso que está sendo acompanhado. No momento, quando a dimensão `Service` é `CloudWatch` , o único valor válido para `Type` é `API`.
`Resource`	O nome da operação da API. Entre os valores válidos estão os seguintes : `DeleteAlarms` , `DeleteDashboards` , `DescribeAlarmHistory` , `DescribeAlarms` , `GetDashboard` , `GetMetricData` , `GetMetricStatistics` , `ListMetrics` , `PutDashboard` e `PutMetricData`

Tutoriais do CloudWatch

Os cenários a seguir ilustram usos do Amazon CloudWatch. No primeiro cenário, use o console do CloudWatch para criar um alarme de faturamento que monitora a utilização da AWS e informa quando você excedeu um determinado limite de gasto. No segundo cenário, mais avançado, você usa a AWS Command Line Interface (AWS CLI) para publicar uma única métrica de um aplicativo hipotético chamado GetStarted.

Cenários

- Monitorar suas estimativas de cobrança
- Publicar métricas

Cenário: monitorar estimativas de custos usando o CloudWatch

Neste cenário, você cria um alarme do Amazon CloudWatch para monitorar suas estimativas de gastos. Quando você habilita o monitoramento de estimativas de cobrança para sua conta da AWS, as estimativas de cobrança são calculadas e enviadas várias vezes por dia para o CloudWatch como dados de métrica.

Os dados de métrica de faturamento são armazenados na região Leste dos EUA (Norte da Virgínia) e refletem cobranças mundiais. Esses dados incluem as estimativas de cobrança para cada produto da AWS que você usar, bem como o total geral estimado de suas cobranças da AWS.

É possível optar por receber alertas por e-mail quando as cobranças excederem um determinado limite. Esses alertas são acionados pelo CloudWatch, e as mensagens são enviadas usando o Amazon Simple Notification Service (Amazon SNS).

> **Note**
>
> Para obter informações sobre como analisar as cobranças do CloudWatch nas quais você já incorreu, consulte Faturamento e custos do CloudWatch.

Tarefas

- Etapa 1: Habilitar alertas de faturamento

Etapa 1: Habilitar alertas de faturamento

Para criar um alarme para suas estimativas de despesas, habilite alertas de faturamento para poder monitorar suas estimativas de despesas da AWS e criar um alarme usando dados de métrica de faturamento. Depois de ativar alertas de faturamento, você não poderá desativar a coleta de dados, mas poderá excluir qualquer alarme de faturamento que tenha criado.

Depois que habilitar alertas de pagamento pela primeira vez, levará cerca de 15 minutos para que você possa visualizar dados de faturamento e definir alertas de pagamento.

Requisitos

- Você deverá estar conectado usando as credenciais de usuário raiz ou como um usuário que tenha recebido permissão para visualizar as informações de faturamento.
- Para contas de faturamento consolidado, os dados de faturamento para cada conta vinculada podem ser encontrados fazendo login como a conta de pagamento. Você pode visualizar dados de faturamento para o total de cobranças estimadas e cobranças estimadas por serviço para cada conta vinculada, além da conta consolidada.
- Em uma conta de faturamento consolidado, as métricas da conta vinculada ao membro serão capturadas somente se a conta pagante habilitar a preferência Receber alertas de faturamento. Se você alterar qual é a conta de gerenciamento/pagante, será necessário habilitar os alertas de faturamento na nova conta de gerenciamento/pagante.
- A conta não deve fazer parte da Rede de parceiros da Amazon (APN) porque as métricas de faturamento não são publicadas no CloudWatch para contas do APN. Para obter mais informações, consulte Rede de parceiros da AWS.

Para ativar o monitoramento de estimativas de gastos

1. Abra o console do AWS Billing em https://console.aws.amazon.com/billing/.
2. No painel de navegação, selecione Billing Preferences (Preferências de faturamento).
3. Em Preferências de alertas, escolha Editar.

4. Escolha Receber alertas de faturamento do CloudWatch.
5. Escolha Save preferences (Salvar preferências).

Etapa 2: Criar um alarme de faturamento

> **Important**
>
> Antes de criar um alarme de faturamento, defina a região como Leste dos EUA (Norte da Virgínia). Os dados de métrica de faturamento são armazenados nessa região e representam as despesas em todo o mundo. É necessário habilitar alertas de faturamento na conta ou na conta de gerenciamento/pagante (se você estiver usando faturamento consolidado). Para obter mais informações, consulte: Etapa 1: habilitar alertas de faturamento.

Neste procedimento, você cria um alarme que envia uma notificação quando suas estimativas de cobrança da AWS excedem um limite definido.

Para criar um alarme de cobrança usando o console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Alarms (Alarmes) e depois escolha All alarms (Todos os alarmes).
3. Selecione Create alarm (Criar alarme).
4. Escolha Select metric (Selecionar métrica). Em Browse (Navegar), escolha Billing (Faturamento) e escolha Total Estimated Charge (Total da cobrança estimada).

   > **Note**
   >
   > Caso não veja a métrica Faturamento/Total da cobrança estimada, habilite os alertas de faturamento e altere a região para Leste dos EUA (Norte da Virgínia). Para obter mais informações, consulte Habilitar alertas de faturamento.

5. Marque a caixa de seleção EstimatedCharges e escolha Select metric (Selecionar métrica).
6. Em Statistic (Estatística), escolha Maximum (Máximo).
7. Em Period (Período), escolha 6 hours (6 horas).
8. Em Threshold type (Tipo de limite), escolha Static (Estático).

9. Em Whenever EstimatedCharges is… (Sempre que EstimatedCharges for…), escolha Greater (Maior).
10. Em que . . ., defina o valor para o qual você deseja que seu alarme seja acionado. Por exemplo, **200** USD.

    Os valores da métrica EstimatedCharges estão somente em dólares americanos (USD), e a conversão da moeda é fornecida pela Amazon Services LLC. Para obter mais informações, consulte O que é o AWS Billing?.

    > **Note**
    >
    > Após definir um valor limite, o gráfico de pré-visualização exibirá suas cobranças estimadas do mês atual.

11. Em Configuração adicional, faça o seguinte:

    - Em Datapoints to alarm (Pontos de dados para alarme), especifique 1 of 1 (1 de 1).
    - Em Missing data treatment (Tratamento de dados ausentes), escolha Treat missing data as missing (Tratar dados ausentes como ausentes).

12. Escolha Next (Próximo).
13. Em Notificação, certifique-se de que a opção Em alarme esteja selecionada. Em seguida, especifique um tópico do Amazon SNS a ser notificado quando o alarme estiver no estado `ALARM`. O tópico do Amazon SNS pode incluir seu endereço de email para que você receba emails quando o valor do faturamento ultrapassar o limite especificado.

    É possível selecionar um tópico existente do Amazon SNS, criar um novo tópico do Amazon SNS ou usar um ARN do tópico para notificar outra conta. Se quiser que o alarme envie várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

14. Escolha Next (Próximo).
15. Em Name and description (Nome e descrição), insira um nome para o alarme.

    - (Opcional) Insira uma descrição do alarme.

16. Escolha Next (Próximo).
17. Em Preview and create (Pré-visualizar e criar), verifique se a configuração está correta e escolha Create alarm (Criar alarme).

Etapa 3: verificar o status do alarme

Agora, verifique o status do alarme de faturamento que você acabou de criar.

Para verificar o status do alarme

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Se necessário, altere a região para US East (N. Virginia) (Leste dos EUA (Norte da Virgínia)). Os dados da métrica de faturamento são armazenados nessa região e refletem os custos em todo o mundo.
3. No painel de navegação, selecione Alarmes.
4. Marque a caixa de seleção ao lado do alarme. Até a assinatura ser confirmada, ela aparece como "Confirmação pendente". Após a confirmação da assinatura, atualize o console para mostrar o status atualizado.

Etapa 4: editar um alarme de faturamento

Por exemplo, talvez você queira aumentar o valor gasto com a AWS mensalmente de USD 200 para USD 400. É possível editar seu alarme de faturamento existente e aumentar a quantia monetária que deve ser excedida para que o alarme seja acionado.

Para editar um alarme de faturamento

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Se necessário, altere a região para US East (N. Virginia) (Leste dos EUA (Norte da Virgínia)). Os dados da métrica de faturamento são armazenados nessa região e refletem os custos em todo o mundo.
3. No painel de navegação, selecione Alarmes.
4. Marque a caixa de seleção ao lado do alarme e escolha Actions (Ações), Modify (Modificar).
5. Em Sempre que meu total de cobranças da AWS no mês exceder a, especifique o novo valor que deverá ser excedido para acionar o alarme e enviar uma notificação por email.
6. Escolha Save Changes (Salvar alterações).

Etapa 5: excluir um alarme de faturamento

Exclua o alarme de faturamento caso não precise mais dele.

Para excluir um alarme de faturamento

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Se necessário, altere a região para US East (N. Virginia) (Leste dos EUA (Norte da Virgínia)). Os dados da métrica de faturamento são armazenados nessa região e refletem os custos em todo o mundo.
3. No painel de navegação, selecione Alarmes.
4. Marque a caixa de seleção ao lado do alarme e escolha Actions (Ações), Delete (Excluir).
5. Quando a confirmação for solicitada, escolha Yes, Delete (Sim, excluir).

Cenário: publicar métricas no CloudWatch

Neste cenário, você usa a AWS Command Line Interface (AWS CLI) para publicar uma única métrica de um aplicativo hipotético chamado GetStarted. Se você ainda não tiver instalado e configurado a AWS CLI, consulte Configurar a AWS Command Line Interface no Manual do usuário da AWS Command Line Interface.

Tarefas

- Etapa 1: definir a configuração dos dados
- Etapa 2: adicionar métricas ao CloudWatch
- Etapa 3: obter estatísticas do CloudWatch
- Etapa 4: visualizar gráficos com o console

Etapa 1: definir a configuração dos dados

Neste cenário, você publica pontos de dados que rastreiam a latência de solicitação do aplicativo. Escolha nomes para sua métrica e namespace que façam sentido para você. Para este exemplo, dê à métrica o nome RequestLatency e coloque todos os pontos de dados no namespace GetStarted.

Você publica diversos pontos de dados que, coletivamente, representam três horas de dados de latência. Os dados brutos compreendem 15 leituras de latência de solicitação distribuídas ao longo de três horas. Cada leitura é feita em milissegundos:

- Primeira hora: 87, 51, 125, 235
- Segunda hora: 121, 113, 189, 65, 89

- Terceira hora: 100, 47, 133, 98, 100, 328

Você pode publicar dados no CloudWatch como pontos de dados únicos ou como um conjunto agregado de pontos de dados denominado conjunto de estatísticas. Você pode agregar métricas para uma granularidade de até um minuto. É possível publicar os pontos de dados agregados no CloudWatch como um conjunto de estatísticas com quatro chaves predefinidas: `Sum` `Minimum`, `Maximum` e `SampleCount`.

Você publica os pontos de dados da primeira hora como pontos de dados únicos. Para os dados da segunda e terceira horas, você agrega os pontos de dados e publica um conjunto de estatísticas para cada hora. Os principais valores são mostrados na tabela a seguir.

Hora	Dados brutos	Sum	Mínimo	Máximo	SampleCount
1	87				
1	51				
1	125				
1	235				
2	121, 113, 189, 65, 89	577	65	189	5
3	100, 47, 133, 98, 100, 328	806	47	328	6

Etapa 2: adicionar métricas ao CloudWatch

Depois de definir a configuração de dados, você estará pronto para adicionar dados.

Para publicar pontos de dados no CloudWatch

1. Em um prompt de comando, execute os seguintes comandos put-metric-data para adicionar dados para a primeira hora. Substitua o time stamp de exemplo por um time stamp que seja duas horas no passado, em Universal Coordinated Time (UTC – Tempo universal coordenado).

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 87 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 51 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 125 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 235 --unit Milliseconds
```

2. Adicione dados para a segunda hora usando um time stamp que seja uma hora posterior à primeira hora.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T21:30:00Z --statistic-values
 Sum=577,Minimum=65,Maximum=189,SampleCount=5 --unit Milliseconds
```

3. Adicione dados para a terceira hora omitindo o time stamp para o padrão à hora atual.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--statistic-values Sum=806,Minimum=47,Maximum=328,SampleCount=6 --unit Milliseconds
```

Etapa 3: obter estatísticas do CloudWatch

Agora que você publicou métricas no CloudWatch, pode recuperar as estatísticas de acordo com as métricas usando o comando get-metric-statistics, como se segue. Especifique `--start-time` e `--end-time` o suficiente no passado para cobrir o primeiro time stamp publicado.

```
aws cloudwatch get-metric-statistics --namespace GetStarted --metric-name
 RequestLatency --statistics Average \
--start-time 2016-10-14T00:00:00Z --end-time 2016-10-15T00:00:00Z --period 60
```

A seguir está um exemplo de saída:

```
{
 "Datapoints": [],
 "Label": "Request:Latency"
}
```

Etapa 4: visualizar gráficos com o console

Depois de publicar métricas no CloudWatch, você pode usar o console do CloudWatch para visualizar gráficos estatísticos.

Para visualizar gráficos de estatísticas no console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. Na guia All metrics (Todas as métricas), na caixa de pesquisa, digite RequestLatency e pressione Enter.
4. Marque a caixa de seleção da métrica RequestLatency. Um gráfico dos dados de métrica é exibido no painel superior.

Para obter mais informações, consulte Criar gráficos de métricas.

Como usar o CloudWatch com um AWS SDK

Os kits de desenvolvimento de software (SDKs) da AWS estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que facilitam a criação de aplicações em seu idioma preferido pelos desenvolvedores.

Documentação do SDK	Exemplos de código
AWS SDK for C++	Exemplos de código do AWS SDK for C++
AWS CLI	Exemplos de código do AWS CLI
AWS SDK for Go	Exemplos de código do AWS SDK for Go
AWS SDK for Java	Exemplos de código do AWS SDK for Java
AWS SDK for JavaScript	Exemplos de código do AWS SDK for JavaScript
AWS SDK para Kotlin	Exemplos de código do AWS SDK para Kotlin
AWS SDK for .NET	Exemplos de código do AWS SDK for .NET
AWS SDK for PHP	Exemplos de código do AWS SDK for PHP
AWS Tools for PowerShell	Tools for PowerShell code examples
AWS SDK for Python (Boto3)	Exemplos de código do AWS SDK for Python (Boto3)
AWS SDK for Ruby	Exemplos de código do AWS SDK for Ruby
AWS SDK para Rust	Exemplos de código do AWS SDK para Rust
SDK da AWS para SAP ABAP	Exemplos de código do SDK da AWS para SAP ABAP
AWS SDK for Swift	Exemplos de código do AWS SDK for Swift

Para obter exemplos específicos do CloudWatch, consulte Exemplos de código para o CloudWatch usando AWS SDKs.

Exemplo de disponibilidade

Você não consegue encontrar o que precisa? Solicite um código de exemplo no link Fornecer feedback na parte inferior desta página.

Exemplos de código para o CloudWatch usando AWS SDKs

Os exemplos de código a seguir mostram como usar o CloudWatch com um kit de desenvolvimento de software (SDK) da AWS.

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço específicas, é possível ver as ações contextualizadas em seus devidos cenários e exemplos entre serviços.

Cenários são exemplos de código que mostram como realizar uma tarefa específica chamando várias funções dentro do mesmo serviço.

Exemplos entre serviços são amostras de aplicações que funcionam em vários Serviços da AWS.

Para obter uma lista completa dos Guias do desenvolvedor do SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Conceitos básicos

Olá, CloudWatch

O exemplo de código a seguir mostra como começar a usar o CloudWatch.

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
using Amazon.CloudWatch;
using Amazon.CloudWatch.Model;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
```

```
namespace CloudWatchActions;

public static class HelloCloudWatch
{
    static async Task Main(string[] args)
    {
        // Use the AWS .NET Core Setup package to set up dependency injection for
 the Amazon CloudWatch service.
        // Use your AWS profile name, or leave it blank to use the default
 profile.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonCloudWatch>()
            ).Build();

        // Now the client is available for injection.
        var cloudWatchClient =
 host.Services.GetRequiredService<IAmazonCloudWatch>();

        // You can use await and any of the async methods to get a response.
        var metricNamespace = "AWS/Billing";
        var response = await cloudWatchClient.ListMetricsAsync(new
 ListMetricsRequest
        {
            Namespace = metricNamespace
        });
        Console.WriteLine($"Hello Amazon CloudWatch! Following are some metrics
 available in the {metricNamespace} namespace:");
        Console.WriteLine();
        foreach (var metric in response.Metrics.Take(5))
        {
            Console.WriteLine($"\tMetric: {metric.MetricName}");
            Console.WriteLine($"\tNamespace: {metric.Namespace}");
            Console.WriteLine($"\tDimensions: {string.Join(", ",
 metric.Dimensions.Select(m => $"{m.Name}:{m.Value}"))}");
            Console.WriteLine();
        }
    }
}
```

- Para obter detalhes da API, consulte ListMetrics na Referência da API AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.paginators.ListMetricsIterable;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloService {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <namespace>\s

                Where:
                  namespace - The namespace to filter against (for example, AWS/
EC2).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String namespace = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        listMets(cw, namespace);
        cw.close();
    }

    public static void listMets(CloudWatchClient cw, String namespace) {
        try {
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> System.out.println(" Retrieved metric is:
 " + metrics.metricName()));

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte ListMetrics na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
 */
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
           <namespace>
        Where:
           namespace - The namespace to filter against (for example, AWS/EC2).
    """

    if (args.size != 1) {
        println(usage)
        exitProcess(0)
    }

    val namespace = args[0]
    listAllMets(namespace)
}

suspend fun listAllMets(namespaceVal: String?) {
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listMetricsPaginated(request)
            .transform { it.metrics?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.metricName}")
                println("Namespace is ${obj.namespace}")
            }
    }
}
```

- Para obter detalhes da API, consulte ListMetrics na Referência da API AWS SDK para Kotlin.

Exemplos de código

- Ações para o CloudWatch usando AWS SDKs
  - Usar DeleteAlarms com o AWS SDK ou a CLI
  - Usar DeleteAnomalyDetector com o AWS SDK ou a CLI
  - Usar DeleteDashboards com o AWS SDK ou a CLI
  - Usar DescribeAlarmHistory com o AWS SDK ou a CLI
  - Usar DescribeAlarms com o AWS SDK ou a CLI
  - Usar DescribeAlarmsForMetric com o AWS SDK ou a CLI
  - Usar DescribeAnomalyDetectors com o AWS SDK ou a CLI
  - Usar DisableAlarmActions com o AWS SDK ou a CLI
  - Usar EnableAlarmActions com o AWS SDK ou a CLI
  - Usar GetDashboard com o AWS SDK ou a CLI
  - Usar GetMetricData com o AWS SDK ou a CLI
  - Usar GetMetricStatistics com o AWS SDK ou a CLI
  - Usar GetMetricWidgetImage com o AWS SDK ou a CLI
  - Usar ListDashboards com o AWS SDK ou a CLI
  - Usar ListMetrics com o AWS SDK ou a CLI
  - Usar PutAnomalyDetector com o AWS SDK ou a CLI
  - Usar PutDashboard com o AWS SDK ou a CLI
  - Usar PutMetricAlarm com o AWS SDK ou a CLI
  - Usar PutMetricData com o AWS SDK ou a CLI
- Cenários para o CloudWatch usando AWS SDKs
  - Começar a usar alarmes do CloudWatch usando um AWS SDK
  - Começar a usar métricas, painéis e alarmes do CloudWatch usando um AWS SDK
  - Gerenciar métricas e alarmes do CloudWatch usando um AWS SDK
- Exemplos de serviços cruzados para o CloudWatch Logs usando AWS SDKs
  - Monitoramento do desempenho do Amazon DynamoDB usando um AWS SDK

Ações para o CloudWatch usando AWS SDKs

Os exemplos de código a seguir demonstram como realizar ações individuais do CloudWatch com SDKs da AWS. Esses trechos chamam a API do CloudWatch Logs e são trechos de código de programas maiores que devem ser executados no contexto. Cada exemplo inclui um link para o GitHub, em que é possível encontrar instruções para configurar e executar o código.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a Referência de APIs do Amazon CloudWatch.

Exemplos

- Usar DeleteAlarms com o AWS SDK ou a CLI
- Usar DeleteAnomalyDetector com o AWS SDK ou a CLI
- Usar DeleteDashboards com o AWS SDK ou a CLI
- Usar DescribeAlarmHistory com o AWS SDK ou a CLI
- Usar DescribeAlarms com o AWS SDK ou a CLI
- Usar DescribeAlarmsForMetric com o AWS SDK ou a CLI
- Usar DescribeAnomalyDetectors com o AWS SDK ou a CLI
- Usar DisableAlarmActions com o AWS SDK ou a CLI
- Usar EnableAlarmActions com o AWS SDK ou a CLI
- Usar GetDashboard com o AWS SDK ou a CLI
- Usar GetMetricData com o AWS SDK ou a CLI
- Usar GetMetricStatistics com o AWS SDK ou a CLI
- Usar GetMetricWidgetImage com o AWS SDK ou a CLI
- Usar ListDashboards com o AWS SDK ou a CLI
- Usar ListMetrics com o AWS SDK ou a CLI
- Usar PutAnomalyDetector com o AWS SDK ou a CLI
- Usar PutDashboard com o AWS SDK ou a CLI
- Usar PutMetricAlarm com o AWS SDK ou a CLI
- Usar PutMetricData com o AWS SDK ou a CLI

Usar **`DeleteAlarms`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DeleteAlarms`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- Começar a usar alarmes
- Começar a usar métricas, painéis e alarmes
- Gerencie métricas e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
/// <summary>
/// Delete a list of alarms from CloudWatch.
/// </summary>
/// <param name="alarmNames">A list of names of alarms to delete.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteAlarms(List<string> alarmNames)
{
    var deleteAlarmsResult = await _amazonCloudWatch.DeleteAlarmsAsync(
        new DeleteAlarmsRequest()
        {
            AlarmNames = alarmNames
        });

    return deleteAlarmsResult.HttpStatusCode == HttpStatusCode.OK;
}
```

- Para obter detalhes da API, consulte DeleteAlarms na Referência da API AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Inclua os arquivos necessários.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DeleteAlarmsRequest.h>
#include <iostream>
```

Excluir o alarme.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::DeleteAlarmsRequest request;
        request.AddAlarmNames(alarm_name);

        auto outcome = cw.DeleteAlarms(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to delete CloudWatch alarm:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully deleted CloudWatch alarm " << alarm_name
                << std::endl;
        }
```

- Para obter detalhes da API, consulte DeleteAlarms na Referência da API AWS SDK for C++.

CLI

AWS CLI

Como excluir um alarme

O seguinte exemplo usa o comando `delete-alarms` para excluir o alarme “myalarm” do Amazon CloudWatch:

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

Saída:

```
This command returns to the prompt if successful.
```

- Para obter detalhes da API, consulte DeleteAlarms na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.DeleteAlarmsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
```

```
 */

public class DeleteAlarm {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to delete (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarmName = args[0];
        Region region = Region.US_EAST_2;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        deleteCWAlarm(cw, alarmName);
        cw.close();
    }

    public static void deleteCWAlarm(CloudWatchClient cw, String alarmName) {
        try {
            DeleteAlarmsRequest request = DeleteAlarmsRequest.builder()
                    .alarmNames(alarmName)
                    .build();

            cw.deleteAlarms(request);
            System.out.printf("Successfully deleted alarm %s", alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte DeleteAlarms na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Importe o SDK e os módulos do cliente e chame a API.

```
import { DeleteAlarmsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteAlarmsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Crie o cliente em um módulo separado e exporte-o.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";
```

```
export const client = new CloudWatchClient({});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte DeleteAlarms na Referência da API AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Importe o SDK e os módulos do cliente e chame a API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmNames: ["Web_Server_CPU_Utilization"],
};

cw.deleteAlarms(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.

- Para obter detalhes da API, consulte DeleteAlarms na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
suspend fun deleteAlarm(alarmNameVal: String) {
    val request = DeleteAlarmsRequest {
        alarmNames = listOf(alarmNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAlarms(request)
        println("Successfully deleted alarm $alarmNameVal")
    }
}
```

- Para obter detalhes da API, consulte DeleteAlarms na Referência da API AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class CloudWatchWrapper:
```

```
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def delete_metric_alarms(self, metric_namespace, metric_name):
        """
        Deletes all of the alarms that are currently watching the specified
metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            metric.alarms.delete()
            logger.info(
                "Deleted alarms for metric %s.%s.", metric_namespace, metric_name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete alarms for metric %s.%s.",
                metric_namespace,
                metric_name,
            )
            raise
```

- Para obter detalhes da API, consulte DeleteAlarms na Referência da API AWS SDK para Python (Boto3).

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
TRY.
    lo_cwt->deletealarms(
      it_alarmnames = it_alarm_names
    ).
    MESSAGE 'Alarms deleted.' TYPE 'I'.
  CATCH /aws1/cx_cwtresourcenotfound .
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes sobre a API, consulte DeleteAlarms na Referência da API do AWS para SAP ABAP.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteAnomalyDetector`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DeleteAnomalyDetector`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar métricas, painéis e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Delete a single metric anomaly detector.
    /// </summary>
    /// <param name="anomalyDetector">The anomaly detector to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
    {
        var deleteAnomalyDetectorResponse = await
_amazonCloudWatch.DeleteAnomalyDetectorAsync(
            new DeleteAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return deleteAnomalyDetectorResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obter detalhes da API, consulte DeleteAnomalyDetector, na Referência da API AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   public static void deleteAnomalyDetector(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .stat("Maximum")
                   .build();

           DeleteAnomalyDetectorRequest request =
DeleteAnomalyDetectorRequest.builder()
                   .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                   .build();

           cw.deleteAnomalyDetector(request);
           System.out.println("Successfully deleted the Anomaly Detector.");

       } catch (CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       } catch (IOException e) {
           e.printStackTrace();
       }
   }
```

- Para obter detalhes da API, consulte DeleteAnomalyDetector, na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun deleteAnomalyDetector(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val request = DeleteAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAnomalyDetector(request)
        println("Successfully deleted the Anomaly Detector.")
    }
}
```

- Para obter detalhes da API, consulte DeleteAnomalyDetector na Referência da API do AWS SDK para Kotlin.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DeleteDashboards`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DeleteDashboards`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar métricas, painéis e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Delete a list of CloudWatch dashboards.
    /// </summary>
    /// <param name="dashboardNames">List of dashboard names to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDashboards(List<string> dashboardNames)
    {
        var deleteDashboardsResponse = await
_amazonCloudWatch.DeleteDashboardsAsync(
            new DeleteDashboardsRequest()
            {
                DashboardNames = dashboardNames
            });

        return deleteDashboardsResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obter detalhes da API, consulte DeleteDashboards, na Referência da API AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   public static void deleteDashboard(CloudWatchClient cw, String dashboardName)
{
       try {
           DeleteDashboardsRequest dashboardsRequest =
DeleteDashboardsRequest.builder()
                   .dashboardNames(dashboardName)
                   .build();
           cw.deleteDashboards(dashboardsRequest);
           System.out.println(dashboardName + " was successfully deleted.");

       } catch (CloudWatchException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }
```

- Para obter detalhes da API, consulte DeleteDashboards, na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun deleteDashboard(dashboardName: String) {
    val dashboardsRequest = DeleteDashboardsRequest {
        dashboardNames = listOf(dashboardName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteDashboards(dashboardsRequest)
        println("$dashboardName was successfully deleted.")
    }
}
```

- Para obter detalhes da API, consulte DeleteDashboards, na Referência da API AWS SDK para Kotlin.

PowerShell

Tools for PowerShell

Exemplo 1: exclui o painel especificado, solicitando uma confirmação antes de continuar. Para ignorar a confirmação, adicione a opção -Force para o comando.

```
Remove-CWDashboard -DashboardName Dashboard1
```

- Para obter detalhes da API, consulte DeleteDashboards na Referência de cmdlet do AWS Tools for PowerShell.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAlarmHistory`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeAlarmHistory`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar métricas, painéis e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Describe the history of an alarm for a number of days in the past.
    /// </summary>
    /// <param name="alarmName">The name of the alarm.</param>
    /// <param name="historyDays">The number of days in the past.</param>
    /// <returns>The list of alarm history data.</returns>
    public async Task<List<AlarmHistoryItem>> DescribeAlarmHistory(string
alarmName, int historyDays)
    {
        List<AlarmHistoryItem> alarmHistory = new List<AlarmHistoryItem>();
        var paginatedAlarmHistory =
_amazonCloudWatch.Paginators.DescribeAlarmHistory(
            new DescribeAlarmHistoryRequest()
            {
                AlarmName = alarmName,
                EndDateUtc = DateTime.UtcNow,
                HistoryItemType = HistoryItemType.StateUpdate,
                StartDateUtc = DateTime.UtcNow.AddDays(-historyDays)
            });

        await foreach (var data in paginatedAlarmHistory.AlarmHistoryItems)
        {
            alarmHistory.Add(data);
        }
        return alarmHistory;
    }
```

- Para obter detalhes, consulte o DescribeAlarmHistory, na Referência da API AWS SDK for .NET.

CLI

AWS CLI

Como recuperar o histórico de um alarme

O seguinte exemplo usa o comando `describe-alarm-history` para recuperar o histórico do alarme “myalarm” do Amazon CloudWatch:

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-type
 StateUpdate
```

Saída:

```
{
    "AlarmHistoryItems": [
        {
            "Timestamp": "2014-04-09T18:59:06.442Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"ALARM\",\"stateReason\":\"testing purposes\"},\"newState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were not
 greater than the threshold (70.0). The most recent datapoints: [38.958,
 40.292].\",\"stateReasonData\":{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}}}",
            "HistorySummary": "Alarm updated from ALARM to OK"
        },
        {
            "Timestamp": "2014-04-09T18:59:05.805Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were
 not greater than the threshold (70.0). The most recent datapoints:
 [38.839999999999996, 39.714].\",\"stateReasonData\":{\"version\":
\"1.0\",\"queryDate\":\"2014-03-11T22:45:41.569+0000\",\"startDate\":
\"2014-03-11T22:30:00.000+0000\",\"statistic\":\"Average\",\"period\":300,
\"recentDatapoints\":[38.839999999999996,39.714],\"threshold\":70.0}},\"newState
\":{\"stateValue\":\"ALARM\",\"stateReason\":\"testing purposes\"}}",
            "HistorySummary": "Alarm updated from OK to ALARM"
```

```
        }
    ]
}
```

- Para obter detalhes da API, consulte DescribeAlarmHistory na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   public static void getAlarmHistory(CloudWatchClient cw, String fileName,
String date) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String alarmName = rootNode.findValue("exampleAlarmName").asText();

           Instant start = Instant.parse(date);
           Instant endDate = Instant.now();
           DescribeAlarmHistoryRequest historyRequest =
DescribeAlarmHistoryRequest.builder()
                   .startDate(start)
                   .endDate(endDate)
                   .alarmName(alarmName)
                   .historyItemType(HistoryItemType.ACTION)
                   .build();

           DescribeAlarmHistoryResponse response =
cw.describeAlarmHistory(historyRequest);
           List<AlarmHistoryItem> historyItems = response.alarmHistoryItems();
           if (historyItems.isEmpty()) {
```

```
                System.out.println("No alarm history data found for " + alarmName
 + ".");
            } else {
                for (AlarmHistoryItem item : historyItems) {
                    System.out.println("History summary: " +
 item.historySummary());
                    System.out.println("Time stamp: " + item.timestamp());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte o DescribeAlarmHistory, na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun getAlarmHistory(fileName: String, date: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val start = Instant.parse(date)
    val endDateVal = Instant.now()

    val historyRequest = DescribeAlarmHistoryRequest {
        startDate = aws.smithy.kotlin.runtime.time.Instant(start)
        endDate = aws.smithy.kotlin.runtime.time.Instant(endDateVal)
```

```
        alarmName = alarmNameVal
        historyItemType = HistoryItemType.Action
    }

    CloudWatchClient { credentialsProvider = EnvironmentCredentialsProvider();
 region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarmHistory(historyRequest)
        val historyItems = response.alarmHistoryItems
        if (historyItems != null) {
            if (historyItems.isEmpty()) {
                println("No alarm history data found for $alarmNameVal.")
            } else {
                for (item in historyItems) {
                    println("History summary ${item.historySummary}")
                    println("Time stamp: ${item.timestamp}")
                }
            }
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeAlarmHistory, na Referência de APIs do AWS SDK para Kotlin.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAlarms`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeAlarms`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- Começar a usar alarmes
- Começar a usar métricas, painéis e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Describe the current alarms, optionally filtered by state.
    /// </summary>
    /// <param name="stateValue">Optional filter for alarm state.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarms(StateValue? stateValue =
null)
    {
        List<MetricAlarm> alarms = new List<MetricAlarm>();
        var paginatedDescribeAlarms =
_amazonCloudWatch.Paginators.DescribeAlarms(
            new DescribeAlarmsRequest()
            {
                StateValue = stateValue
            });

        await foreach (var data in paginatedDescribeAlarms.MetricAlarms)
        {
            alarms.Add(data);
        }
        return alarms;
    }
```

- Para obter detalhes da API, consulte DescribeAlarms, na Referência da API AWS SDK for .NET.

CLI

AWS CLI

Como listar informações sobre um alarme

O seguinte exemplo usa o comando `describe-alarms` para fornecer informações sobre o alarme chamado “myalarm”:

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

Saída:

```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:myalarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2012-12-27T00:49:54.032Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:123456789012:myHighCpuAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myalarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
```

```
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Para obter detalhes da API, consulte DescribeAlarms na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    public static void describeAlarms(CloudWatchClient cw) {
        try {
            List<AlarmType> typeList = new ArrayList<>();
            typeList.add(AlarmType.METRIC_ALARM);

            DescribeAlarmsRequest alarmsRequest = DescribeAlarmsRequest.builder()
                    .alarmTypes(typeList)
                    .maxRecords(10)
                    .build();

            DescribeAlarmsResponse response = cw.describeAlarms(alarmsRequest);
            List<MetricAlarm> alarmList = response.metricAlarms();
            for (MetricAlarm alarm : alarmList) {
                System.out.println("Alarm name: " + alarm.alarmName());
                System.out.println("Alarm description: " +
 alarm.alarmDescription());
            }
        } catch (CloudWatchException e) {
```

```
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte DescribeAlarms, na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun describeAlarms() {
    val typeList = ArrayList<AlarmType>()
    typeList.add(AlarmType.MetricAlarm)
    val alarmsRequest = DescribeAlarmsRequest {
        alarmTypes = typeList
        maxRecords = 10
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarms(alarmsRequest)
        response.metricAlarms?.forEach { alarm ->
            println("Alarm name: ${alarm.alarmName}")
            println("Alarm description: ${alarm.alarmDescription}")
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeAlarms, na Referência da API AWS SDK para Kotlin.

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require "aws-sdk-cloudwatch"

# Lists the names of available Amazon CloudWatch alarms.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   list_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def list_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms
  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts alarm.alarm_name
    end
  else
    puts "No alarms found."
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end
```

- Para obter detalhes da API, consulte DescribeAlarms, na Referência da API AWS SDK for Ruby.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    TRY.
        oo_result = lo_cwt->describealarms(                " oo_result is
 returned for testing purposes. "
          it_alarmnames = it_alarm_names
        ).
        MESSAGE 'Alarms retrieved.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obter detalhes sobre a API, consulte DescribeAlarms na Referência de APIs do AWS SDK para SAP ABAP.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAlarmsForMetric`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeAlarmsForMetric`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- Começar a usar métricas, painéis e alarmes
- Gerencie métricas e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Describe the current alarms for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarmsForMetric(string
metricNamespace, string metricName)
    {
        var alarmsResult = await _amazonCloudWatch.DescribeAlarmsForMetricAsync(
            new DescribeAlarmsForMetricRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName
            });

        return alarmsResult.MetricAlarms;
    }
```

- Para obter detalhes da API, consulte DescribeAlarmsForMetric na Referência da API AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Inclua os arquivos necessários.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DescribeAlarmsRequest.h>
#include <aws/monitoring/model/DescribeAlarmsResult.h>
#include <iomanip>
#include <iostream>
```

Descreva os alarmes.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::DescribeAlarmsRequest request;
        request.SetMaxRecords(1);

        bool done = false;
        bool header = false;
        while (!done)
        {
            auto outcome = cw.DescribeAlarms(request);
            if (!outcome.IsSuccess())
            {
                std::cout << "Failed to describe CloudWatch alarms:" <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header)
            {
                std::cout << std::left <<
                    std::setw(32) << "Name" <<
```

```
                    std::setw(64) << "Arn" <<
                    std::setw(64) << "Description" <<
                    std::setw(20) << "LastUpdated" <<
                    std::endl;
                header = true;
            }

            const auto &alarms = outcome.GetResult().GetMetricAlarms();
            for (const auto &alarm : alarms)
            {
                std::cout << std::left <<
                    std::setw(32) << alarm.GetAlarmName() <<
                    std::setw(64) << alarm.GetAlarmArn() <<
                    std::setw(64) << alarm.GetAlarmDescription() <<
                    std::setw(20) <<
                    alarm.GetAlarmConfigurationUpdatedTimestamp().ToGmtString(
                        SIMPLE_DATE_FORMAT_STR) <<
                    std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Para obter detalhes da API, consulte DescribeAlarmsForMetric na Referência da API AWS SDK for C++.

CLI

AWS CLI

Como exibir informações sobre os alarmes associados a uma métrica

O seguinte exemplo usa o comando `describe-alarms-for-metric` para exibir informações sobre qualquer alarme associado à métrica CPUUtilization do Amazon EC2 e à instância com o ID i-0c986c72.:

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --
namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

Saída:


```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 10,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm2",
            "StateUpdatedTimestamp": "2013-10-30T03:03:51.479Z",
            "AlarmConfigurationUpdatedTimestamp": "2013-10-30T03:03:50.865Z",
            "ComparisonOperator": "GreaterThanOrEqualToThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:NotifyMe"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2013-10-30T03:03:51.479+0000\",\"startDate\":\"2013-10-30T02:08:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":
[40.698,39.612,42.432,39.796,38.816,42.28,42.854,40.088,40.760000000000005,41.316],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 10 datapoints were not
 greater than or equal to the threshold (70.0). The most recent datapoints:
 [40.760000000000005, 41.316].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        },
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm",
```

```
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2014-04-09T22:26:05.958Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:HighCPUAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": false,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Para obter detalhes da API, consulte DescribeAlarmsForMetric na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
   public static void checkForMetricAlarm(CloudWatchClient cw, String fileName)
{
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           boolean hasAlarm = false;
           int retries = 10;

           DescribeAlarmsForMetricRequest metricRequest =
DescribeAlarmsForMetricRequest.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           while (!hasAlarm && retries > 0) {
               DescribeAlarmsForMetricResponse response =
cw.describeAlarmsForMetric(metricRequest);
               hasAlarm = response.hasMetricAlarms();
               retries--;
               Thread.sleep(20000);
               System.out.println(".");
           }
           if (!hasAlarm)
               System.out.println("No Alarm state found for " + customMetricName
+ " after 10 retries.");
```

```
            else
                System.out.println("Alarm state found for " + customMetricName +
 ".");

        } catch (CloudWatchException | IOException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte DescribeAlarmsForMetric na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Importe o SDK e os módulos do cliente e chame a API.

```
import { DescribeAlarmsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DescribeAlarmsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};
```

```
export default run();
```

Crie o cliente em um módulo separado e exporte-o.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte DescribeAlarmsForMetric na Referência da API AWS SDK for JavaScript .

SDK para JavaScript (v2)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

cw.describeAlarms({ StateValue: "INSUFFICIENT_DATA" }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    // List the names of all current alarms in the console
    data.MetricAlarms.forEach(function (item, index, array) {
      console.log(item.AlarmName);
    });
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte DescribeAlarmsForMetric na Referência da API AWS SDK for JavaScript .

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
suspend fun checkForMetricAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    var hasAlarm = false
    var retries = 10

    val metricRequest = DescribeAlarmsForMetricRequest {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        while (!hasAlarm && retries > 0) {
            val response = cwClient.describeAlarmsForMetric(metricRequest)
            if (response.metricAlarms?.count()!! > 0) {
                hasAlarm = true
            }
            retries--
            delay(20000)
            println(".")
        }
```

```
        if (!hasAlarm) println("No Alarm state found for $customMetricName after
 10 retries.") else println("Alarm state found for $customMetricName.")
    }
}
```

- Para obter mais detalhes da API, consulte DescribeAlarmsForMetric em Referência da API AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def get_metric_alarms(self, metric_namespace, metric_name):
        """
        Gets the alarms that are currently watching the specified metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :returns: An iterator that yields the alarms.
        """
        metric = self.cloudwatch_resource.Metric(metric_namespace, metric_name)
        alarm_iter = metric.alarms.all()
        logger.info("Got alarms for metric %s.%s.", metric_namespace,
 metric_name)
```

```
        return alarm_iter
```

- Para obter mais detalhes da API, consulte DescribeAlarmsForMetric em Referência da API AWS SDK para Python (Boto3) .

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   describe_metric_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def describe_metric_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms

  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts "-" * 16
      puts "Name:           " + alarm.alarm_name
      puts "State value:    " + alarm.state_value
      puts "State reason:   " + alarm.state_reason
      puts "Metric:         " + alarm.metric_name
      puts "Namespace:      " + alarm.namespace
      puts "Statistic:      " + alarm.statistic
      puts "Period:         " + alarm.period.to_s
      puts "Unit:           " + alarm.unit.to_s
      puts "Eval. periods:  " + alarm.evaluation_periods.to_s
      puts "Threshold:      " + alarm.threshold.to_s
      puts "Comp. operator: " + alarm.comparison_operator
```

```
      if alarm.key?(:ok_actions) && alarm.ok_actions.count.positive?
        puts "OK actions:"
        alarm.ok_actions.each do |a|
          puts "  " + a
        end
      end

      if alarm.key?(:alarm_actions) && alarm.alarm_actions.count.positive?
        puts "Alarm actions:"
        alarm.alarm_actions.each do |a|
          puts "  " + a
        end
      end

      if alarm.key?(:insufficient_data_actions) &&
          alarm.insufficient_data_actions.count.positive?
        puts "Insufficient data actions:"
        alarm.insufficient_data_actions.each do |a|
          puts "  " + a
        end
      end

      puts "Dimensions:"
      if alarm.key?(:dimensions) && alarm.dimensions.count.positive?
        alarm.dimensions.each do |d|
          puts "  Name: " + d.name + ", Value: " + d.value
        end
      else
        puts "  None for this alarm."
      end
    end
  else
    puts "No alarms found."
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end

# Example usage:
def run_me
  region = ""

  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
```

```
    puts "Usage:   ruby cw-ruby-example-show-alarms.rb REGION"
    puts "Example: ruby cw-ruby-example-show-alarms.rb us-east-1"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = "us-east-1"
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
  puts "Available alarms:"
  describe_metric_alarms(cloudwatch_client)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte DescribeAlarmsForMetric na Referência da API AWS SDK for Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DescribeAnomalyDetectors`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DescribeAnomalyDetectors`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar métricas, painéis e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Describe anomaly detectors for a metric and namespace.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The metric of the anomaly detectors.</param>
    /// <returns>The list of detectors.</returns>
    public async Task<List<AnomalyDetector>> DescribeAnomalyDetectors(string
metricNamespace, string metricName)
    {
        List<AnomalyDetector> detectors = new List<AnomalyDetector>();
        var paginatedDescribeAnomalyDetectors =
_amazonCloudWatch.Paginators.DescribeAnomalyDetectors(
            new DescribeAnomalyDetectorsRequest()
            {
                MetricName = metricName,
                Namespace = metricNamespace
            });

        await foreach (var data in
paginatedDescribeAnomalyDetectors.AnomalyDetectors)
        {
            detectors.Add(data);
        }

        return detectors;
    }
```

- Para obter detalhes da API, consulte DescribeAnomalyDetectors, na Referência da API AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   public static void describeAnomalyDetectors(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           DescribeAnomalyDetectorsRequest detectorsRequest =
DescribeAnomalyDetectorsRequest.builder()
                   .maxResults(10)
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           DescribeAnomalyDetectorsResponse response =
cw.describeAnomalyDetectors(detectorsRequest);
           List<AnomalyDetector> anomalyDetectorList =
response.anomalyDetectors();
           for (AnomalyDetector detector : anomalyDetectorList) {
               System.out.println("Metric name: " +
detector.singleMetricAnomalyDetector().metricName());
               System.out.println("State: " + detector.stateValue());
           }

       } catch (CloudWatchException | IOException e) {
           System.err.println(e.getMessage());
           System.exit(1);
```

```
        }
    }
```

- Para obter detalhes da API, consulte DescribeAnomalyDetectors, na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun describeAnomalyDetectors(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val detectorsRequest = DescribeAnomalyDetectorsRequest {
        maxResults = 10
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAnomalyDetectors(detectorsRequest)
        response.anomalyDetectors?.forEach { detector ->
            println("Metric name:
 ${detector.singleMetricAnomalyDetector?.metricName}")
            println("State: ${detector.stateValue}")
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeAnomalyDetectors, na Referência de APIs do AWS SDK para Kotlin.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`DisableAlarmActions`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `DisableAlarmActions`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- Começar a usar alarmes
- Gerencie métricas e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
    /// <summary>
    /// Disable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableAlarmActions(List<string> alarmNames)
    {
        var disableAlarmActionsResult = await
_amazonCloudWatch.DisableAlarmActionsAsync(
            new DisableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });
```

```
        return disableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obter detalhes da API, consulte DisableAlarmActions na Referência da API AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Inclua os arquivos necessários.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DisableAlarmActionsRequest.h>
#include <iostream>
```

Desabilite as ações de alarme.

```
        Aws::CloudWatch::CloudWatchClient cw;

        Aws::CloudWatch::Model::DisableAlarmActionsRequest
disableAlarmActionsRequest;
        disableAlarmActionsRequest.AddAlarmNames(alarm_name);

        auto disableAlarmActionsOutcome =
cw.DisableAlarmActions(disableAlarmActionsRequest);
        if (!disableAlarmActionsOutcome.IsSuccess())
        {
            std::cout << "Failed to disable actions for alarm " << alarm_name <<
                ": " << disableAlarmActionsOutcome.GetError().GetMessage() <<
                std::endl;
```

```
        }
        else
        {
            std::cout << "Successfully disabled actions for alarm " <<
                alarm_name << std::endl;
        }
```

- Para obter detalhes da API, consulte DisableAlarmActions na Referência da API AWS SDK for C++.

CLI

AWS CLI

Como desabilitar ações de um alarme

O seguinte exemplo usa o comando `disable-alarm-actions` para desabilitar todas as ações do alarme “myalarm”:

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

Esse comando retornará à solicitação, se houver êxito.

- Para obter detalhes da API, consulte DisableAlarmActions na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
```

```
import
 software.amazon.awssdk.services.cloudwatch.model.DisableAlarmActionsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DisableAlarmActions {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to disable (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarmName = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        disableActions(cw, alarmName);
        cw.close();
    }

    public static void disableActions(CloudWatchClient cw, String alarmName) {
        try {
            DisableAlarmActionsRequest request =
 DisableAlarmActionsRequest.builder()
                    .alarmNames(alarmName)
                    .build();
```

```
            cw.disableAlarmActions(request);
            System.out.printf("Successfully disabled actions on alarm %s",
 alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte DisableAlarmActions na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Importe o SDK e os módulos do cliente e chame a API.

```
import { DisableAlarmActionsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DisableAlarmActionsCommand({
    AlarmNames: process.env.CLOUDWATCH_ALARM_NAME, // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
```

```
  }
};

export default run();
```

Crie o cliente em um módulo separado e exporte-o.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte DisableAlarmActions na Referência da API AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Importe o SDK e os módulos do cliente e chame a API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

cw.disableAlarmActions(
  { AlarmNames: ["Web_Server_CPU_Utilization"] },
  function (err, data) {
    if (err) {
      console.log("Error", err);
```

```
    } else {
      console.log("Success", data);
    }
  }
);
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte DisableAlarmActions na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
suspend fun disableActions(alarmName: String) {

    val request = DisableAlarmActionsRequest {
        alarmNames = listOf(alarmName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.disableAlarmActions(request)
        println("Successfully disabled actions on alarm $alarmName")
    }
}
```

- Para obter detalhes da API, consulte DisableAlarmActions na Referência da API AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def enable_alarm_actions(self, alarm_name, enable):
        """
        Enables or disables actions on the specified alarm. Alarm actions can be
        used to send notifications or automate responses when an alarm enters a
        particular state.

        :param alarm_name: The name of the alarm.
        :param enable: When True, actions are enabled for the alarm. Otherwise,
 they
                       disabled.
        """
        try:
            alarm = self.cloudwatch_resource.Alarm(alarm_name)
            if enable:
                alarm.enable_actions()
            else:
                alarm.disable_actions()
            logger.info(
                "%s actions for alarm %s.",
                "Enabled" if enable else "Disabled",
                alarm_name,
            )
```

```
        except ClientError:
            logger.exception(
                "Couldn't %s actions alarm %s.",
                "enable" if enable else "disable",
                alarm_name,
            )
            raise
```

- Para obter detalhes da API, consulte DisableAlarmActions na Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
# Disables an alarm in Amazon CloudWatch.
#
# Prerequisites.
#
# - The alarm to disable.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm to disable.
# @return [Boolean] true if the alarm was disabled; otherwise, false.
# @example
#   exit 1 unless alarm_actions_disabled?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket'
#   )
def alarm_actions_disabled?(cloudwatch_client, alarm_name)
  cloudwatch_client.disable_alarm_actions(alarm_names: [alarm_name])
  return true
```

```
rescue StandardError => e
  puts "Error disabling alarm actions: #{e.message}"
  return false
end

# Example usage:
def run_me
  alarm_name = "ObjectsInBucket"
  alarm_description = "Objects exist in this bucket for more than 1 day."
  metric_name = "NumberOfObjects"
  # Notify this Amazon Simple Notification Service (Amazon SNS) topic when
  # the alarm transitions to the ALARM state.
  alarm_actions = ["arn:aws:sns:us-
east-1:111111111111:Default_CloudWatch_Alarms_Topic"]
  namespace = "AWS/S3"
  statistic = "Average"
  dimensions = [
    {
      name: "BucketName",
      value: "doc-example-bucket"
    },
    {
      name: "StorageType",
      value: "AllStorageTypes"
    }
  ]
  period = 86_400 # Daily (24 hours * 60 minutes * 60 seconds = 86400 seconds).
  unit = "Count"
  evaluation_periods = 1 # More than one day.
  threshold = 1 # One object.
  comparison_operator = "GreaterThanThreshold" # More than one object.
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  if alarm_created_or_updated?(
    cloudwatch_client,
    alarm_name,
    alarm_description,
    metric_name,
    alarm_actions,
    namespace,
    statistic,
```

```
    dimensions,
    period,
    unit,
    evaluation_periods,
    threshold,
    comparison_operator
  )
    puts "Alarm '#{alarm_name}' created or updated."
  else
    puts "Could not create or update alarm '#{alarm_name}'."
  end

  if alarm_actions_disabled?(cloudwatch_client, alarm_name)
    puts "Alarm '#{alarm_name}' disabled."
  else
    puts "Could not disable alarm '#{alarm_name}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte DisableAlarmActions na Referência da API AWS SDK for Ruby.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    "Disables actions on the specified alarm. "
    TRY.
        lo_cwt->disablealarmactions(
          it_alarmnames = it_alarm_names
        ).
```

```
        MESSAGE 'Alarm actions disabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obter detalhes sobre a API, consulte DisableAlarmActions na Referência de APIs do AWS SDK para SAP ABAP.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`EnableAlarmActions`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `EnableAlarmActions`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Gerencie métricas e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
    /// <summary>
    /// Enable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
```

```
    public async Task<bool> EnableAlarmActions(List<string> alarmNames)
    {
        var enableAlarmActionsResult = await
_amazonCloudWatch.EnableAlarmActionsAsync(
            new EnableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return enableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obter detalhes da API, consulte EnableAlarmActions em Referência da API AWS SDK for .NET .

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Inclua os arquivos necessários.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/EnableAlarmActionsRequest.h>
#include <aws/monitoring/model/PutMetricAlarmRequest.h>
#include <iostream>
```

Habilite as ações de alarme.

```
    Aws::CloudWatch::CloudWatchClient cw;
    Aws::CloudWatch::Model::PutMetricAlarmRequest request;
    request.SetAlarmName(alarm_name);
```

```
    request.SetComparisonOperator(
        Aws::CloudWatch::Model::ComparisonOperator::GreaterThanThreshold);
    request.SetEvaluationPeriods(1);
    request.SetMetricName("CPUUtilization");
    request.SetNamespace("AWS/EC2");
    request.SetPeriod(60);
    request.SetStatistic(Aws::CloudWatch::Model::Statistic::Average);
    request.SetThreshold(70.0);
    request.SetActionsEnabled(false);
    request.SetAlarmDescription("Alarm when server CPU exceeds 70%");
    request.SetUnit(Aws::CloudWatch::Model::StandardUnit::Seconds);
    request.AddAlarmActions(actionArn);

    Aws::CloudWatch::Model::Dimension dimension;
    dimension.SetName("InstanceId");
    dimension.SetValue(instanceId);
    request.AddDimensions(dimension);

    auto outcome = cw.PutMetricAlarm(request);
    if (!outcome.IsSuccess())
    {
        std::cout << "Failed to create CloudWatch alarm:" <<
            outcome.GetError().GetMessage() << std::endl;
        return;
    }

    Aws::CloudWatch::Model::EnableAlarmActionsRequest enable_request;
    enable_request.AddAlarmNames(alarm_name);

    auto enable_outcome = cw.EnableAlarmActions(enable_request);
    if (!enable_outcome.IsSuccess())
    {
        std::cout << "Failed to enable alarm actions:" <<
            enable_outcome.GetError().GetMessage() << std::endl;
        return;
    }

    std::cout << "Successfully created alarm " << alarm_name <<
        " and enabled actions on it." << std::endl;
```

- Para obter detalhes da API, consulte EnableAlarmActions em Referência da API AWS SDK for C++ .

CLI

AWS CLI

Como habilitar todas as ações de um alarme

O seguinte exemplo usa o comando `enable-alarm-actions` para habilitar todas as ações para o alarme “myalarm”:

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

Esse comando retornará à solicitação, se houver êxito.

- Para obter detalhes da API, consulte EnableAlarmActions na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import
 software.amazon.awssdk.services.cloudwatch.model.EnableAlarmActionsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class EnableAlarmActions {
```

```
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to enable (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarm = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        enableActions(cw, alarm);
        cw.close();
    }

    public static void enableActions(CloudWatchClient cw, String alarm) {
        try {
            EnableAlarmActionsRequest request =
 EnableAlarmActionsRequest.builder()
                    .alarmNames(alarm)
                    .build();

            cw.enableAlarmActions(request);
            System.out.printf("Successfully enabled actions on alarm %s", alarm);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte EnableAlarmActions em Referência da API AWS SDK for Java 2.x .

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Importe o SDK e os módulos do cliente e chame a API.

```
import { EnableAlarmActionsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new EnableAlarmActionsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Crie o cliente em um módulo separado e exporte-o.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte EnableAlarmActions em Referência da API AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Importe o SDK e os módulos do cliente e chame a API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmName: "Web_Server_CPU_Utilization",
  ComparisonOperator: "GreaterThanThreshold",
  EvaluationPeriods: 1,
  MetricName: "CPUUtilization",
  Namespace: "AWS/EC2",
  Period: 60,
  Statistic: "Average",
  Threshold: 70.0,
  ActionsEnabled: true,
  AlarmActions: ["ACTION_ARN"],
  AlarmDescription: "Alarm when server CPU exceeds 70%",
  Dimensions: [
    {
      Name: "InstanceId",
      Value: "INSTANCE_ID",
    },
  ],
  Unit: "Percent",
```

```
};

cw.putMetricAlarm(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Alarm action added", data);
    var paramsEnableAlarmAction = {
      AlarmNames: [params.AlarmName],
    };
    cw.enableAlarmActions(paramsEnableAlarmAction, function (err, data) {
      if (err) {
        console.log("Error", err);
      } else {
        console.log("Alarm action enabled", data);
      }
    });
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte EnableAlarmActions em Referência da API AWS SDK for JavaScript .

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
suspend fun enableActions(alarm: String) {

    val request = EnableAlarmActionsRequest {
        alarmNames = listOf(alarm)
    }
```

```
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.enableAlarmActions(request)
        println("Successfully enabled actions on alarm $alarm")
    }
}
```

- Para obter detalhes da API, consulte EnableAlarmActions em Referência da API AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def enable_alarm_actions(self, alarm_name, enable):
        """
        Enables or disables actions on the specified alarm. Alarm actions can be
        used to send notifications or automate responses when an alarm enters a
        particular state.

        :param alarm_name: The name of the alarm.
        :param enable: When True, actions are enabled for the alarm. Otherwise,
 they
                       disabled.
```

```
        """
        try:
            alarm = self.cloudwatch_resource.Alarm(alarm_name)
            if enable:
                alarm.enable_actions()
            else:
                alarm.disable_actions()
            logger.info(
                "%s actions for alarm %s.",
                "Enabled" if enable else "Disabled",
                alarm_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't %s actions alarm %s.",
                "enable" if enable else "disable",
                alarm_name,
            )
            raise
```

- Para obter detalhes da API, consulte EnableAlarmActions na Referência da API AWS SDK para Python (Boto3).

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
"Enable actions on the specified alarm."
TRY.
    lo_cwt->enablealarmactions(
      it_alarmnames = it_alarm_names
    ).
```

```
        MESSAGE 'Alarm actions enabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Para obter detalhes sobre a API, consulte EnableAlarmActions, na Referência de APIs do AWS SDK para SAP ABAP.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetDashboard`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `GetDashboard`.

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
    /// <summary>
    /// Get information on a dashboard.
    /// </summary>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A JSON object with dashboard information.</returns>
    public async Task<string> GetDashboard(string dashboardName)
    {
        var dashboardResponse = await _amazonCloudWatch.GetDashboardAsync(
            new GetDashboardRequest()
            {
                DashboardName = dashboardName
```

```
            });

        return dashboardResponse.DashboardBody;
    }
```

- Para obter detalhes da API, consulte GetDashboard na Referência da API AWS SDK for .NET .

PowerShell

Tools for PowerShell

Exemplo 1: retorna o ARN do corpo do painel especificado.

```
Get-CWDashboard -DashboardName Dashboard1
```

Saída:

```
DashboardArn                                              DashboardBody
------------                                              -------------
arn:aws:cloudwatch::123456789012:dashboard/Dashboard1 {...
```

- Para obter detalhes da API, consulte GetDashboard na Referência de cmdlet do AWS Tools for PowerShell.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetMetricData`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `GetMetricData`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar métricas, painéis e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Get data for CloudWatch metrics.
    /// </summary>
    /// <param name="minutesOfData">The number of minutes of data to include.</
param>
    /// <param name="useDescendingTime">True to return the data descending by
 time.</param>
    /// <param name="endDateUtc">The end date for the data, in UTC.</param>
    /// <param name="maxDataPoints">The maximum data points to include.</param>
    /// <param name="dataQueries">Optional data queries to include.</param>
    /// <returns>A list of the requested metric data.</returns>
    public async Task<List<MetricDataResult>> GetMetricData(int minutesOfData,
 bool useDescendingTime, DateTime? endDateUtc = null,
        int maxDataPoints = 0, List<MetricDataQuery>? dataQueries = null)
    {
        var metricData = new List<MetricDataResult>();
        // If no end time is provided, use the current time for the end time.
        endDateUtc ??= DateTime.UtcNow;
        var timeZoneOffset =
 TimeZoneInfo.Local.GetUtcOffset(endDateUtc.Value.ToLocalTime());
        var startTimeUtc = endDateUtc.Value.AddMinutes(-minutesOfData);
        // The timezone string should be in the format +0000, so use the timezone
 offset to format it correctly.
        var timeZoneString = $"{timeZoneOffset.Hours:D2}
{timeZoneOffset.Minutes:D2}";
        var paginatedMetricData = _amazonCloudWatch.Paginators.GetMetricData(
            new GetMetricDataRequest()
            {
                StartTimeUtc = startTimeUtc,
                EndTimeUtc = endDateUtc.Value,
                LabelOptions = new LabelOptions { Timezone = timeZoneString },
```

```
                ScanBy = useDescendingTime ? ScanBy.TimestampDescending :
ScanBy.TimestampAscending,
                MaxDatapoints = maxDataPoints,
                MetricDataQueries = dataQueries,
            });

        await foreach (var data in paginatedMetricData.MetricDataResults)
        {
            metricData.Add(data);
        }
        return metricData;
    }
```

- Para obter detalhes da API, consulte GetMetricData, na Referência da API AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    public static void getCustomMetricData(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set the date.
```

```
            Instant nowDate = Instant.now();

            long hours = 1;
            long minutes = 30;
            Instant date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(minutes,
                    ChronoUnit.MINUTES);

            Metric met = Metric.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            MetricStat metStat = MetricStat.builder()
                    .stat("Maximum")
                    .period(1)
                    .metric(met)
                    .build();

            MetricDataQuery dataQUery = MetricDataQuery.builder()
                    .metricStat(metStat)
                    .id("foo2")
                    .returnData(true)
                    .build();

            List<MetricDataQuery> dq = new ArrayList<>();
            dq.add(dataQUery);

            GetMetricDataRequest getMetReq = GetMetricDataRequest.builder()
                    .maxDatapoints(10)
                    .scanBy(ScanBy.TIMESTAMP_DESCENDING)
                    .startTime(nowDate)
                    .endTime(date2)
                    .metricDataQueries(dq)
                    .build();

            GetMetricDataResponse response = cw.getMetricData(getMetReq);
            List<MetricDataResult> data = response.metricDataResults();
            for (MetricDataResult item : data) {
                System.out.println("The label is " + item.label());
                System.out.println("The status code is " +
item.statusCode().toString());
            }

        } catch (CloudWatchException | IOException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte GetMetricData, na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun getCustomMetricData(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set the date.
    val nowDate = Instant.now()
    val hours: Long = 1
    val minutes: Long = 30
    val date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(
        minutes,
        ChronoUnit.MINUTES
    )

    val met = Metric {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
```

```
    val metStat = MetricStat {
        stat = "Maximum"
        period = 1
        metric = met
    }

    val dataQUery = MetricDataQuery {
        metricStat = metStat
        id = "foo2"
        returnData = true
    }

    val dq = ArrayList<MetricDataQuery>()
    dq.add(dataQUery)
    val getMetReq = GetMetricDataRequest {
        maxDatapoints = 10
        scanBy = ScanBy.TimestampDescending
        startTime = aws.smithy.kotlin.runtime.time.Instant(nowDate)
        endTime = aws.smithy.kotlin.runtime.time.Instant(date2)
        metricDataQueries = dq
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricData(getMetReq)
        response.metricDataResults?.forEach { item ->
            println("The label is ${item.label}")
            println("The status code is ${item.statusCode}")
        }
    }
}
```

- Para obter detalhes da API, consulte GetMetricData, na Referência de APIs do AWS SDK para Kotlin.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetMetricStatistics`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `GetMetricStatistics`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- Começar a usar métricas, painéis e alarmes
- Gerencie métricas e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Get billing statistics using a call to a wrapper class.
    /// </summary>
    /// <returns>A collection of billing statistics.</returns>
    private static async Task<List<Datapoint>> SetupBillingStatistics()
    {
        // Make a request for EstimatedCharges with a period of one day for the
past seven days.
        var billingStatistics = await _cloudWatchWrapper.GetMetricStatistics(
            "AWS/Billing",
            "EstimatedCharges",
            new List<string>() { "Maximum" },
            new List<Dimension>() { new Dimension { Name = "Currency", Value =
"USD" } },
            7,
            86400);

        billingStatistics = billingStatistics.OrderBy(n => n.Timestamp).ToList();

        return billingStatistics;
    }

    /// <summary>
    /// Wrapper to get statistics for a specific CloudWatch metric.
    /// </summary>
```

```
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <param name="statistics">The list of statistics to include.</param>
    /// <param name="dimensions">The list of dimensions to include.</param>
    /// <param name="days">The number of days in the past to include.</param>
    /// <param name="period">The period for the data.</param>
    /// <returns>A list of DataPoint objects for the statistics.</returns>
    public async Task<List<Datapoint>> GetMetricStatistics(string
metricNamespace,
        string metricName, List<string> statistics, List<Dimension> dimensions,
int days, int period)
    {
        var metricStatistics = await _amazonCloudWatch.GetMetricStatisticsAsync(
            new GetMetricStatisticsRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName,
                Dimensions = dimensions,
                Statistics = statistics,
                StartTimeUtc = DateTime.UtcNow.AddDays(-days),
                EndTimeUtc = DateTime.UtcNow,
                Period = period
            });

        return metricStatistics.Datapoints;
    }
```

- Para obter detalhes da API, consulte GetMetricStatistics, na Referência da API AWS SDK for .NET.

CLI

AWS CLI

Como obter a utilização da CPU por instância do EC2

O exemplo a seguir usa o comando `get-metric-statistics` para obter a utilização da CPU de uma instância do EC2 com o ID i-abcdef.

```
aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time
 2014-04-08T23:18:00Z --end-time 2014-04-09T23:18:00Z --period 3600 --namespace
 AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-abcdef
```

Saída:

```
{
    "Datapoints": [
        {
            "Timestamp": "2014-04-09T11:18:00Z",
            "Maximum": 44.79,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T20:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T19:18:00Z",
            "Maximum": 50.85,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T09:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T03:18:00Z",
            "Maximum": 76.84,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T21:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T14:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
```

```
        {
            "Timestamp": "2014-04-09T08:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T16:18:00Z",
            "Maximum": 45.55,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T06:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T13:18:00Z",
            "Maximum": 45.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T05:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T18:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T17:18:00Z",
            "Maximum": 52.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T07:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T02:18:00Z",
            "Maximum": 51.23,
            "Unit": "Percent"
```

```
        },
        {
            "Timestamp": "2014-04-09T12:18:00Z",
            "Maximum": 47.67,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-08T23:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T10:18:00Z",
            "Maximum": 51.91,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T04:18:00Z",
            "Maximum": 47.13,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T15:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T00:18:00Z",
            "Maximum": 48.16,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T01:18:00Z",
            "Maximum": 49.18,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```


Especificar várias dimensões

O exemplo a seguir ilustra como especificar diversas dimensões. Cada dimensão é especificada como um par de nome/valor, com uma vírgula entre o nome e o valor. As diversas dimensões são separadas por um espaço. Se uma única métrica incluir diversas dimensões, você deverá especificar um valor para cada dimensão definida.

Para obter mais exemplos relacionados ao uso do comando `get-metric-statistics`, consulte Obter estatísticas de uma métrica no Guia do desenvolvedor do Amazon CloudWatch.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --
namespace MyNameSpace --dimensions Name=InstanceID,Value=i-abcdef
 Name=InstanceType,Value=m1.small --start-time 2016-10-15T04:00:00Z --end-time
 2016-10-19T07:00:00Z --statistics Average --period 60
```

- Para obter detalhes da API, consulte GetMetricStatistics na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   public static void getAndDisplayMetricStatistics(CloudWatchClient cw, String
nameSpace, String metVal,
           String metricOption, String date, Dimension myDimension) {
       try {
           Instant start = Instant.parse(date);
           Instant endDate = Instant.now();

           GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                   .endTime(endDate)
                   .startTime(start)
                   .dimensions(myDimension)
                   .metricName(metVal)
```

```
                    .namespace(nameSpace)
                    .period(86400)
                    .statistics(Statistic.fromValue(metricOption))
                    .build();

            GetMetricStatisticsResponse response =
 cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
                            .println("Timestamp: " + datapoint.timestamp() + "
 Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte GetMetricStatistics, na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun getAndDisplayMetricStatistics(nameSpaceVal: String, metVal: String,
 metricOption: String, date: String, myDimension: Dimension) {
    val start = Instant.parse(date)
```

```
    val endDate = Instant.now()
    val statisticsRequest = GetMetricStatisticsRequest {
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        dimensions = listOf(myDimension)
        metricName = metVal
        namespace = nameSpaceVal
        period = 86400
        statistics = listOf(Statistic.fromValue(metricOption))
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data = response.datapoints
        if (data != null) {
            if (data.isNotEmpty()) {
                for (datapoint in data) {
                    println("Timestamp: ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}
```

- Para obter detalhes da API, consulte GetMetricStatistics, na Referência da API AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def get_metric_statistics(self, namespace, name, start, end, period,
 stat_types):
        """
        Gets statistics for a metric within a specified time span. Metrics are
 grouped
        into the specified period.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param start: The UTC start time of the time span to retrieve.
        :param end: The UTC end time of the time span to retrieve.
        :param period: The period, in seconds, in which to group metrics. The
 period
                       must match the granularity of the metric, which depends on
                       the metric's age. For example, metrics that are older than
                       three hours have a one-minute granularity, so the period
 must
                       be at least 60 and must be a multiple of 60.
        :param stat_types: The type of statistics to retrieve, such as average
 value
                           or maximum value.
        :return: The retrieved statistics for the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            stats = metric.get_statistics(
                StartTime=start, EndTime=end, Period=period,
 Statistics=stat_types
            )
            logger.info(
                "Got %s statistics for %s.", len(stats["Datapoints"]),
 stats["Label"]
            )
```

```
        except ClientError:
            logger.exception("Couldn't get statistics for %s.%s.", namespace,
name)
            raise
        else:
            return stats
```

- Para obter detalhes da API, consulte GetMetricStatistics na Referência da API AWS SDK para Python (Boto3).

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`GetMetricWidgetImage`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `GetMetricWidgetImage`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar métricas, painéis e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Get an image for a metric graphed over time.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metric">The name of the metric.</param>
```

```
    /// <param name="stat">The name of the stat to chart.</param>
    /// <param name="period">The period to use for the chart.</param>
    /// <returns>A memory stream for the chart image.</returns>
    public async Task<MemoryStream> GetTimeSeriesMetricImage(string
metricNamespace, string metric, string stat, int period)
    {
        var metricImageWidget = new
        {
            title = "Example Metric Graph",
            view = "timeSeries",
            stacked = false,
            period = period,
            width = 1400,
            height = 600,
            metrics = new List<List<object>>
                { new() { metricNamespace, metric, new { stat } } }
        };

        var metricImageWidgetString =
JsonSerializer.Serialize(metricImageWidget);
        var imageResponse = await _amazonCloudWatch.GetMetricWidgetImageAsync(
            new GetMetricWidgetImageRequest()
            {
                MetricWidget = metricImageWidgetString
            });

        return imageResponse.MetricWidgetImage;
    }

    /// <summary>
    /// Save a metric image to a file.
    /// </summary>
    /// <param name="memoryStream">The MemoryStream for the metric image.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The path to the file.</returns>
    public string SaveMetricImage(MemoryStream memoryStream, string metricName)
    {
        var metricFileName = $"{metricName}_{DateTime.Now.Ticks}.png";
        using var sr = new StreamReader(memoryStream);
        // Writes the memory stream to a file.
        File.WriteAllBytes(metricFileName, memoryStream.ToArray());
        var filePath = Path.Join(AppDomain.CurrentDomain.BaseDirectory,
            metricFileName);
        return filePath;
```

```
    }
```

- Para obter detalhes da API, consulte GetMetricWidgetImage, na Referência da API AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   public static void getAndOpenMetricImage(CloudWatchClient cw, String
fileName) {
       System.out.println("Getting Image data for custom metric.");
       try {
           String myJSON = "{\n" +
                   "  \"title\": \"Example Metric Graph\",\n" +
                   "  \"view\": \"timeSeries\",\n" +
                   "  \"stacked \": false,\n" +
                   "  \"period\": 10,\n" +
                   "  \"width\": 1400,\n" +
                   "  \"height\": 600,\n" +
                   "  \"metrics\": [\n" +
                   "    [\n" +
                   "      \"AWS/Billing\",\n" +
                   "      \"EstimatedCharges\",\n" +
                   "      \"Currency\",\n" +
                   "      \"USD\"\n" +
                   "    ]\n" +
                   "  ]\n" +
                   "}";

           GetMetricWidgetImageRequest imageRequest =
GetMetricWidgetImageRequest.builder()
                   .metricWidget(myJSON)
                   .build();
```

```
            GetMetricWidgetImageResponse response =
 cw.getMetricWidgetImage(imageRequest);
            SdkBytes sdkBytes = response.metricWidgetImage();
            byte[] bytes = sdkBytes.asByteArray();
            File outputFile = new File(fileName);
            try (FileOutputStream outputStream = new
 FileOutputStream(outputFile)) {
                outputStream.write(bytes);
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte GetMetricWidgetImage, na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun getAndOpenMetricImage(fileName: String) {
    println("Getting Image data for custom metric.")
    val myJSON = """{
        "title": "Example Metric Graph",
        "view": "timeSeries",
        "stacked ": false,
        "period": 10,
        "width": 1400,
        "height": 600,
        "metrics": [
```

```
            [
            "AWS/Billing",
            "EstimatedCharges",
            "Currency",
            "USD"
            ]
        ]
        }"""

    val imageRequest = GetMetricWidgetImageRequest {
        metricWidget = myJSON
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricWidgetImage(imageRequest)
        val bytes = response.metricWidgetImage
        if (bytes != null) {
            File(fileName).writeBytes(bytes)
        }
    }
    println("You have successfully written data to $fileName")
}
```

- Para obter detalhes da API, consulte GetMetricWidgetImage, na Referência de APIs do AWS SDK para Kotlin.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListDashboards`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `ListDashboards`.

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
/// <summary>
/// Get a list of dashboards.
/// </summary>
/// <returns>A list of DashboardEntry objects.</returns>
public async Task<List<DashboardEntry>> ListDashboards()
{
    var results = new List<DashboardEntry>();
    var paginateDashboards = _amazonCloudWatch.Paginators.ListDashboards(
        new ListDashboardsRequest());
    // Get the entire list using the paginator.
    await foreach (var data in paginateDashboards.DashboardEntries)
    {
        results.Add(data);
    }

    return results;
}
```

- Para obter detalhes da API, consulte ListDashboards na Referência da API AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    public static void listDashboards(CloudWatchClient cw) {
        try {
            ListDashboardsIterable listRes = cw.listDashboardsPaginator();
            listRes.stream()
                    .flatMap(r -> r.dashboardEntries().stream())
                    .forEach(entry -> {
                        System.out.println("Dashboard name is: " +
entry.dashboardName());
                        System.out.println("Dashboard ARN is: " +
entry.dashboardArn());
                    });

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte ListDashboards na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun listDashboards() {
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listDashboardsPaginated({})
            .transform { it.dashboardEntries?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.dashboardName}")
                println("Dashboard ARN is ${obj.dashboardArn}")
            }
    }
```

```
}
```

- Para obter detalhes da API, consulte ListDashboards, na Referência da API AWS SDK para Kotlin.

PowerShell

Tools for PowerShell

Exemplo 1: retorna a coleção de painéis para sua conta.

```
Get-CWDashboardList
```

Saída:

```
DashboardArn DashboardName LastModified        Size
------------ ------------- ------------        ----
arn:...      Dashboard1    7/6/2017 8:14:15 PM 252
```

Exemplo 2: retorna a coleção de painéis para sua conta cujos nomes começam com o prefixo "dev".

```
Get-CWDashboardList -DashboardNamePrefix dev
```

- Para obter detalhes da API, consulte ListDashboards na Referência de cmdlet do AWS Tools for PowerShell.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`ListMetrics`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `ListMetrics`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- Começar a usar métricas, painéis e alarmes

- Gerencie métricas e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
   /// <summary>
   /// List metrics available, optionally within a namespace.
   /// </summary>
   /// <param name="metricNamespace">Optional CloudWatch namespace to use when
listing metrics.</param>
   /// <param name="filter">Optional dimension filter.</param>
   /// <param name="metricName">Optional metric name filter.</param>
   /// <returns>The list of metrics.</returns>
   public async Task<List<Metric>> ListMetrics(string? metricNamespace = null,
DimensionFilter? filter = null, string? metricName = null)
   {
       var results = new List<Metric>();
       var paginateMetrics = _amazonCloudWatch.Paginators.ListMetrics(
           new ListMetricsRequest
           {
               Namespace = metricNamespace,
               Dimensions = filter != null ? new List<DimensionFilter>
{ filter } : null,
               MetricName = metricName
           });
       // Get the entire list using the paginator.
       await foreach (var metric in paginateMetrics.Metrics)
       {
           results.Add(metric);
       }

       return results;
   }
```

- Para obter detalhes da API, consulte ListMetrics na Referência da API AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Inclua os arquivos necessários.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/ListMetricsRequest.h>
#include <aws/monitoring/model/ListMetricsResult.h>
#include <iomanip>
#include <iostream>
```

Liste as métricas.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::ListMetricsRequest request;

        if (argc > 1)
        {
            request.SetMetricName(argv[1]);
        }

        if (argc > 2)
        {
            request.SetNamespace(argv[2]);
        }

        bool done = false;
        bool header = false;
        while (!done)
        {
```

```
            auto outcome = cw.ListMetrics(request);
            if (!outcome.IsSuccess())
            {
                std::cout << "Failed to list CloudWatch metrics:" <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header)
            {
                std::cout << std::left << std::setw(48) << "MetricName" <<
                    std::setw(32) << "Namespace" << "DimensionNameValuePairs" <<
                    std::endl;
                header = true;
            }

            const auto &metrics = outcome.GetResult().GetMetrics();
            for (const auto &metric : metrics)
            {
                std::cout << std::left << std::setw(48) <<
                    metric.GetMetricName() << std::setw(32) <<
                    metric.GetNamespace();
                const auto &dimensions = metric.GetDimensions();
                for (auto iter = dimensions.cbegin();
                    iter != dimensions.cend(); ++iter)
                {
                    const auto &dimkv = *iter;
                    std::cout << dimkv.GetName() << " = " << dimkv.GetValue();
                    if (iter + 1 != dimensions.cend())
                    {
                        std::cout << ", ";
                    }
                }
                std::cout << std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Para obter detalhes da API, consulte ListMetrics na Referência da API AWS SDK for C++.

CLI

AWS CLI

Como listar as métricas do Amazon SNS

O exemplo apresentado a seguir para `list-metrics` exibe as métricas do Amazon SNS.

```
aws cloudwatch list-metrics \
    --namespace "AWS/SNS"
```

Saída:

```
{
    "Metrics": [
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
```

```
            "MetricName": "NumberOfNotificationsFailed"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
```

```
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        }
    ]
}
```

- Para obter detalhes da API, consulte ListMetrics na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Metric;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListMetrics {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:
                  <namespace>\s

                Where:
                  namespace - The namespace to filter against (for example, AWS/
EC2).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String namespace = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        listMets(cw, namespace);
        cw.close();
    }

    public static void listMets(CloudWatchClient cw, String namespace) {
        boolean done = false;
        String nextToken = null;

        try {
            while (!done) {

                ListMetricsResponse response;
                if (nextToken == null) {
                    ListMetricsRequest request = ListMetricsRequest.builder()
                            .namespace(namespace)
                            .build();

                    response = cw.listMetrics(request);
                } else {
                    ListMetricsRequest request = ListMetricsRequest.builder()
                            .namespace(namespace)
                            .nextToken(nextToken)
                            .build();
```

```
                    response = cw.listMetrics(request);
                }

                for (Metric metric : response.metrics()) {
                    System.out.printf("Retrieved metric %s",
 metric.metricName());
                    System.out.println();
                }

                if (response.nextToken() == null) {
                    done = true;
                } else {
                    nextToken = response.nextToken();
                }
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte ListMetrics na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Importe o SDK e os módulos do cliente e chame a API.

```
import { ListMetricsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";
```

```
export const main = () => {
  // Use the AWS console to see available namespaces and metric names. Custom
 metrics can also be created.
  // https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
viewing_metrics_with_cloudwatch.html
  const command = new ListMetricsCommand({
    Dimensions: [
      {
        Name: "LogGroupName",
      },
    ],
    MetricName: "IncomingLogEvents",
    Namespace: "AWS/Logs",
  });

  return client.send(command);
};
```

Crie o cliente em um módulo separado e exporte-o.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte ListMetrics na Referência da API do AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
```

```
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  Dimensions: [
    {
      Name: "LogGroupName" /* required */,
    },
  ],
  MetricName: "IncomingLogEvents",
  Namespace: "AWS/Logs",
};

cw.listMetrics(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Metrics", JSON.stringify(data.Metrics));
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte ListMetrics na Referência da API do AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun listMets(namespaceVal: String?): ArrayList<String>? {
    val metList = ArrayList<String>()
```

```
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val reponse = cwClient.listMetrics(request)
        reponse.metrics?.forEach { metrics ->
            val data = metrics.metricName
            if (!metList.contains(data)) {
                metList.add(data!!)
            }
        }
    }
    return metList
}
```

- Para obter detalhes da API, consulte ListMetrics na Referência da API AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def list_metrics(self, namespace, name, recent=False):
        """
```

```
        Gets the metrics within a namespace that have the specified name.
        If the metric has no dimensions, a single metric is returned.
        Otherwise, metrics for all dimensions are returned.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param recent: When True, only metrics that have been active in the last
                       three hours are returned.
        :return: An iterator that yields the retrieved metrics.
        """
        try:
            kwargs = {"Namespace": namespace, "MetricName": name}
            if recent:
                kwargs["RecentlyActive"] = "PT3H"  # List past 3 hours only
            metric_iter = self.cloudwatch_resource.metrics.filter(**kwargs)
            logger.info("Got metrics for %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't get metrics for %s.%s.", namespace, name)
            raise
        else:
            return metric_iter
```

- Para obter detalhes da API, consulte ListMetrics na Referência da API AWS SDK for Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
# Lists available metrics for a metric namespace in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
```

```
# @param metric_namespace [String] The namespace of the metric.
# @example
#   list_metrics_for_namespace(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC'
#   )
def list_metrics_for_namespace(cloudwatch_client, metric_namespace)
  response = cloudwatch_client.list_metrics(namespace: metric_namespace)

  if response.metrics.count.positive?
    response.metrics.each do |metric|
      puts "  Metric name: #{metric.metric_name}"
      if metric.dimensions.count.positive?
        puts "    Dimensions:"
        metric.dimensions.each do |dimension|
          puts "      Name: #{dimension.name}, Value: #{dimension.value}"
        end
      else
        puts "No dimensions found."
      end
    end
  else
    puts "No metrics found for namespace '#{metric_namespace}'. " \
      "Note that it could take up to 15 minutes for recently-added metrics " \
      "to become available."
  end
end

# Example usage:
def run_me
  metric_namespace = "SITE/TRAFFIC"
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  # Add three datapoints.
  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "UniqueVisitors",
    "SiteName",
    "example.com",
    5_885.0,
```

```
    "Count"
  )

  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "UniqueVisits",
    "SiteName",
    "example.com",
    8_628.0,
    "Count"
  )

  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "PageViews",
    "PageURL",
    "example.html",
    18_057.0,
    "Count"
  )

  puts "Metrics for namespace '#{metric_namespace}':"
  list_metrics_for_namespace(cloudwatch_client, metric_namespace)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte ListMetrics na Referência da API AWS SDK for Ruby.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   "The following list-metrics example displays the metrics for Amazon
CloudWatch."
   TRY.
       oo_result = lo_cwt->listmetrics(            " oo_result is returned for
testing purposes. "
         iv_namespace = iv_namespace
       ).
       DATA(lt_metrics) = oo_result->get_metrics( ).
       MESSAGE 'Metrics retrieved.' TYPE 'I'.
     CATCH /aws1/cx_cwtinvparamvalueex .
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
   ENDTRY.
```

- Para obter detalhes sobre a API, consulte ListMetrics na Referência de APIs do AWS SDK para SAP ABAP.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`PutAnomalyDetector`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `PutAnomalyDetector`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar métricas, painéis e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Add an anomaly detector for a single metric.
    /// </summary>
    /// <param name="anomalyDetector">A single metric anomaly detector.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
    {
        var putAlarmDetectorResult = await
_amazonCloudWatch.PutAnomalyDetectorAsync(
            new PutAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return putAlarmDetectorResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obter detalhes sobre a API, consulte PutAnomalyDetector, na Referência da API AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    public static void addAnomalyDetector(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
```

```
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .stat("Maximum")
                    .build();

            PutAnomalyDetectorRequest anomalyDetectorRequest =
PutAnomalyDetectorRequest.builder()
                    .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                    .build();

            cw.putAnomalyDetector(anomalyDetectorRequest);
            System.out.println("Added anomaly detector for metric " +
customMetricName + ".");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes sobre a API, consulte PutAnomalyDetector, na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun addAnomalyDetector(fileName: String?) {
    // Read values from the JSON file.
```

```
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val anomalyDetectorRequest = PutAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putAnomalyDetector(anomalyDetectorRequest)
        println("Added anomaly detector for metric $customMetricName.")
    }
}
```

- Para obter detalhes da API, consulte PutAnomalyDetector, na Referência de APIs do AWS SDK para Kotlin.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`PutDashboard`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `PutDashboard`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Começar a usar métricas, painéis e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    /// <summary>
    /// Set up a dashboard using a call to the wrapper class.
    /// </summary>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A list of validation messages.</returns>
    private static async Task<List<DashboardValidationMessage>> SetupDashboard(
        string customMetricNamespace, string customMetricName, string
dashboardName)
    {
        // Get the dashboard model from configuration.
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);

        // Add a new metric to the dashboard.
        newDashboard.Widgets.Add(new Widget
        {
            Height = 8,
            Width = 8,
            Y = 8,
            X = 0,
            Type = "metric",
            Properties = new Properties
            {
                Metrics = new List<List<object>>
                    { new() { customMetricNamespace, customMetricName } },
                View = "timeSeries",
                Region = "us-east-1",
                Stat = "Sum",
                Period = 86400,
```

```
                YAxis = new YAxis { Left = new Left { Min = 0, Max = 100 } },
                Title = "Custom Metric Widget",
                LiveData = true,
                Sparkline = true,
                Trend = true,
                Stacked = false,
                SetPeriodToTimeRange = false
            }
        });

        var newDashboardString = JsonSerializer.Serialize(newDashboard,
            new JsonSerializerOptions
            { DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
 newDashboardString);

        return validationMessages;
    }

    /// <summary>
    /// Wrapper to create or add to a dashboard with metrics.
    /// </summary>
    /// <param name="dashboardName">The name for the dashboard.</param>
    /// <param name="dashboardBody">The metric data in JSON for the dashboard.</
param>
    /// <returns>A list of validation messages for the dashboard.</returns>
    public async Task<List<DashboardValidationMessage>> PutDashboard(string
 dashboardName,
        string dashboardBody)
    {
        // Updating a dashboard replaces all contents.
        // Best practice is to include a text widget indicating this dashboard
 was created programmatically.
        var dashboardResponse = await _amazonCloudWatch.PutDashboardAsync(
            new PutDashboardRequest()
            {
                DashboardName = dashboardName,
                DashboardBody = dashboardBody
            });

        return dashboardResponse.DashboardValidationMessages;
    }
```

- Para obter detalhes da API, consulte PutDashboard na Referência da API AWS SDK for .NET.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   public static void createDashboardWithMetrics(CloudWatchClient cw, String
dashboardName, String fileName) {
       try {
           PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                   .dashboardName(dashboardName)
                   .dashboardBody(readFileAsString(fileName))
                   .build();

           PutDashboardResponse response = cw.putDashboard(dashboardRequest);
           System.out.println(dashboardName + " was successfully created.");
           List<DashboardValidationMessage> messages =
response.dashboardValidationMessages();
           if (messages.isEmpty()) {
               System.out.println("There are no messages in the new Dashboard");
           } else {
               for (DashboardValidationMessage message : messages) {
                   System.out.println("Message is: " + message.message());
               }
           }

       } catch (CloudWatchException | IOException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }
```

- Para obter detalhes da API, consulte PutDashboard na Referência da API AWS SDK for Java 2.x.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun createDashboardWithMetrics(dashboardNameVal: String, fileNameVal:
 String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully created.")
        val messages = response.dashboardValidationMessages
        if (messages != null) {
            if (messages.isEmpty()) {
                println("There are no messages in the new Dashboard")
            } else {
                for (message in messages) {
                    println("Message is: ${message.message}")
                }
            }
        }
    }
}
```

- Para obter detalhes da API, consulte PutDashboard na Referência da API AWS SDK para Kotlin.

PowerShell

Tools for PowerShell

Exemplo 1: cria ou atualiza o painel denominado “Dashboard1” para incluir dois widgets de métricas lado a lado.

```
$dashBody = @"
{
    "widgets":[
        {
             "type":"metric",
             "x":0,
             "y":0,
             "width":12,
             "height":6,
             "properties":{
                "metrics":[
                   [
                      "AWS/EC2",
                      "CPUUtilization",
                      "InstanceId",
                      "i-012345"
                   ]
                ],
                "period":300,
                "stat":"Average",
                "region":"us-east-1",
                "title":"EC2 Instance CPU"
             }
        },
        {
             "type":"metric",
             "x":12,
             "y":0,
             "width":12,
             "height":6,
             "properties":{
                "metrics":[
                   [
                      "AWS/S3",
                      "BucketSizeBytes",
                      "BucketName",
                      "MyBucketName"
```

```
                    ]
                ],
                "period":86400,
                "stat":"Maximum",
                "region":"us-east-1",
                "title":"MyBucketName bytes"
            }
        }
    ]
}
"@

Write-CWDashboard -DashboardName Dashboard1 -DashboardBody $dashBody
```

Exemplo 2: cria ou atualiza o painel, redirecionando o conteúdo que descreve o painel para o cmdlet.

```
$dashBody = @"
{
...
}
"@

$dashBody | Write-CWDashboard -DashboardName Dashboard1
```

- Para obter detalhes da API, consulte PutDashboard na Referência de cmdlet do AWS Tools for PowerShell.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`PutMetricAlarm`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `PutMetricAlarm`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- Começar a usar alarmes
- Começar a usar métricas, painéis e alarmes

- Gerencie métricas e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   /// <summary>
   /// Add a metric alarm to send an email when the metric passes a threshold.
   /// </summary>
   /// <param name="alarmDescription">A description of the alarm.</param>
   /// <param name="alarmName">The name for the alarm.</param>
   /// <param name="comparison">The type of comparison to use.</param>
   /// <param name="metricName">The name of the metric for the alarm.</param>
   /// <param name="metricNamespace">The namespace of the metric.</param>
   /// <param name="threshold">The threshold value for the alarm.</param>
   /// <param name="alarmActions">Optional actions to execute when in an alarm
state.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> PutMetricEmailAlarm(string alarmDescription, string
alarmName, ComparisonOperator comparison,
       string metricName, string metricNamespace, double threshold, List<string>
alarmActions = null!)
   {
       try
       {
           var putEmailAlarmResponse = await
_amazonCloudWatch.PutMetricAlarmAsync(
               new PutMetricAlarmRequest()
               {
                   AlarmActions = alarmActions,
                   AlarmDescription = alarmDescription,
                   AlarmName = alarmName,
                   ComparisonOperator = comparison,
                   Threshold = threshold,
                   Namespace = metricNamespace,
                   MetricName = metricName,
```

```
                    EvaluationPeriods = 1,
                    Period = 10,
                    Statistic = new Statistic("Maximum"),
                    DatapointsToAlarm = 1,
                    TreatMissingData = "ignore"
                });
            return putEmailAlarmResponse.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (LimitExceededException lex)
        {
            _logger.LogError(lex, $"Unable to add alarm {alarmName}. Alarm quota
 has already been reached.");
        }

        return false;
    }

    /// <summary>
    /// Add specific email actions to a list of action strings for a CloudWatch
 alarm.
    /// </summary>
    /// <param name="accountId">The AccountId for the alarm.</param>
    /// <param name="region">The region for the alarm.</param>
    /// <param name="emailTopicName">An Amazon Simple Notification Service (SNS)
 topic for the alarm email.</param>
    /// <param name="alarmActions">Optional list of existing alarm actions to
 append to.</param>
    /// <returns>A list of string actions for an alarm.</returns>
    public List<string> AddEmailAlarmAction(string accountId, string region,
        string emailTopicName, List<string>? alarmActions = null)
    {
        alarmActions ??= new List<string>();
        var snsAlarmAction = $"arn:aws:sns:{region}:{accountId}:
{emailTopicName}";
        alarmActions.Add(snsAlarmAction);
        return alarmActions;
    }
```

- Para obter detalhes da API, consulte PutMetricAlarm na Referência da API AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Inclua os arquivos necessários.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/PutMetricAlarmRequest.h>
#include <iostream>
```

Crie o alarme para vigiar a métrica.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::PutMetricAlarmRequest request;
        request.SetAlarmName(alarm_name);
        request.SetComparisonOperator(
            Aws::CloudWatch::Model::ComparisonOperator::GreaterThanThreshold);
        request.SetEvaluationPeriods(1);
        request.SetMetricName("CPUUtilization");
        request.SetNamespace("AWS/EC2");
        request.SetPeriod(60);
        request.SetStatistic(Aws::CloudWatch::Model::Statistic::Average);
        request.SetThreshold(70.0);
        request.SetActionsEnabled(false);
        request.SetAlarmDescription("Alarm when server CPU exceeds 70%");
        request.SetUnit(Aws::CloudWatch::Model::StandardUnit::Seconds);

        Aws::CloudWatch::Model::Dimension dimension;
        dimension.SetName("InstanceId");
        dimension.SetValue(instanceId);

        request.AddDimensions(dimension);

        auto outcome = cw.PutMetricAlarm(request);
```

```
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch alarm:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch alarm " << alarm_name
                << std::endl;
        }
```

- Para obter detalhes da API, consulte PutMetricAlarm na Referência da API AWS SDK for C ++.

CLI

AWS CLI

Como enviar uma mensagem de e-mail do Amazon Simple Notification Service quando a utilização da CPU exceder 70%

O seguinte exemplo usa o comando `put-metric-alarm` para enviar uma mensagem de e-mail do Amazon Simple Notification Service quando a utilização da CPU excede 70%:

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/
EC2 --statistic Average --period 300 --threshold 70 --comparison-operator
 GreaterThanThreshold  --dimensions "Name=InstanceId,Value=i-12345678" --
evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic
 --unit Percent
```

Esse comando retornará à solicitação, se houver êxito. Se já existir um alarme com o mesmo nome, ele será substituído pelo novo alarme.

Como especificar diversas dimensões

O exemplo a seguir ilustra como especificar diversas dimensões. Cada dimensão é especificada como um par de nome/valor, com uma vírgula entre o nome e o valor. As diversas dimensões são separadas por um espaço:

```
aws cloudwatch put-metric-alarm --alarm-name "Default_Test_Alarm3" --alarm-
description "The default example alarm" --namespace "CW EXAMPLE METRICS" --
metric-name Default_Test --statistic Average --period 60 --evaluation-periods 3
 --threshold 50 --comparison-operator GreaterThanOrEqualToThreshold --dimensions
 Name=key1,Value=value1 Name=key2,Value=value2
```

- Para obter detalhes da API, consulte PutMetricAlarm na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
    public static String createAlarm(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            String alarmName = rootNode.findValue("exampleAlarmName").asText();
            String emailTopic = rootNode.findValue("emailTopic").asText();
            String accountId = rootNode.findValue("accountId").asText();
            String region = rootNode.findValue("region").asText();

            // Create a List for alarm actions.
            List<String> alarmActions = new ArrayList<>();
            alarmActions.add("arn:aws:sns:" + region + ":" + accountId + ":" +
emailTopic);
            PutMetricAlarmRequest alarmRequest = PutMetricAlarmRequest.builder()
                    .alarmActions(alarmActions)
```

```
                    .alarmDescription("Example metric alarm")
                    .alarmName(alarmName)

.comparisonOperator(ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD)
                    .threshold(100.00)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .evaluationPeriods(1)
                    .period(10)
                    .statistic("Maximum")
                    .datapointsToAlarm(1)
                    .treatMissingData("ignore")
                    .build();

            cw.putMetricAlarm(alarmRequest);
            System.out.println(alarmName + " was successfully created!");
            return alarmName;

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }
```

- Para obter detalhes da API, consulte PutMetricAlarm na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Importe o SDK e os módulos do cliente e chame a API.

```
import { PutMetricAlarmCommand } from "@aws-sdk/client-cloudwatch";
```

```
import { client } from "../libs/client.js";

const run = async () => {
  // This alarm triggers when CPUUtilization exceeds 70% for one minute.
  const command = new PutMetricAlarmCommand({
    AlarmName: process.env.CLOUDWATCH_ALARM_NAME, // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
    ComparisonOperator: "GreaterThanThreshold",
    EvaluationPeriods: 1,
    MetricName: "CPUUtilization",
    Namespace: "AWS/EC2",
    Period: 60,
    Statistic: "Average",
    Threshold: 70.0,
    ActionsEnabled: false,
    AlarmDescription: "Alarm when server CPU exceeds 70%",
    Dimensions: [
      {
        Name: "InstanceId",
        Value: process.env.EC2_INSTANCE_ID, // Set the value of EC_INSTANCE_ID to
 the Id of an existing Amazon EC2 instance.
      },
    ],
    Unit: "Percent",
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Crie o cliente em um módulo separado e exporte-o.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte PutMetricAlarm na Referência da API AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmName: "Web_Server_CPU_Utilization",
  ComparisonOperator: "GreaterThanThreshold",
  EvaluationPeriods: 1,
  MetricName: "CPUUtilization",
  Namespace: "AWS/EC2",
  Period: 60,
  Statistic: "Average",
  Threshold: 70.0,
  ActionsEnabled: false,
  AlarmDescription: "Alarm when server CPU exceeds 70%",
  Dimensions: [
    {
      Name: "InstanceId",
      Value: "INSTANCE_ID",
    },
  ],
  Unit: "Percent",
};

cw.putMetricAlarm(params, function (err, data) {
```

```
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte PutMetricAlarm na Referência da API AWS SDK for JavaScript.

Kotlin

SDK para Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
suspend fun putMetricAlarm(alarmNameVal: String, instanceIdVal: String) {

    val dimensionOb = Dimension {
        name = "InstanceId"
        value = instanceIdVal
    }

    val request = PutMetricAlarmRequest {
        alarmName = alarmNameVal
        comparisonOperator = ComparisonOperator.GreaterThanThreshold
        evaluationPeriods = 1
        metricName = "CPUUtilization"
        namespace = "AWS/EC2"
        period = 60
        statistic = Statistic.fromValue("Average")
        threshold = 70.0
        actionsEnabled = false
```

```
        alarmDescription = "An Alarm created by the Kotlin SDK when server CPU
 utilization exceeds 70%"
        unit = StandardUnit.fromValue("Seconds")
        dimensions = listOf(dimensionOb)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricAlarm(request)
        println("Successfully created an alarm with name $alarmNameVal")
    }
}
```

- Para obter detalhes da API, consulte PutMetricAlarm em Referência da API AWS SDK para Kotlin.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def create_metric_alarm(
        self,
        metric_namespace,
        metric_name,
        alarm_name,
```

```
        stat_type,
        period,
        eval_periods,
        threshold,
        comparison_op,
    ):
        """
        Creates an alarm that watches a metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :param alarm_name: The name of the alarm.
        :param stat_type: The type of statistic the alarm watches.
        :param period: The period in which metric data are grouped to calculate
                       statistics.
        :param eval_periods: The number of periods that the metric must be over
the
                             alarm threshold before the alarm is set into an
alarmed
                             state.
        :param threshold: The threshold value to compare against the metric
statistic.
        :param comparison_op: The comparison operation used to compare the
threshold
                              against the metric.
        :return: The newly created alarm.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            alarm = metric.put_alarm(
                AlarmName=alarm_name,
                Statistic=stat_type,
                Period=period,
                EvaluationPeriods=eval_periods,
                Threshold=threshold,
                ComparisonOperator=comparison_op,
            )
            logger.info(
                "Added alarm %s to track metric %s.%s.",
                alarm_name,
                metric_namespace,
                metric_name,
            )
```

```
        except ClientError:
            logger.exception(
                "Couldn't add alarm %s to metric %s.%s",
                alarm_name,
                metric_namespace,
                metric_name,
            )
            raise
        else:
            return alarm
```

- Para obter detalhes da API, consulte PutMetricAlarm em Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
# Creates or updates an alarm in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm.
# @param alarm_description [String] A description about the alarm.
# @param metric_name [String] The name of the metric associated with the alarm.
# @param alarm_actions [Array] A list of Strings representing the
#   Amazon Resource Names (ARNs) to execute when the alarm transitions to the
#   ALARM state.
# @param namespace [String] The namespace for the metric to alarm on.
# @param statistic [String] The statistic for the metric.
# @param dimensions [Array] A list of dimensions for the metric, specified as
#   Aws::CloudWatch::Types::Dimension.
# @param period [Integer] The number of seconds before re-evaluating the metric.
```

```
# @param unit [String] The unit of measure for the statistic.
# @param evaluation_periods [Integer] The number of periods over which data is
#   compared to the specified threshold.
# @param theshold [Float] The value against which the specified statistic is
 compared.
# @param comparison_operator [String] The arithmetic operation to use when
#   comparing the specified statistic and threshold.
# @return [Boolean] true if the alarm was created or updated; otherwise, false.
# @example
#   exit 1 unless alarm_created_or_updated?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket',
#     'Objects exist in this bucket for more than 1 day.',
#     'NumberOfObjects',
#     ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic'],
#     'AWS/S3',
#     'Average',
#     [
#       {
#         name: 'BucketName',
#         value: 'doc-example-bucket'
#       },
#       {
#         name: 'StorageType',
#         value: 'AllStorageTypes'
#       }
#     ],
#     86_400,
#     'Count',
#     1,
#     1,
#     'GreaterThanThreshold'
#   )
def alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
  alarm_description,
  metric_name,
  alarm_actions,
  namespace,
  statistic,
  dimensions,
  period,
  unit,
```

```
  evaluation_periods,
  threshold,
  comparison_operator
)
  cloudwatch_client.put_metric_alarm(
    alarm_name: alarm_name,
    alarm_description: alarm_description,
    metric_name: metric_name,
    alarm_actions: alarm_actions,
    namespace: namespace,
    statistic: statistic,
    dimensions: dimensions,
    period: period,
    unit: unit,
    evaluation_periods: evaluation_periods,
    threshold: threshold,
    comparison_operator: comparison_operator
  )
  return true
rescue StandardError => e
  puts "Error creating alarm: #{e.message}"
  return false
end
```

- Para obter detalhes da API, consulte PutMetricAlarm na Referência da API AWS SDK for Ruby.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
    TRY.
        lo_cwt->putmetricalarm(
          iv_alarmname                = iv_alarm_name
```

```
          iv_comparisonoperator        = iv_comparison_operator
          iv_evaluationperiods         = iv_evaluation_periods
          iv_metricname                = iv_metric_name
          iv_namespace                 = iv_namespace
          iv_statistic                 = iv_statistic
          iv_threshold                 = iv_threshold
          iv_actionsenabled            = iv_actions_enabled
          iv_alarmdescription          = iv_alarm_description
          iv_unit                      = iv_unit
          iv_period                    = iv_period
          it_dimensions                = it_dimensions
        ).
        MESSAGE 'Alarm created.' TYPE 'I'.
      CATCH /aws1/cx_cwtlimitexceededfault.
        MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
    ENDTRY.
```

- Para obter detalhes sobre a API, consulte PutMetricAlarm na Referência da API do AWS SDK para SAP ABAP.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Usar **`PutMetricData`** com o AWS SDK ou a CLI

Os exemplos de código a seguir mostram como usar o `PutMetricData`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto nos seguintes exemplos de código:

- Começar a usar métricas, painéis e alarmes
- Gerencie métricas e alarmes

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
   /// <summary>
   /// Add some metric data using a call to a wrapper class.
   /// </summary>
   /// <param name="customMetricName">The metric name.</param>
   /// <param name="customMetricNamespace">The metric namespace.</param>
   /// <returns></returns>
   private static async Task<List<MetricDatum>> PutRandomMetricData(string
customMetricName,
       string customMetricNamespace)
   {
       List<MetricDatum> customData = new List<MetricDatum>();
       Random rnd = new Random();

       // Add 10 random values up to 100, starting with a timestamp 15 minutes
in the past.
       var utcNowMinus15 = DateTime.UtcNow.AddMinutes(-15);
       for (int i = 0; i < 10; i++)
       {
           var metricValue = rnd.Next(0, 100);
           customData.Add(
               new MetricDatum
               {
                   MetricName = customMetricName,
                   Value = metricValue,
                   TimestampUtc = utcNowMinus15.AddMinutes(i)
               }
           );
       }

       await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
```

```
        return customData;
    }

    /// <summary>
    /// Wrapper to add metric data to a CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricData">A data object for the metric data.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricData(string metricNamespace,
        List<MetricDatum> metricData)
    {
        var putDataResponse = await _amazonCloudWatch.PutMetricDataAsync(
            new PutMetricDataRequest()
            {
                MetricData = metricData,
                Namespace = metricNamespace,
            });

        return putDataResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obter detalhes da API, consulte PutMetricData na Referência da API AWS SDK for .NET.

C++

SDK para C++

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Inclua os arquivos necessários.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/PutMetricDataRequest.h>
```

```
#include <iostream>
```

Insira dados na métrica.

```
        Aws::CloudWatch::CloudWatchClient cw;

        Aws::CloudWatch::Model::Dimension dimension;
        dimension.SetName("UNIQUE_PAGES");
        dimension.SetValue("URLS");

        Aws::CloudWatch::Model::MetricDatum datum;
        datum.SetMetricName("PAGES_VISITED");
        datum.SetUnit(Aws::CloudWatch::Model::StandardUnit::None);
        datum.SetValue(data_point);
        datum.AddDimensions(dimension);

        Aws::CloudWatch::Model::PutMetricDataRequest request;
        request.SetNamespace("SITE/TRAFFIC");
        request.AddMetricData(datum);

        auto outcome = cw.PutMetricData(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to put sample metric data:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully put sample metric data" << std::endl;
        }
```

- Para obter detalhes da API, consulte PutMetricData na Referência da API AWS SDK for C++.

CLI

AWS CLI

Como publicar uma métrica personalizada no Amazon CloudWatch

O seguinte exemplo usa o comando `put-metric-data` para publicar uma métrica personalizada no Amazon CloudWatch:

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://
metric.json
```

Os valores da própria métrica estão armazenados no arquivo em JSON, `metric.json`.

Veja o conteúdo desse arquivo:

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

Para obter mais informações, consulte Publicar métricas personalizadas no Guia do desenvolvedor do Amazon CloudWatch.

Como especificar diversas dimensões

O exemplo a seguir ilustra como especificar diversas dimensões. Cada dimensão é especificada como um par de nome/valor. As diversas dimensões são separadas por uma vírgula.

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace
 MyNameSpace --unit Bytes --value 231434333 --dimensions
 InstanceID=1-23456789,InstanceType=m1.small
```

- Para obter detalhes da API, consulte PutMetricData na Referência de comandos da AWS CLI.

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
   public static void addMetricDataForAlarm(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           // Set an Instant object.
           String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
           Instant instant = Instant.parse(time);

           MetricDatum datum = MetricDatum.builder()
                   .metricName(customMetricName)
                   .unit(StandardUnit.NONE)
                   .value(1001.00)
                   .timestamp(instant)
                   .build();

           MetricDatum datum2 = MetricDatum.builder()
                   .metricName(customMetricName)
                   .unit(StandardUnit.NONE)
                   .value(1002.00)
                   .timestamp(instant)
                   .build();
```

```
            List<MetricDatum> metricDataList = new ArrayList<>();
            metricDataList.add(datum);
            metricDataList.add(datum2);

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace(customMetricNamespace)
                    .metricData(metricDataList)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric " +
customMetricName);

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte PutMetricData na Referência da API AWS SDK for Java 2.x.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Importe o SDK e os módulos do cliente e chame a API.

```
import { PutMetricDataCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  // See https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/
API_PutMetricData.html#API_PutMetricData_RequestParameters
```

```
  // and https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
publishingMetrics.html
  // for more information about the parameters in this command.
  const command = new PutMetricDataCommand({
    MetricData: [
      {
        MetricName: "PAGES_VISITED",
        Dimensions: [
          {
            Name: "UNIQUE_PAGES",
            Value: "URLS",
          },
        ],
        Unit: "None",
        Value: 1.0,
      },
    ],
    Namespace: "SITE/TRAFFIC",
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Crie o cliente em um módulo separado e exporte-o.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte PutMetricData na Referência da API AWS SDK for JavaScript.

SDK para JavaScript (v2)

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

// Create parameters JSON for putMetricData
var params = {
  MetricData: [
    {
      MetricName: "PAGES_VISITED",
      Dimensions: [
        {
          Name: "UNIQUE_PAGES",
          Value: "URLS",
        },
      ],
      Unit: "None",
      Value: 1.0,
    },
  ],
  Namespace: "SITE/TRAFFIC",
};

cw.putMetricData(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", JSON.stringify(data));
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte PutMetricData na Referência da API AWS SDK for JavaScript.

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
suspend fun addMetricDataForAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1001.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val datum2 = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1002.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }
```

```
    val metricDataList = ArrayList<MetricDatum>()
    metricDataList.add(datum)
    metricDataList.add(datum2)

    val request = PutMetricDataRequest {
        namespace = customMetricNamespace
        metricData = metricDataList
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric $customMetricName")
    }
}
```

- Para obter detalhes da API, consulte PutMetricData, na Referência da API AWS SDK para Kotlin.

PowerShell

Tools for PowerShell

Exemplo 1: cria um novo objeto MetricDatum e o grava nas métricas do CloudWatch da Amazon Web Services.

```
### Create a MetricDatum .NET object
$Metric = New-Object -TypeName Amazon.CloudWatch.Model.MetricDatum
$Metric.Timestamp = [DateTime]::UtcNow
$Metric.MetricName = 'CPU'
$Metric.Value = 50

### Write the metric data to the CloudWatch service
Write-CWMetricData -Namespace instance1 -MetricData $Metric
```

- Para obter detalhes da API, consulte PutMetricData na Referência de Cmdlet do AWS Tools for PowerShell.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data(self, namespace, name, value, unit):
        """
        Sends a single data value to CloudWatch for a metric. This metric is
 given
        a timestamp of the current UTC time.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param value: The value of the metric.
        :param unit: The unit of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{"MetricName": name, "Value": value, "Unit": unit}],
            )
            logger.info("Put data for metric %s.%s", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data for metric %s.%s", namespace,
 name)
            raise
```

Defina um conjunto de dados em uma métrica do CloudWatch.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data_set(self, namespace, name, timestamp, unit, data_set):
        """
        Sends a set of data to CloudWatch for a metric. All of the data in the
 set
        have the same timestamp and unit.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param timestamp: The UTC timestamp for the metric.
        :param unit: The unit of the metric.
        :param data_set: The set of data to send. This set is a dictionary that
                         contains a list of values and a list of corresponding
 counts.
                         The value and count lists must be the same length.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[
                    {
                        "MetricName": name,
                        "Timestamp": timestamp,
                        "Values": data_set["values"],
                        "Counts": data_set["counts"],
                        "Unit": unit,
                    }
                ],
```

```
            )
            logger.info("Put data set for metric %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data set for metric %s.%s.",
namespace, name)
            raise
```

- Para obter detalhes da API, consulte PutMetricData na Referência da API AWS SDK para Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
require "aws-sdk-cloudwatch"

# Adds a datapoint to a metric in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric to add the
#   datapoint to.
# @param metric_name [String] The name of the metric to add the datapoint to.
# @param dimension_name [String] The name of the dimension to add the
#   datapoint to.
# @param dimension_value [String] The value of the dimension to add the
#   datapoint to.
# @param metric_value [Float] The value of the datapoint.
# @param metric_unit [String] The unit of measurement for the datapoint.
# @return [Boolean]
# @example
#   exit 1 unless datapoint_added_to_metric?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
```

```
#     'SITE/TRAFFIC',
#     'UniqueVisitors',
#     'SiteName',
#     'example.com',
#     5_885.0,
#     'Count'
#   )
def datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  metric_name,
  dimension_name,
  dimension_value,
  metric_value,
  metric_unit
)
  cloudwatch_client.put_metric_data(
    namespace: metric_namespace,
    metric_data: [
      {
        metric_name: metric_name,
        dimensions: [
          {
            name: dimension_name,
            value: dimension_value
          }
        ],
        value: metric_value,
        unit: metric_unit
      }
    ]
  )
  puts "Added data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}'."
  return true
rescue StandardError => e
  puts "Error adding data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}': #{e.message}"
  return false
end
```

- Para obter detalhes da API, consulte PutMetricData na Referência da API AWS SDK for Ruby.

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cenários para o CloudWatch usando AWS SDKs

Os exemplos de código a seguir mostram como implementar cenários comuns no CloudWatch com os SDKs da AWS. Esses cenários mostram como realizar tarefas específicas chamando várias funções no CloudWatch. Cada exemplo inclui um link para o GitHub, onde você pode encontrar instruções sobre como configurar e executar o código.

Exemplos

- Começar a usar alarmes do CloudWatch usando um AWS SDK
- Começar a usar métricas, painéis e alarmes do CloudWatch usando um AWS SDK
- Gerenciar métricas e alarmes do CloudWatch usando um AWS SDK

Começar a usar alarmes do CloudWatch usando um AWS SDK

O exemplo de código a seguir mostra como:

- Criar um alarme.
- Desabilitar ações de alarme.
- Descrever um alarme.
- Exclua um alarme.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
DATA lt_alarmnames TYPE /aws1/cl_cwtalarmnames_w=>tt_alarmnames.
DATA lo_alarmname TYPE REF TO /aws1/cl_cwtalarmnames_w.

"Create an alarm"
TRY.
    lo_cwt->putmetricalarm(
      iv_alarmname                = iv_alarm_name
      iv_comparisonoperator       = iv_comparison_operator
      iv_evaluationperiods        = iv_evaluation_periods
      iv_metricname               = iv_metric_name
      iv_namespace                = iv_namespace
      iv_statistic                = iv_statistic
      iv_threshold                = iv_threshold
      iv_actionsenabled           = iv_actions_enabled
      iv_alarmdescription         = iv_alarm_description
      iv_unit                     = iv_unit
      iv_period                   = iv_period
      it_dimensions               = it_dimensions
    ).
    MESSAGE 'Alarm created' TYPE 'I'.
  CATCH /aws1/cx_cwtlimitexceededfault.
    MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
ENDTRY.

"Create an ABAP internal table for the created alarm."
CREATE OBJECT lo_alarmname EXPORTING iv_value = iv_alarm_name.
INSERT lo_alarmname INTO TABLE lt_alarmnames.

"Disable alarm actions."
TRY.
    lo_cwt->disablealarmactions(
```

```
          it_alarmnames               = lt_alarmnames
        ).
        MESSAGE 'Alarm actions disabled' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_disablealarm_exception).
        DATA(lv_disablealarm_error) = |"{ lo_disablealarm_exception-
>av_err_code }" - { lo_disablealarm_exception->av_err_msg }|.
        MESSAGE lv_disablealarm_error TYPE 'E'.
    ENDTRY.

    "Describe alarm using the same ABAP internal table."
    TRY.
        oo_result = lo_cwt->describealarms(                      " oo_result is
 returned for testing purpose "
          it_alarmnames               = lt_alarmnames
        ).
        MESSAGE 'Alarms retrieved' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_describealarms_exception).
        DATA(lv_describealarms_error) = |"{ lo_describealarms_exception-
>av_err_code }" - { lo_describealarms_exception->av_err_msg }|.
        MESSAGE lv_describealarms_error TYPE 'E'.
    ENDTRY.

    "Delete alarm."
    TRY.
        lo_cwt->deletealarms(
          it_alarmnames = lt_alarmnames
        ).
        MESSAGE 'Alarms deleted' TYPE 'I'.
      CATCH /aws1/cx_cwtresourcenotfound .
        MESSAGE 'Resource being access is not found.' TYPE 'E'.
    ENDTRY.
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para SAP ABAP.
  - DeleteAlarms
  - DescribeAlarms
  - DisableAlarmActions
  - PutMetricAlarm

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Começar a usar métricas, painéis e alarmes do CloudWatch usando um AWS SDK

Os exemplos de código a seguir mostram como:

- Listar namespaces e métricas do CloudWatch.
- Obter estatísticas para uma métrica e para faturamento estimado.
- Criar e atualizar um painel.
- Criar e adicionar dados a uma métrica.
- Criar e acionar um alarme e, em seguida, visualizar o histórico de alarmes.
- Criar um detector de anomalias.
- Obtenha uma imagem de métrica e, em seguida, limpe os recursos.

.NET

AWS SDK for .NET

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

Execute um cenário interativo em um prompt de comando.

```
public class CloudWatchScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    To enable billing metrics and statistics for this example, make sure billing
 alerts are enabled for your account:
```

```
    https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics

    This .NET example performs the following tasks:
        1. List and select a CloudWatch namespace.
        2. List and select a CloudWatch metric.
        3. Get statistics for a CloudWatch metric.
        4. Get estimated billing statistics for the last week.
        5. Create a new CloudWatch dashboard with two metrics.
        6. List current CloudWatch dashboards.
        7. Create a CloudWatch custom metric and add metric data.
        8. Add the custom metric to the dashboard.
        9. Create a CloudWatch alarm for the custom metric.
       10. Describe current CloudWatch alarms.
       11. Get recent data for the custom metric.
       12. Add data to the custom metric to trigger the alarm.
       13. Wait for an alarm state.
       14. Get history for the CloudWatch alarm.
       15. Add an anomaly detector.
       16. Describe current anomaly detectors.
       17. Get and display a metric image.
       18. Clean up resources.
    */

    private static ILogger logger = null!;
    private static CloudWatchWrapper _cloudWatchWrapper = null!;
    private static IConfiguration _configuration = null!;
    private static readonly List<string> _statTypes = new List<string>
{ "SampleCount", "Average", "Sum", "Minimum", "Maximum" };
    private static SingleMetricAnomalyDetector? anomalyDetector = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for the Amazon service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonCloudWatch>()
            .AddTransient<CloudWatchWrapper>()
```

```
        )
        .Build();

        _configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally, load local settings.
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<CloudWatchScenario>();

        _cloudWatchWrapper =
host.Services.GetRequiredService<CloudWatchWrapper>();

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the Amazon CloudWatch example scenario.");
        Console.WriteLine(new string('-', 80));

        try
        {
            var selectedNamespace = await SelectNamespace();
            var selectedMetric = await SelectMetric(selectedNamespace);
            await GetAndDisplayMetricStatistics(selectedNamespace,
selectedMetric);
            await GetAndDisplayEstimatedBilling();
            await CreateDashboardWithMetrics();
            await ListDashboards();
            await CreateNewCustomMetric();
            await AddMetricToDashboard();
            await CreateMetricAlarm();
            await DescribeAlarms();
            await GetCustomMetricData();
            await AddMetricDataForAlarm();
            await CheckForMetricAlarm();
            await GetAlarmHistory();
            anomalyDetector = await AddAnomalyDetector();
            await DescribeAnomalyDetectors();
            await GetAndOpenMetricImage();
            await CleanupResources();
        }
        catch (Exception ex)
        {
```

```
            logger.LogError(ex, "There was a problem executing the scenario.");
            await CleanupResources();
        }

    }

    /// <summary>
    /// Select a namespace.
    /// </summary>
    /// <returns>The selected namespace.</returns>
    private static async Task<string> SelectNamespace()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"1. Select a CloudWatch Namespace from a list of
Namespaces.");
        var metrics = await _cloudWatchWrapper.ListMetrics();
        // Get a distinct list of namespaces.
        var namespaces = metrics.Select(m => m.Namespace).Distinct().ToList();
        for (int i = 0; i < namespaces.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {namespaces[i]}");
        }

        var namespaceChoiceNumber = 0;
        while (namespaceChoiceNumber < 1 || namespaceChoiceNumber >
namespaces.Count)
        {
            Console.WriteLine(
                "Select a namespace by entering a number from the preceding
list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out namespaceChoiceNumber);
        }

        var selectedNamespace = namespaces[namespaceChoiceNumber - 1];

        Console.WriteLine(new string('-', 80));

        return selectedNamespace;
    }

    /// <summary>
    /// Select a metric from a namespace.
    /// </summary>
```

```
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <returns>The metric name.</returns>
    private static async Task<Metric> SelectMetric(string metricNamespace)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"2. Select a CloudWatch metric from a namespace.");

        var namespaceMetrics = await
_cloudWatchWrapper.ListMetrics(metricNamespace);

        for (int i = 0; i < namespaceMetrics.Count && i < 15; i++)
        {
            var dimensionsWithValues = namespaceMetrics[i].Dimensions
                .Where(d => !string.Equals("None", d.Value));
            Console.WriteLine($"\t{i + 1}. {namespaceMetrics[i].MetricName} " +
                              $"{string.Join(", :", dimensionsWithValues.Select(d
=> d.Value))}");
        }

        var metricChoiceNumber = 0;
        while (metricChoiceNumber < 1 || metricChoiceNumber >
namespaceMetrics.Count)
        {
            Console.WriteLine(
                "Select a metric by entering a number from the preceding list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out metricChoiceNumber);
        }

        var selectedMetric = namespaceMetrics[metricChoiceNumber - 1];

        Console.WriteLine(new string('-', 80));

        return selectedMetric;
    }

    /// <summary>
    /// Get and display metric statistics for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task GetAndDisplayMetricStatistics(string
metricNamespace, Metric metric)
```

```
   {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"3. Get CloudWatch metric statistics for the last
day.");

        for (int i = 0; i < _statTypes.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {_statTypes[i]}");
        }

        var statisticChoiceNumber = 0;
        while (statisticChoiceNumber < 1 || statisticChoiceNumber >
_statTypes.Count)
        {
            Console.WriteLine(
                "Select a metric statistic by entering a number from the
preceding list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out statisticChoiceNumber);
        }

        var selectedStatistic = _statTypes[statisticChoiceNumber - 1];
        var statisticsList = new List<string> { selectedStatistic };

        var metricStatistics = await
_cloudWatchWrapper.GetMetricStatistics(metricNamespace, metric.MetricName,
statisticsList, metric.Dimensions, 1, 60);

        if (!metricStatistics.Any())
        {
            Console.WriteLine($"No {selectedStatistic} statistics found for
{metric} in namespace {metricNamespace}.");
        }

        metricStatistics = metricStatistics.OrderBy(s => s.Timestamp).ToList();
        for (int i = 0; i < metricStatistics.Count && i < 10; i++)
        {
            var metricStat = metricStatistics[i];
            var statValue =
metricStat.GetType().GetProperty(selectedStatistic)!.GetValue(metricStat, null);
            Console.WriteLine($"\t{i + 1}. Timestamp
{metricStatistics[i].Timestamp:G} {selectedStatistic}: {statValue}");
        }
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get and display estimated billing statistics.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task GetAndDisplayEstimatedBilling()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"4. Get CloudWatch estimated billing for the last
week.");

        var billingStatistics = await SetupBillingStatistics();

        for (int i = 0; i < billingStatistics.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. Timestamp
{billingStatistics[i].Timestamp:G} : {billingStatistics[i].Maximum}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get billing statistics using a call to a wrapper class.
    /// </summary>
    /// <returns>A collection of billing statistics.</returns>
    private static async Task<List<Datapoint>> SetupBillingStatistics()
    {
        // Make a request for EstimatedCharges with a period of one day for the
past seven days.
        var billingStatistics = await _cloudWatchWrapper.GetMetricStatistics(
            "AWS/Billing",
            "EstimatedCharges",
            new List<string>() { "Maximum" },
            new List<Dimension>() { new Dimension { Name = "Currency", Value =
"USD" } },
            7,
            86400);

        billingStatistics = billingStatistics.OrderBy(n => n.Timestamp).ToList();
```

```
        return billingStatistics;
    }

    /// <summary>
    /// Create a dashboard with metrics.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task CreateDashboardWithMetrics()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"5. Create a new CloudWatch dashboard with metrics.");
        var dashboardName = _configuration["dashboardName"];
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);
        var newDashboardString = JsonSerializer.Serialize(
            newDashboard,
            new JsonSerializerOptions
            {
                DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull
            });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
newDashboardString);

        Console.WriteLine(validationMessages.Any() ? $"\tValidation messages:" :
null);
        for (int i = 0; i < validationMessages.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {validationMessages[i].Message}");
        }
        Console.WriteLine($"\tDashboard {dashboardName} was created.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List dashboards.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task ListDashboards()
    {
        Console.WriteLine(new string('-', 80));
```

```
        Console.WriteLine($"6. List the CloudWatch dashboards in the current
account.");

        var dashboards = await _cloudWatchWrapper.ListDashboards();

        for (int i = 0; i < dashboards.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {dashboards[i].DashboardName}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create and add data for a new custom metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateNewCustomMetric()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"7. Create and add data for a new custom metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var customData = await PutRandomMetricData(customMetricName,
customMetricNamespace);

        var valuesString = string.Join(',', customData.Select(d => d.Value));
        Console.WriteLine($"\tAdded metric values for for metric
{customMetricName}: \n\t{valuesString}");

        Console.WriteLine(new string('-', 80));
    }


    /// <summary>
    /// Add some metric data using a call to a wrapper class.
    /// </summary>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <returns></returns>
    private static async Task<List<MetricDatum>> PutRandomMetricData(string
customMetricName,
```

```
        string customMetricNamespace)
    {
        List<MetricDatum> customData = new List<MetricDatum>();
        Random rnd = new Random();

        // Add 10 random values up to 100, starting with a timestamp 15 minutes
in the past.
        var utcNowMinus15 = DateTime.UtcNow.AddMinutes(-15);
        for (int i = 0; i < 10; i++)
        {
            var metricValue = rnd.Next(0, 100);
            customData.Add(
                new MetricDatum
                {
                    MetricName = customMetricName,
                    Value = metricValue,
                    TimestampUtc = utcNowMinus15.AddMinutes(i)
                }
            );
        }

        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
        return customData;
    }

    /// <summary>
    /// Add the custom metric to the dashboard.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task AddMetricToDashboard()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"8. Add the new custom metric to the dashboard.");

        var dashboardName = _configuration["dashboardName"];

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var validationMessages = await SetupDashboard(customMetricNamespace,
customMetricName, dashboardName);
```

```
        Console.WriteLine(validationMessages.Any() ? $"\tValidation messages:" :
null);
        for (int i = 0; i < validationMessages.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {validationMessages[i].Message}");
        }
        Console.WriteLine($"\tDashboard {dashboardName} updated with metric
{customMetricName}.");
        Console.WriteLine(new string('-', 80));
    }


    /// <summary>
    /// Set up a dashboard using a call to the wrapper class.
    /// </summary>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A list of validation messages.</returns>
    private static async Task<List<DashboardValidationMessage>> SetupDashboard(
        string customMetricNamespace, string customMetricName, string
dashboardName)
    {
        // Get the dashboard model from configuration.
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);

        // Add a new metric to the dashboard.
        newDashboard.Widgets.Add(new Widget
        {
            Height = 8,
            Width = 8,
            Y = 8,
            X = 0,
            Type = "metric",
            Properties = new Properties
            {
                Metrics = new List<List<object>>
                    { new() { customMetricNamespace, customMetricName } },
                View = "timeSeries",
                Region = "us-east-1",
                Stat = "Sum",
                Period = 86400,
                YAxis = new YAxis { Left = new Left { Min = 0, Max = 100 } },
```

```
                Title = "Custom Metric Widget",
                LiveData = true,
                Sparkline = true,
                Trend = true,
                Stacked = false,
                SetPeriodToTimeRange = false
            }
        });

        var newDashboardString = JsonSerializer.Serialize(newDashboard,
            new JsonSerializerOptions
            { DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
newDashboardString);

        return validationMessages;
    }

    /// <summary>
    /// Create a CloudWatch alarm for the new metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateMetricAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"9. Create a CloudWatch alarm for the new metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var alarmName = _configuration["exampleAlarmName"];
        var accountId = _configuration["accountId"];
        var region = _configuration["region"];
        var emailTopic = _configuration["emailTopic"];
        var alarmActions = new List<string>();

        if (GetYesNoResponse(
                $"\tAdd an email action for topic {emailTopic} to alarm
{alarmName}? (y/n)"))
        {
            _cloudWatchWrapper.AddEmailAlarmAction(accountId, region, emailTopic,
alarmActions);
        }
```

```
        await _cloudWatchWrapper.PutMetricEmailAlarm(
            "Example metric alarm",
            alarmName,
            ComparisonOperator.GreaterThanOrEqualToThreshold,
            customMetricName,
            customMetricNamespace,
            100,
            alarmActions);

        Console.WriteLine($"\tAlarm {alarmName} added for metric
{customMetricName}.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Describe Alarms.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task DescribeAlarms()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"10. Describe CloudWatch alarms in the current
account.");

        var alarms = await _cloudWatchWrapper.DescribeAlarms();
        alarms = alarms.OrderByDescending(a => a.StateUpdatedTimestamp).ToList();

        for (int i = 0; i < alarms.Count && i < 10; i++)
        {
            var alarm = alarms[i];
            Console.WriteLine($"\t{i + 1}. {alarm.AlarmName}");
            Console.WriteLine($"\tState: {alarm.StateValue} for
{alarm.MetricName} {alarm.ComparisonOperator} {alarm.Threshold}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get the recent data for the metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetCustomMetricData()
```

```
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"11. Get current data for new custom metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var accountId = _configuration["accountId"];

        var query = new List<MetricDataQuery>
        {
            new MetricDataQuery
            {
                AccountId = accountId,
                Id = "m1",
                Label = "Custom Metric Data",
                MetricStat = new MetricStat
                {
                    Metric = new Metric
                    {
                        MetricName = customMetricName,
                        Namespace = customMetricNamespace,
                    },
                    Period = 1,
                    Stat = "Maximum"
                }
            }
        };

        var metricData = await _cloudWatchWrapper.GetMetricData(
            20,
            true,
            DateTime.UtcNow.AddMinutes(1),
            20,
            query);

        for (int i = 0; i < metricData.Count; i++)
        {
            for (int j = 0; j < metricData[i].Values.Count; j++)
            {
                Console.WriteLine(
                    $"\tTimestamp {metricData[i].Timestamps[j]:G} Value:
{metricData[i].Values[j]}");
            }
        }
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add metric data to trigger an alarm.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task AddMetricDataForAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"12. Add metric data to the custom metric to trigger
an alarm.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var nowUtc = DateTime.UtcNow;
        List<MetricDatum> customData = new List<MetricDatum>
        {
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc.AddMinutes(-2)
            },
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc.AddMinutes(-1)
            },
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc
            }
        };
        var valuesString = string.Join(',', customData.Select(d => d.Value));
        Console.WriteLine($"\tAdded metric values for for metric
{customMetricName}: \n\t{valuesString}");
        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Check for a metric alarm using the DescribeAlarmsForMetric action.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CheckForMetricAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"13. Checking for an alarm state.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var hasAlarm = false;
        var retries = 10;
        while (!hasAlarm && retries > 0)
        {
            var alarms = await
_cloudWatchWrapper.DescribeAlarmsForMetric(customMetricNamespace,
customMetricName);
            hasAlarm = alarms.Any(a => a.StateValue == StateValue.ALARM);
            retries--;
            Thread.Sleep(20000);
        }

        Console.WriteLine(hasAlarm
            ? $"\tAlarm state found for {customMetricName}."
            : $"\tNo Alarm state found for {customMetricName} after 10
retries.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get history for an alarm.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetAlarmHistory()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"14. Get alarm history.");

        var exampleAlarmName = _configuration["exampleAlarmName"];
```

```
        var alarmHistory = await
_cloudWatchWrapper.DescribeAlarmHistory(exampleAlarmName, 2);

        for (int i = 0; i < alarmHistory.Count; i++)
        {
            var history = alarmHistory[i];
            Console.WriteLine($"\t{i + 1}. {history.HistorySummary}, time
{history.Timestamp:g}");
        }
        if (!alarmHistory.Any())
        {
            Console.WriteLine($"\tNo alarm history data found for
{exampleAlarmName}.");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add an anomaly detector.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task<SingleMetricAnomalyDetector> AddAnomalyDetector()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"15. Add an anomaly detector.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var detector = new SingleMetricAnomalyDetector
        {
            MetricName = customMetricName,
            Namespace = customMetricNamespace,
            Stat = "Maximum"
        };
        await _cloudWatchWrapper.PutAnomalyDetector(detector);
        Console.WriteLine($"\tAdded anomaly detector for metric
{customMetricName}.");

        Console.WriteLine(new string('-', 80));
        return detector;
    }
```

```
    /// <summary>
    /// Describe anomaly detectors.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task DescribeAnomalyDetectors()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"16. Describe anomaly detectors in the current
account.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var detectors = await
_cloudWatchWrapper.DescribeAnomalyDetectors(customMetricNamespace,
customMetricName);

        for (int i = 0; i < detectors.Count; i++)
        {
            var detector = detectors[i];
            Console.WriteLine($"\t{i + 1}.
{detector.SingleMetricAnomalyDetector.MetricName}, state
{detector.StateValue}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Fetch and open a metrics image for a CloudWatch metric and namespace.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetAndOpenMetricImage()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("17. Get a metric image from CloudWatch.");

        Console.WriteLine($"\tGetting Image data for custom metric.");
        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
```

```
        var memoryStream = await
_cloudWatchWrapper.GetTimeSeriesMetricImage(customMetricNamespace,
customMetricName, "Maximum", 10);
        var file = _cloudWatchWrapper.SaveMetricImage(memoryStream,
"MetricImages");

        ProcessStartInfo info = new ProcessStartInfo();

        Console.WriteLine($"\tFile saved as {Path.GetFileName(file)}.");
        Console.WriteLine($"\tPress enter to open the image.");
        Console.ReadLine();
        info.FileName = Path.Combine("ms-photos://", file);
        info.UseShellExecute = true;
        info.CreateNoWindow = true;
        info.Verb = string.Empty;

        Process.Start(info);

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Clean up created resources.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task CleanupResources()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"18. Clean up resources.");

        var dashboardName = _configuration["dashboardName"];
        if (GetYesNoResponse($"\tDelete dashboard {dashboardName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting dashboard.");
            var dashboardList = new List<string> { dashboardName };
            await _cloudWatchWrapper.DeleteDashboards(dashboardList);
        }

        var alarmName = _configuration["exampleAlarmName"];
        if (GetYesNoResponse($"\tDelete alarm {alarmName}? (y/n)"))
        {
            Console.WriteLine($"\tCleaning up alarms.");
```

```
            var alarms = new List<string> { alarmName };
            await _cloudWatchWrapper.DeleteAlarms(alarms);
        }

        if (GetYesNoResponse($"\tDelete anomaly detector? (y/n)") &&
 anomalyDetector != null)
        {
            Console.WriteLine($"\tCleaning up anomaly detector.");

            await _cloudWatchWrapper.DeleteAnomalyDetector(
                anomalyDetector);
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get a yes or no response from the user.
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <returns>True if the user responds with a yes.</returns>
    private static bool GetYesNoResponse(string question)
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null &&
                       ynResponse.Equals("y",
                           StringComparison.InvariantCultureIgnoreCase);
        return response;
    }
}
```

Os métodos de wrapper usados pelo cenário para as ações do CloudWatch.

```
/// <summary>
/// Wrapper class for Amazon CloudWatch methods.
/// </summary>
public class CloudWatchWrapper
{
    private readonly IAmazonCloudWatch _amazonCloudWatch;
    private readonly ILogger<CloudWatchWrapper> _logger;
```

```
    /// <summary>
    /// Constructor for the CloudWatch wrapper.
    /// </summary>
    /// <param name="amazonCloudWatch">The injected CloudWatch client.</param>
    /// <param name="logger">The injected logger for the wrapper.</param>
    public CloudWatchWrapper(IAmazonCloudWatch amazonCloudWatch,
ILogger<CloudWatchWrapper> logger)

    {
        _logger = logger;
        _amazonCloudWatch = amazonCloudWatch;
    }

    /// <summary>
    /// List metrics available, optionally within a namespace.
    /// </summary>
    /// <param name="metricNamespace">Optional CloudWatch namespace to use when
listing metrics.</param>
    /// <param name="filter">Optional dimension filter.</param>
    /// <param name="metricName">Optional metric name filter.</param>
    /// <returns>The list of metrics.</returns>
    public async Task<List<Metric>> ListMetrics(string? metricNamespace = null,
DimensionFilter? filter = null, string? metricName = null)
    {
        var results = new List<Metric>();
        var paginateMetrics = _amazonCloudWatch.Paginators.ListMetrics(
            new ListMetricsRequest
            {
                Namespace = metricNamespace,
                Dimensions = filter != null ? new List<DimensionFilter>
{ filter } : null,
                MetricName = metricName
            });
        // Get the entire list using the paginator.
        await foreach (var metric in paginateMetrics.Metrics)
        {
            results.Add(metric);
        }

        return results;
    }

    /// <summary>
```

```
    /// Wrapper to get statistics for a specific CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <param name="statistics">The list of statistics to include.</param>
    /// <param name="dimensions">The list of dimensions to include.</param>
    /// <param name="days">The number of days in the past to include.</param>
    /// <param name="period">The period for the data.</param>
    /// <returns>A list of DataPoint objects for the statistics.</returns>
    public async Task<List<Datapoint>> GetMetricStatistics(string
 metricNamespace,
        string metricName, List<string> statistics, List<Dimension> dimensions,
 int days, int period)
    {
        var metricStatistics = await _amazonCloudWatch.GetMetricStatisticsAsync(
            new GetMetricStatisticsRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName,
                Dimensions = dimensions,
                Statistics = statistics,
                StartTimeUtc = DateTime.UtcNow.AddDays(-days),
                EndTimeUtc = DateTime.UtcNow,
                Period = period
            });

        return metricStatistics.Datapoints;
    }

    /// <summary>
    /// Wrapper to create or add to a dashboard with metrics.
    /// </summary>
    /// <param name="dashboardName">The name for the dashboard.</param>
    /// <param name="dashboardBody">The metric data in JSON for the dashboard.</
param>
    /// <returns>A list of validation messages for the dashboard.</returns>
    public async Task<List<DashboardValidationMessage>> PutDashboard(string
 dashboardName,
        string dashboardBody)
    {
        // Updating a dashboard replaces all contents.
        // Best practice is to include a text widget indicating this dashboard
 was created programmatically.
        var dashboardResponse = await _amazonCloudWatch.PutDashboardAsync(
```

```
            new PutDashboardRequest()
            {
                DashboardName = dashboardName,
                DashboardBody = dashboardBody
            });

        return dashboardResponse.DashboardValidationMessages;
    }


    /// <summary>
    /// Get information on a dashboard.
    /// </summary>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A JSON object with dashboard information.</returns>
    public async Task<string> GetDashboard(string dashboardName)
    {
        var dashboardResponse = await _amazonCloudWatch.GetDashboardAsync(
            new GetDashboardRequest()
            {
                DashboardName = dashboardName
            });

        return dashboardResponse.DashboardBody;
    }


    /// <summary>
    /// Get a list of dashboards.
    /// </summary>
    /// <returns>A list of DashboardEntry objects.</returns>
    public async Task<List<DashboardEntry>> ListDashboards()
    {
        var results = new List<DashboardEntry>();
        var paginateDashboards = _amazonCloudWatch.Paginators.ListDashboards(
            new ListDashboardsRequest());
        // Get the entire list using the paginator.
        await foreach (var data in paginateDashboards.DashboardEntries)
        {
            results.Add(data);
        }

        return results;
    }
```

```
    /// <summary>
    /// Wrapper to add metric data to a CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricData">A data object for the metric data.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricData(string metricNamespace,
        List<MetricDatum> metricData)
    {
        var putDataResponse = await _amazonCloudWatch.PutMetricDataAsync(
            new PutMetricDataRequest()
            {
                MetricData = metricData,
                Namespace = metricNamespace,
            });

        return putDataResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Get an image for a metric graphed over time.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metric">The name of the metric.</param>
    /// <param name="stat">The name of the stat to chart.</param>
    /// <param name="period">The period to use for the chart.</param>
    /// <returns>A memory stream for the chart image.</returns>
    public async Task<MemoryStream> GetTimeSeriesMetricImage(string
metricNamespace, string metric, string stat, int period)
    {
        var metricImageWidget = new
        {
            title = "Example Metric Graph",
            view = "timeSeries",
            stacked = false,
            period = period,
            width = 1400,
            height = 600,
            metrics = new List<List<object>>
                { new() { metricNamespace, metric, new { stat } } }
        };
```

```
        var metricImageWidgetString =
 JsonSerializer.Serialize(metricImageWidget);
        var imageResponse = await _amazonCloudWatch.GetMetricWidgetImageAsync(
            new GetMetricWidgetImageRequest()
            {
                MetricWidget = metricImageWidgetString
            });

        return imageResponse.MetricWidgetImage;
    }

    /// <summary>
    /// Save a metric image to a file.
    /// </summary>
    /// <param name="memoryStream">The MemoryStream for the metric image.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The path to the file.</returns>
    public string SaveMetricImage(MemoryStream memoryStream, string metricName)
    {
        var metricFileName = $"{metricName}_{DateTime.Now.Ticks}.png";
        using var sr = new StreamReader(memoryStream);
        // Writes the memory stream to a file.
        File.WriteAllBytes(metricFileName, memoryStream.ToArray());
        var filePath = Path.Join(AppDomain.CurrentDomain.BaseDirectory,
            metricFileName);
        return filePath;
    }

    /// <summary>
    /// Get data for CloudWatch metrics.
    /// </summary>
    /// <param name="minutesOfData">The number of minutes of data to include.</
param>
    /// <param name="useDescendingTime">True to return the data descending by
 time.</param>
    /// <param name="endDateUtc">The end date for the data, in UTC.</param>
    /// <param name="maxDataPoints">The maximum data points to include.</param>
    /// <param name="dataQueries">Optional data queries to include.</param>
    /// <returns>A list of the requested metric data.</returns>
    public async Task<List<MetricDataResult>> GetMetricData(int minutesOfData,
 bool useDescendingTime, DateTime? endDateUtc = null,
        int maxDataPoints = 0, List<MetricDataQuery>? dataQueries = null)
    {
        var metricData = new List<MetricDataResult>();
```

```
        // If no end time is provided, use the current time for the end time.
        endDateUtc ??= DateTime.UtcNow;
        var timeZoneOffset =
 TimeZoneInfo.Local.GetUtcOffset(endDateUtc.Value.ToLocalTime());
        var startTimeUtc = endDateUtc.Value.AddMinutes(-minutesOfData);
        // The timezone string should be in the format +0000, so use the timezone
 offset to format it correctly.
        var timeZoneString = $"{timeZoneOffset.Hours:D2}
{timeZoneOffset.Minutes:D2}";
        var paginatedMetricData = _amazonCloudWatch.Paginators.GetMetricData(
            new GetMetricDataRequest()
            {
                StartTimeUtc = startTimeUtc,
                EndTimeUtc = endDateUtc.Value,
                LabelOptions = new LabelOptions { Timezone = timeZoneString },
                ScanBy = useDescendingTime ? ScanBy.TimestampDescending :
 ScanBy.TimestampAscending,
                MaxDatapoints = maxDataPoints,
                MetricDataQueries = dataQueries,
            });

        await foreach (var data in paginatedMetricData.MetricDataResults)
        {
            metricData.Add(data);
        }
        return metricData;
    }

    /// <summary>
    /// Add a metric alarm to send an email when the metric passes a threshold.
    /// </summary>
    /// <param name="alarmDescription">A description of the alarm.</param>
    /// <param name="alarmName">The name for the alarm.</param>
    /// <param name="comparison">The type of comparison to use.</param>
    /// <param name="metricName">The name of the metric for the alarm.</param>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="threshold">The threshold value for the alarm.</param>
    /// <param name="alarmActions">Optional actions to execute when in an alarm
 state.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricEmailAlarm(string alarmDescription, string
 alarmName, ComparisonOperator comparison,
        string metricName, string metricNamespace, double threshold, List<string>
 alarmActions = null!)
```

```
    {
        try
        {
            var putEmailAlarmResponse = await
_amazonCloudWatch.PutMetricAlarmAsync(
                new PutMetricAlarmRequest()
                {
                    AlarmActions = alarmActions,
                    AlarmDescription = alarmDescription,
                    AlarmName = alarmName,
                    ComparisonOperator = comparison,
                    Threshold = threshold,
                    Namespace = metricNamespace,
                    MetricName = metricName,
                    EvaluationPeriods = 1,
                    Period = 10,
                    Statistic = new Statistic("Maximum"),
                    DatapointsToAlarm = 1,
                    TreatMissingData = "ignore"
                });
            return putEmailAlarmResponse.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (LimitExceededException lex)
        {
            _logger.LogError(lex, $"Unable to add alarm {alarmName}. Alarm quota
has already been reached.");
        }

        return false;
    }

    /// <summary>
    /// Add specific email actions to a list of action strings for a CloudWatch
alarm.
    /// </summary>
    /// <param name="accountId">The AccountId for the alarm.</param>
    /// <param name="region">The region for the alarm.</param>
    /// <param name="emailTopicName">An Amazon Simple Notification Service (SNS)
topic for the alarm email.</param>
    /// <param name="alarmActions">Optional list of existing alarm actions to
append to.</param>
    /// <returns>A list of string actions for an alarm.</returns>
    public List<string> AddEmailAlarmAction(string accountId, string region,
        string emailTopicName, List<string>? alarmActions = null)
```

```
    {
        alarmActions ??= new List<string>();
        var snsAlarmAction = $"arn:aws:sns:{region}:{accountId}:
{emailTopicName}";
        alarmActions.Add(snsAlarmAction);
        return alarmActions;
    }

    /// <summary>
    /// Describe the current alarms, optionally filtered by state.
    /// </summary>
    /// <param name="stateValue">Optional filter for alarm state.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarms(StateValue? stateValue =
 null)
    {
        List<MetricAlarm> alarms = new List<MetricAlarm>();
        var paginatedDescribeAlarms =
 _amazonCloudWatch.Paginators.DescribeAlarms(
            new DescribeAlarmsRequest()
            {
                StateValue = stateValue
            });

        await foreach (var data in paginatedDescribeAlarms.MetricAlarms)
        {
            alarms.Add(data);
        }
        return alarms;
    }

    /// <summary>
    /// Describe the current alarms for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarmsForMetric(string
 metricNamespace, string metricName)
    {
        var alarmsResult = await _amazonCloudWatch.DescribeAlarmsForMetricAsync(
            new DescribeAlarmsForMetricRequest()
            {
                Namespace = metricNamespace,
```

```
                MetricName = metricName
            });

        return alarmsResult.MetricAlarms;
    }

    /// <summary>
    /// Describe the history of an alarm for a number of days in the past.
    /// </summary>
    /// <param name="alarmName">The name of the alarm.</param>
    /// <param name="historyDays">The number of days in the past.</param>
    /// <returns>The list of alarm history data.</returns>
    public async Task<List<AlarmHistoryItem>> DescribeAlarmHistory(string
alarmName, int historyDays)
    {
        List<AlarmHistoryItem> alarmHistory = new List<AlarmHistoryItem>();
        var paginatedAlarmHistory =
_amazonCloudWatch.Paginators.DescribeAlarmHistory(
            new DescribeAlarmHistoryRequest()
            {
                AlarmName = alarmName,
                EndDateUtc = DateTime.UtcNow,
                HistoryItemType = HistoryItemType.StateUpdate,
                StartDateUtc = DateTime.UtcNow.AddDays(-historyDays)
            });

        await foreach (var data in paginatedAlarmHistory.AlarmHistoryItems)
        {
            alarmHistory.Add(data);
        }
        return alarmHistory;
    }

    /// <summary>
    /// Delete a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAlarms(List<string> alarmNames)
    {
        var deleteAlarmsResult = await _amazonCloudWatch.DeleteAlarmsAsync(
            new DeleteAlarmsRequest()
            {
                AlarmNames = alarmNames
```

```
            });

        return deleteAlarmsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Disable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableAlarmActions(List<string> alarmNames)
    {
        var disableAlarmActionsResult = await
_amazonCloudWatch.DisableAlarmActionsAsync(
            new DisableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return disableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Enable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableAlarmActions(List<string> alarmNames)
    {
        var enableAlarmActionsResult = await
_amazonCloudWatch.EnableAlarmActionsAsync(
            new EnableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return enableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Add an anomaly detector for a single metric.
    /// </summary>
    /// <param name="anomalyDetector">A single metric anomaly detector.</param>
    /// <returns>True if successful.</returns>
```

```
   public async Task<bool> PutAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
   {
       var putAlarmDetectorResult = await
_amazonCloudWatch.PutAnomalyDetectorAsync(
           new PutAnomalyDetectorRequest()
           {
               SingleMetricAnomalyDetector = anomalyDetector
           });

       return putAlarmDetectorResult.HttpStatusCode == HttpStatusCode.OK;
   }

   /// <summary>
   /// Describe anomaly detectors for a metric and namespace.
   /// </summary>
   /// <param name="metricNamespace">The namespace of the metric.</param>
   /// <param name="metricName">The metric of the anomaly detectors.</param>
   /// <returns>The list of detectors.</returns>
   public async Task<List<AnomalyDetector>> DescribeAnomalyDetectors(string
metricNamespace, string metricName)
   {
       List<AnomalyDetector> detectors = new List<AnomalyDetector>();
       var paginatedDescribeAnomalyDetectors =
_amazonCloudWatch.Paginators.DescribeAnomalyDetectors(
           new DescribeAnomalyDetectorsRequest()
           {
               MetricName = metricName,
               Namespace = metricNamespace
           });

       await foreach (var data in
paginatedDescribeAnomalyDetectors.AnomalyDetectors)
       {
           detectors.Add(data);
       }

       return detectors;
   }

   /// <summary>
   /// Delete a single metric anomaly detector.
   /// </summary>
   /// <param name="anomalyDetector">The anomaly detector to delete.</param>
```

```
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAnomalyDetector(SingleMetricAnomalyDetector
 anomalyDetector)
    {
        var deleteAnomalyDetectorResponse = await
 _amazonCloudWatch.DeleteAnomalyDetectorAsync(
            new DeleteAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return deleteAnomalyDetectorResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Delete a list of CloudWatch dashboards.
    /// </summary>
    /// <param name="dashboardNames">List of dashboard names to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDashboards(List<string> dashboardNames)
    {
        var deleteDashboardsResponse = await
 _amazonCloudWatch.DeleteDashboardsAsync(
            new DeleteDashboardsRequest()
            {
                DashboardNames = dashboardNames
            });

        return deleteDashboardsResponse.HttpStatusCode == HttpStatusCode.OK;
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for .NET.
  - DeleteAlarms
  - DeleteAnomalyDetector
  - DeleteDashboards
  - DescribeAlarmHistory
  - DescribeAlarms
  - DescribeAlarmsForMetric

- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Java

SDK para Java 2.x

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
import com.fasterxml.jackson.core.JsonFactory;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.ObjectMapper;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.AlarmHistoryItem;
import software.amazon.awssdk.services.cloudwatch.model.AlarmType;
import software.amazon.awssdk.services.cloudwatch.model.AnomalyDetector;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ComparisonOperator;
import
 software.amazon.awssdk.services.cloudwatch.model.DashboardValidationMessage;
import software.amazon.awssdk.services.cloudwatch.model.Datapoint;
import software.amazon.awssdk.services.cloudwatch.model.DeleteAlarmsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DeleteAnomalyDetectorRequest;
```

```
import software.amazon.awssdk.services.cloudwatch.model.DeleteDashboardsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmHistoryRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmHistoryResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsForMetricRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsForMetricResponse;
import software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsRequest;
import software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAnomalyDetectorsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAnomalyDetectorsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Dimension;
import software.amazon.awssdk.services.cloudwatch.model.GetMetricDataRequest;
import software.amazon.awssdk.services.cloudwatch.model.GetMetricDataResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricStatisticsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricStatisticsResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricWidgetImageRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricWidgetImageResponse;
import software.amazon.awssdk.services.cloudwatch.model.HistoryItemType;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Metric;
import software.amazon.awssdk.services.cloudwatch.model.MetricAlarm;
import software.amazon.awssdk.services.cloudwatch.model.MetricDataQuery;
import software.amazon.awssdk.services.cloudwatch.model.MetricDataResult;
import software.amazon.awssdk.services.cloudwatch.model.MetricDatum;
import software.amazon.awssdk.services.cloudwatch.model.MetricStat;
import
 software.amazon.awssdk.services.cloudwatch.model.PutAnomalyDetectorRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutDashboardRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutDashboardResponse;
import software.amazon.awssdk.services.cloudwatch.model.PutMetricAlarmRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutMetricDataRequest;
import software.amazon.awssdk.services.cloudwatch.model.ScanBy;
import
 software.amazon.awssdk.services.cloudwatch.model.SingleMetricAnomalyDetector;
```

```
import software.amazon.awssdk.services.cloudwatch.model.StandardUnit;
import software.amazon.awssdk.services.cloudwatch.model.Statistic;
import
 software.amazon.awssdk.services.cloudwatch.paginators.ListDashboardsIterable;
import software.amazon.awssdk.services.cloudwatch.paginators.ListMetricsIterable;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.ZoneOffset;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * To enable billing metrics and statistics for this example, make sure billing
 * alerts are enabled for your account:
 * https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics
 *
 * This Java code example performs the following tasks:
 *
 * 1. List available namespaces from Amazon CloudWatch.
 * 2. List available metrics within the selected Namespace.
 * 3. Get statistics for the selected metric over the last day.
 * 4. Get CloudWatch estimated billing for the last week.
 * 5. Create a new CloudWatch dashboard with metrics.
 * 6. List dashboards using a paginator.
 * 7. Create a new custom metric by adding data for it.
```

```
 * 8. Add the custom metric to the dashboard.
 * 9. Create an alarm for the custom metric.
 * 10. Describe current alarms.
 * 11. Get current data for the new custom metric.
 * 12. Push data into the custom metric to trigger the alarm.
 * 13. Check the alarm state using the action DescribeAlarmsForMetric.
 * 14. Get alarm history for the new alarm.
 * 15. Add an anomaly detector for the custom metric.
 * 16. Describe current anomaly detectors.
 * 17. Get a metric image for the custom metric.
 * 18. Clean up the Amazon CloudWatch resources.
 */
public class CloudWatchScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws IOException {
        final String usage = """

                Usage:
                  <myDate> <costDateWeek> <dashboardName> <dashboardJson>
 <dashboardAdd> <settings> <metricImage> \s

                Where:
                  myDate - The start date to use to get metric statistics. (For
 example, 2023-01-11T18:35:24.00Z.)\s
                  costDateWeek - The start date to use to get AWS/Billinget
 statistics. (For example, 2023-01-11T18:35:24.00Z.)\s
                  dashboardName - The name of the dashboard to create.\s
                  dashboardJson - The location of a JSON file to use to create a
 dashboard. (See Readme file.)\s
                  dashboardAdd - The location of a JSON file to use to update a
 dashboard. (See Readme file.)\s
                  settings - The location of a JSON file from which various
 values are read. (See Readme file.)\s
                  metricImage - The location of a BMP file that is used to create
 a graph.\s
                """;

        if (args.length != 7) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        Region region = Region.US_EAST_1;
        String myDate = args[0];
        String costDateWeek = args[1];
        String dashboardName = args[2];
        String dashboardJson = args[3];
        String dashboardAdd = args[4];
        String settings = args[5];
        String metricImage = args[6];

        Double dataPoint = Double.parseDouble("10.0");
        Scanner sc = new Scanner(System.in);
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon CloudWatch example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "1. List at least five available unique namespaces from Amazon
CloudWatch. Select one from the list.");
        ArrayList<String> list = listNameSpaces(cw);
        for (int z = 0; z < 5; z++) {
            int index = z + 1;
            System.out.println("    " + index + ". " + list.get(z));
        }

        String selectedNamespace = "";
        String selectedMetrics = "";
        int num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            selectedNamespace = list.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + selectedNamespace);
        System.out.println(DASHES);

        System.out.println(DASHES);
```

```
        System.out.println("2. List available metrics within the selected
namespace and select one from the list.");
        ArrayList<String> metList = listMets(cw, selectedNamespace);
        for (int z = 0; z < 5; z++) {
            int index = z + 1;
            System.out.println("    " + index + ". " + metList.get(z));
        }
        num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            selectedMetrics = metList.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + selectedMetrics);
        Dimension myDimension = getSpecificMet(cw, selectedNamespace);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Get statistics for the selected metric over the
last day.");
        String metricOption = "";
        ArrayList<String> statTypes = new ArrayList<>();
        statTypes.add("SampleCount");
        statTypes.add("Average");
        statTypes.add("Sum");
        statTypes.add("Minimum");
        statTypes.add("Maximum");

        for (int t = 0; t < 5; t++) {
            System.out.println("    " + (t + 1) + ". " + statTypes.get(t));
        }
        System.out.println("Select a metric statistic by entering a number from
the preceding list:");
        num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            metricOption = statTypes.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + metricOption);
        getAndDisplayMetricStatistics(cw, selectedNamespace, selectedMetrics,
metricOption, myDate, myDimension);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Get CloudWatch estimated billing for the last
week.");
        getMetricStatistics(cw, costDateWeek);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Create a new CloudWatch dashboard with metrics.");
        createDashboardWithMetrics(cw, dashboardName, dashboardJson);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. List dashboards using a paginator.");
        listDashboards(cw);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Create a new custom metric by adding data to
it.");
        createNewCustomMetric(cw, dataPoint);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Add an additional metric to the dashboard.");
        addMetricToDashboard(cw, dashboardAdd, dashboardName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("9. Create an alarm for the custom metric.");
        String alarmName = createAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("10. Describe ten current alarms.");
        describeAlarms(cw);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("11. Get current data for new custom metric.");
        getCustomMetricData(cw, settings);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("12. Push data into the custom metric to trigger the
alarm.");
        addMetricDataForAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Check the alarm state using the action
DescribeAlarmsForMetric.");
        checkForMetricAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Get alarm history for the new alarm.");
        getAlarmHistory(cw, settings, myDate);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("15. Add an anomaly detector for the custom metric.");
        addAnomalyDetector(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("16. Describe current anomaly detectors.");
        describeAnomalyDetectors(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("17. Get a metric image for the custom metric.");
        getAndOpenMetricImage(cw, metricImage);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("18. Clean up the Amazon CloudWatch resources.");
        deleteDashboard(cw, dashboardName);
        deleteCWAlarm(cw, alarmName);
        deleteAnomalyDetector(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Amazon CloudWatch example scenario is
complete.");
        System.out.println(DASHES);
        cw.close();
```

```
   }

   public static void deleteAnomalyDetector(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .stat("Maximum")
                   .build();

           DeleteAnomalyDetectorRequest request =
DeleteAnomalyDetectorRequest.builder()
                   .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                   .build();

           cw.deleteAnomalyDetector(request);
           System.out.println("Successfully deleted the Anomaly Detector.");

       } catch (CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       } catch (IOException e) {
           e.printStackTrace();
       }
   }

   public static void deleteCWAlarm(CloudWatchClient cw, String alarmName) {
       try {
           DeleteAlarmsRequest request = DeleteAlarmsRequest.builder()
                   .alarmNames(alarmName)
                   .build();
```

```
            cw.deleteAlarms(request);
            System.out.println("Successfully deleted alarm " + alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteDashboard(CloudWatchClient cw, String dashboardName)
{
        try {
            DeleteDashboardsRequest dashboardsRequest =
DeleteDashboardsRequest.builder()
                    .dashboardNames(dashboardName)
                    .build();
            cw.deleteDashboards(dashboardsRequest);
            System.out.println(dashboardName + " was successfully deleted.");

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getAndOpenMetricImage(CloudWatchClient cw, String
fileName) {
        System.out.println("Getting Image data for custom metric.");
        try {
            String myJSON = "{\n" +
                    "  \"title\": \"Example Metric Graph\",\n" +
                    "  \"view\": \"timeSeries\",\n" +
                    "  \"stacked \": false,\n" +
                    "  \"period\": 10,\n" +
                    "  \"width\": 1400,\n" +
                    "  \"height\": 600,\n" +
                    "  \"metrics\": [\n" +
                    "    [\n" +
                    "      \"AWS/Billing\",\n" +
                    "      \"EstimatedCharges\",\n" +
                    "      \"Currency\",\n" +
                    "      \"USD\"\n" +
                    "    ]\n" +
                    "  ]\n" +
```

```
                    "}";

            GetMetricWidgetImageRequest imageRequest =
GetMetricWidgetImageRequest.builder()
                    .metricWidget(myJSON)
                    .build();

            GetMetricWidgetImageResponse response =
cw.getMetricWidgetImage(imageRequest);
            SdkBytes sdkBytes = response.metricWidgetImage();
            byte[] bytes = sdkBytes.asByteArray();
            File outputFile = new File(fileName);
            try (FileOutputStream outputStream = new
FileOutputStream(outputFile)) {
                outputStream.write(bytes);
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeAnomalyDetectors(CloudWatchClient cw, String
fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            DescribeAnomalyDetectorsRequest detectorsRequest =
DescribeAnomalyDetectorsRequest.builder()
                    .maxResults(10)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            DescribeAnomalyDetectorsResponse response =
cw.describeAnomalyDetectors(detectorsRequest);
```

```
            List<AnomalyDetector> anomalyDetectorList =
response.anomalyDetectors();
            for (AnomalyDetector detector : anomalyDetectorList) {
                System.out.println("Metric name: " +
detector.singleMetricAnomalyDetector().metricName());
                System.out.println("State: " + detector.stateValue());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void addAnomalyDetector(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .stat("Maximum")
                    .build();

            PutAnomalyDetectorRequest anomalyDetectorRequest =
PutAnomalyDetectorRequest.builder()
                    .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                    .build();

            cw.putAnomalyDetector(anomalyDetectorRequest);
            System.out.println("Added anomaly detector for metric " +
customMetricName + ".");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
```

```
            System.exit(1);
        }
    }

    public static void getAlarmHistory(CloudWatchClient cw, String fileName,
String date) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String alarmName = rootNode.findValue("exampleAlarmName").asText();

            Instant start = Instant.parse(date);
            Instant endDate = Instant.now();
            DescribeAlarmHistoryRequest historyRequest =
DescribeAlarmHistoryRequest.builder()
                    .startDate(start)
                    .endDate(endDate)
                    .alarmName(alarmName)
                    .historyItemType(HistoryItemType.ACTION)
                    .build();

            DescribeAlarmHistoryResponse response =
cw.describeAlarmHistory(historyRequest);
            List<AlarmHistoryItem> historyItems = response.alarmHistoryItems();
            if (historyItems.isEmpty()) {
                System.out.println("No alarm history data found for " + alarmName
+ ".");
            } else {
                for (AlarmHistoryItem item : historyItems) {
                    System.out.println("History summary: " +
item.historySummary());
                    System.out.println("Time stamp: " + item.timestamp());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

```
   public static void checkForMetricAlarm(CloudWatchClient cw, String fileName)
{
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           boolean hasAlarm = false;
           int retries = 10;

           DescribeAlarmsForMetricRequest metricRequest =
DescribeAlarmsForMetricRequest.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           while (!hasAlarm && retries > 0) {
               DescribeAlarmsForMetricResponse response =
cw.describeAlarmsForMetric(metricRequest);
               hasAlarm = response.hasMetricAlarms();
               retries--;
               Thread.sleep(20000);
               System.out.println(".");
           }
           if (!hasAlarm)
               System.out.println("No Alarm state found for " + customMetricName
+ " after 10 retries.");
           else
               System.out.println("Alarm state found for " + customMetricName +
".");

       } catch (CloudWatchException | IOException | InterruptedException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }

   public static void addMetricDataForAlarm(CloudWatchClient cw, String
fileName) {
```

```
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set an Instant object.
            String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
            Instant instant = Instant.parse(time);

            MetricDatum datum = MetricDatum.builder()
                    .metricName(customMetricName)
                    .unit(StandardUnit.NONE)
                    .value(1001.00)
                    .timestamp(instant)
                    .build();

            MetricDatum datum2 = MetricDatum.builder()
                    .metricName(customMetricName)
                    .unit(StandardUnit.NONE)
                    .value(1002.00)
                    .timestamp(instant)
                    .build();

            List<MetricDatum> metricDataList = new ArrayList<>();
            metricDataList.add(datum);
            metricDataList.add(datum2);

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace(customMetricNamespace)
                    .metricData(metricDataList)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric " +
customMetricName);

        } catch (CloudWatchException | IOException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getCustomMetricData(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set the date.
            Instant nowDate = Instant.now();

            long hours = 1;
            long minutes = 30;
            Instant date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(minutes,
                    ChronoUnit.MINUTES);

            Metric met = Metric.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            MetricStat metStat = MetricStat.builder()
                    .stat("Maximum")
                    .period(1)
                    .metric(met)
                    .build();

            MetricDataQuery dataQUery = MetricDataQuery.builder()
                    .metricStat(metStat)
                    .id("foo2")
                    .returnData(true)
                    .build();

            List<MetricDataQuery> dq = new ArrayList<>();
```

```
            dq.add(dataQUery);

            GetMetricDataRequest getMetReq = GetMetricDataRequest.builder()
                    .maxDatapoints(10)
                    .scanBy(ScanBy.TIMESTAMP_DESCENDING)
                    .startTime(nowDate)
                    .endTime(date2)
                    .metricDataQueries(dq)
                    .build();

            GetMetricDataResponse response = cw.getMetricData(getMetReq);
            List<MetricDataResult> data = response.metricDataResults();
            for (MetricDataResult item : data) {
                System.out.println("The label is " + item.label());
                System.out.println("The status code is " +
item.statusCode().toString());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeAlarms(CloudWatchClient cw) {
        try {
            List<AlarmType> typeList = new ArrayList<>();
            typeList.add(AlarmType.METRIC_ALARM);

            DescribeAlarmsRequest alarmsRequest = DescribeAlarmsRequest.builder()
                    .alarmTypes(typeList)
                    .maxRecords(10)
                    .build();

            DescribeAlarmsResponse response = cw.describeAlarms(alarmsRequest);
            List<MetricAlarm> alarmList = response.metricAlarms();
            for (MetricAlarm alarm : alarmList) {
                System.out.println("Alarm name: " + alarm.alarmName());
                System.out.println("Alarm description: " +
alarm.alarmDescription());
            }
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
```

```
        }
    }

    public static String createAlarm(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            String alarmName = rootNode.findValue("exampleAlarmName").asText();
            String emailTopic = rootNode.findValue("emailTopic").asText();
            String accountId = rootNode.findValue("accountId").asText();
            String region = rootNode.findValue("region").asText();

            // Create a List for alarm actions.
            List<String> alarmActions = new ArrayList<>();
            alarmActions.add("arn:aws:sns:" + region + ":" + accountId + ":" +
emailTopic);
            PutMetricAlarmRequest alarmRequest = PutMetricAlarmRequest.builder()
                    .alarmActions(alarmActions)
                    .alarmDescription("Example metric alarm")
                    .alarmName(alarmName)

.comparisonOperator(ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD)
                    .threshold(100.00)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .evaluationPeriods(1)
                    .period(10)
                    .statistic("Maximum")
                    .datapointsToAlarm(1)
                    .treatMissingData("ignore")
                    .build();

            cw.putMetricAlarm(alarmRequest);
            System.out.println(alarmName + " was successfully created!");
            return alarmName;

        } catch (CloudWatchException | IOException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }

    public static void addMetricToDashboard(CloudWatchClient cw, String fileName,
String dashboardName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully updated.");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void createNewCustomMetric(CloudWatchClient cw, Double
dataPoint) {
        try {
            Dimension dimension = Dimension.builder()
                    .name("UNIQUE_PAGES")
                    .value("URLS")
                    .build();

            // Set an Instant object.
            String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
            Instant instant = Instant.parse(time);

            MetricDatum datum = MetricDatum.builder()
                    .metricName("PAGES_VISITED")
                    .unit(StandardUnit.NONE)
                    .value(dataPoint)
                    .timestamp(instant)
                    .dimensions(dimension)
                    .build();
```

```
            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace("SITE/TRAFFIC")
                    .metricData(datum)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric
PAGES_VISITED");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void listDashboards(CloudWatchClient cw) {
        try {
            ListDashboardsIterable listRes = cw.listDashboardsPaginator();
            listRes.stream()
                    .flatMap(r -> r.dashboardEntries().stream())
                    .forEach(entry -> {
                        System.out.println("Dashboard name is: " +
entry.dashboardName());
                        System.out.println("Dashboard ARN is: " +
entry.dashboardArn());
                    });

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void createDashboardWithMetrics(CloudWatchClient cw, String
dashboardName, String fileName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            PutDashboardResponse response = cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully created.");
```

```
            List<DashboardValidationMessage> messages =
response.dashboardValidationMessages();
            if (messages.isEmpty()) {
                System.out.println("There are no messages in the new Dashboard");
            } else {
                for (DashboardValidationMessage message : messages) {
                    System.out.println("Message is: " + message.message());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static String readFileAsString(String file) throws IOException {
        return new String(Files.readAllBytes(Paths.get(file)));
    }

    public static void getMetricStatistics(CloudWatchClient cw, String
costDateWeek) {
        try {
            Instant start = Instant.parse(costDateWeek);
            Instant endDate = Instant.now();
            Dimension dimension = Dimension.builder()
                    .name("Currency")
                    .value("USD")
                    .build();

            List<Dimension> dimensionList = new ArrayList<>();
            dimensionList.add(dimension);
            GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                    .metricName("EstimatedCharges")
                    .namespace("AWS/Billing")
                    .dimensions(dimensionList)
                    .statistics(Statistic.MAXIMUM)
                    .startTime(start)
                    .endTime(endDate)
                    .period(86400)
                    .build();
```

```
            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void getAndDisplayMetricStatistics(CloudWatchClient cw, String
nameSpace, String metVal,
            String metricOption, String date, Dimension myDimension) {
        try {
            Instant start = Instant.parse(date);
            Instant endDate = Instant.now();

            GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                    .endTime(endDate)
                    .startTime(start)
                    .dimensions(myDimension)
                    .metricName(metVal)
                    .namespace(nameSpace)
                    .period(86400)
                    .statistics(Statistic.fromValue(metricOption))
                    .build();

            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
```

```
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static Dimension getSpecificMet(CloudWatchClient cw, String namespace)
{
        try {
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsResponse response = cw.listMetrics(request);
            List<Metric> myList = response.metrics();
            Metric metric = myList.get(0);
            return metric.dimensions().get(0);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static ArrayList<String> listMets(CloudWatchClient cw, String
namespace) {
        try {
            ArrayList<String> metList = new ArrayList<>();
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> metList.add(metrics.metricName()));
```

```
            return metList;

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static ArrayList<String> listNameSpaces(CloudWatchClient cw) {
        try {
            ArrayList<String> nameSpaceList = new ArrayList<>();
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> {
                        String data = metrics.namespace();
                        if (!nameSpaceList.contains(data)) {
                            nameSpaceList.add(data);
                        }
                    });

            return nameSpaceList;
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Java 2.x.
  - DeleteAlarms
  - DeleteAnomalyDetector
  - DeleteDashboards
  - DescribeAlarmHistory

- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Kotlin

SDK for Kotlin

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
/**
 Before running this Kotlin code example, set up your development environment,
 including your credentials.

 For more information, see the following documentation topic:
 https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html

 To enable billing metrics and statistics for this example, make sure billing
 alerts are enabled for your account:
 https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics

 This Kotlin code example performs the following tasks:
```

```
 1. List available namespaces from Amazon CloudWatch. Select a namespace from the
 list.
 2. List available metrics within the selected namespace.
 3. Get statistics for the selected metric over the last day.
 4. Get CloudWatch estimated billing for the last week.
 5. Create a new CloudWatch dashboard with metrics.
 6. List dashboards using a paginator.
 7. Create a new custom metric by adding data for it.
 8. Add the custom metric to the dashboard.
 9. Create an alarm for the custom metric.
 10. Describe current alarms.
 11. Get current data for the new custom metric.
 12. Push data into the custom metric to trigger the alarm.
 13. Check the alarm state using the action DescribeAlarmsForMetric.
 14. Get alarm history for the new alarm.
 15. Add an anomaly detector for the custom metric.
 16. Describe current anomaly detectors.
 17. Get a metric image for the custom metric.
 18. Clean up the Amazon CloudWatch resources.
 */

val DASHES: String? = String(CharArray(80)).replace("\u0000", "-")
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <myDate> <costDateWeek> <dashboardName> <dashboardJson>
 <dashboardAdd> <settings> <metricImage>

        Where:
            myDate - The start date to use to get metric statistics. (For
 example, 2023-01-11T18:35:24.00Z.)
            costDateWeek - The start date to use to get AWS Billing and Cost
 Management statistics. (For example, 2023-01-11T18:35:24.00Z.)
            dashboardName - The name of the dashboard to create.
            dashboardJson - The location of a JSON file to use to create a
 dashboard. (See Readme file.)
            dashboardAdd - The location of a JSON file to use to update a
 dashboard. (See Readme file.)
            settings - The location of a JSON file from which various values are
 read. (See Readme file.)
            metricImage - The location of a BMP file that is used to create a
 graph.
    """
```

```
    if (args.size != 7) {
        println(usage)
        System.exit(1)
    }

    val myDate = args[0]
    val costDateWeek = args[1]
    val dashboardName = args[2]
    val dashboardJson = args[3]
    val dashboardAdd = args[4]
    val settings = args[5]
    var metricImage = args[6]
    val dataPoint = "10.0".toDouble()
    val inOb = Scanner(System.`in`)

    println(DASHES)
    println("Welcome to the Amazon CloudWatch example scenario.")
    println(DASHES)

    println(DASHES)
    println("1. List at least five available unique namespaces from Amazon
 CloudWatch. Select a CloudWatch namespace from the list.")
    val list: ArrayList<String> = listNameSpaces()
    for (z in 0..4) {
        println("    ${z + 1}. ${list[z]}")
    }

    var selectedNamespace: String
    var selectedMetrics = ""
    var num = inOb.nextLine().toInt()
    println("You selected $num")

    if (1 <= num && num <= 5) {
        selectedNamespace = list[num - 1]
    } else {
        println("You did not select a valid option.")
        exitProcess(1)
    }
    println("You selected $selectedNamespace")
    println(DASHES)

    println(DASHES)
    println("2. List available metrics within the selected namespace and select
 one from the list.")
```

```
    val metList = listMets(selectedNamespace)
    for (z in 0..4) {
        println("    ${ z + 1}. ${metList?.get(z)}")
    }
    num = inOb.nextLine().toInt()
    if (1 <= num && num <= 5) {
        selectedMetrics = metList!![num - 1]
    } else {
        println("You did not select a valid option.")
        System.exit(1)
    }
    println("You selected $selectedMetrics")
    val myDimension = getSpecificMet(selectedNamespace)
    if (myDimension == null) {
        println("Error - Dimension is null")
        exitProcess(1)
    }
    println(DASHES)

    println(DASHES)
    println("3. Get statistics for the selected metric over the last day.")
    val metricOption: String
    val statTypes = ArrayList<String>()
    statTypes.add("SampleCount")
    statTypes.add("Average")
    statTypes.add("Sum")
    statTypes.add("Minimum")
    statTypes.add("Maximum")

    for (t in 0..4) {
        println("    ${t + 1}. ${statTypes[t]}")
    }
    println("Select a metric statistic by entering a number from the preceding
list:")
    num = inOb.nextLine().toInt()
    if (1 <= num && num <= 5) {
        metricOption = statTypes[num - 1]
    } else {
        println("You did not select a valid option.")
        exitProcess(1)
    }
    println("You selected $metricOption")
    getAndDisplayMetricStatistics(selectedNamespace, selectedMetrics,
metricOption, myDate, myDimension)
```

```
    println(DASHES)

    println(DASHES)
    println("4. Get CloudWatch estimated billing for the last week.")
    getMetricStatistics(costDateWeek)
    println(DASHES)

    println(DASHES)
    println("5. Create a new CloudWatch dashboard with metrics.")
    createDashboardWithMetrics(dashboardName, dashboardJson)
    println(DASHES)

    println(DASHES)
    println("6. List dashboards using a paginator.")
    listDashboards()
    println(DASHES)

    println(DASHES)
    println("7. Create a new custom metric by adding data to it.")
    createNewCustomMetric(dataPoint)
    println(DASHES)

    println(DASHES)
    println("8. Add an additional metric to the dashboard.")
    addMetricToDashboard(dashboardAdd, dashboardName)
    println(DASHES)

    println(DASHES)
    println("9. Create an alarm for the custom metric.")
    val alarmName: String = createAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("10. Describe 10 current alarms.")
    describeAlarms()
    println(DASHES)

    println(DASHES)
    println("11. Get current data for the new custom metric.")
    getCustomMetricData(settings)
    println(DASHES)

    println(DASHES)
    println("12. Push data into the custom metric to trigger the alarm.")
```

```
    addMetricDataForAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("13. Check the alarm state using the action
 DescribeAlarmsForMetric.")
    checkForMetricAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("14. Get alarm history for the new alarm.")
    getAlarmHistory(settings, myDate)
    println(DASHES)

    println(DASHES)
    println("15. Add an anomaly detector for the custom metric.")
    addAnomalyDetector(settings)
    println(DASHES)

    println(DASHES)
    println("16. Describe current anomaly detectors.")
    describeAnomalyDetectors(settings)
    println(DASHES)

    println(DASHES)
    println("17. Get a metric image for the custom metric.")
    getAndOpenMetricImage(metricImage)
    println(DASHES)

    println(DASHES)
    println("18. Clean up the Amazon CloudWatch resources.")
    deleteDashboard(dashboardName)
    deleteAlarm(alarmName)
    deleteAnomalyDetector(settings)
    println(DASHES)

    println(DASHES)
    println("The Amazon CloudWatch example scenario is complete.")
    println(DASHES)
}

suspend fun deleteAnomalyDetector(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
```

```
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val request = DeleteAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAnomalyDetector(request)
        println("Successfully deleted the Anomaly Detector.")
    }
}

suspend fun deleteAlarm(alarmNameVal: String) {
    val request = DeleteAlarmsRequest {
        alarmNames = listOf(alarmNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAlarms(request)
        println("Successfully deleted alarm $alarmNameVal")
    }
}

suspend fun deleteDashboard(dashboardName: String) {
    val dashboardsRequest = DeleteDashboardsRequest {
        dashboardNames = listOf(dashboardName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteDashboards(dashboardsRequest)
        println("$dashboardName was successfully deleted.")
    }
}

suspend fun getAndOpenMetricImage(fileName: String) {
    println("Getting Image data for custom metric.")
```

```
    val myJSON = """{
        "title": "Example Metric Graph",
        "view": "timeSeries",
        "stacked ": false,
        "period": 10,
        "width": 1400,
        "height": 600,
        "metrics": [
            [
            "AWS/Billing",
            "EstimatedCharges",
            "Currency",
            "USD"
            ]
        ]
        }"""

    val imageRequest = GetMetricWidgetImageRequest {
        metricWidget = myJSON
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricWidgetImage(imageRequest)
        val bytes = response.metricWidgetImage
        if (bytes != null) {
            File(fileName).writeBytes(bytes)
        }
    }
    println("You have successfully written data to $fileName")
}

suspend fun describeAnomalyDetectors(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val detectorsRequest = DescribeAnomalyDetectorsRequest {
        maxResults = 10
        metricName = customMetricName
        namespace = customMetricNamespace
    }
```

```
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAnomalyDetectors(detectorsRequest)
        response.anomalyDetectors?.forEach { detector ->
            println("Metric name:
 ${detector.singleMetricAnomalyDetector?.metricName}")
            println("State: ${detector.stateValue}")
        }
    }
}

suspend fun addAnomalyDetector(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val anomalyDetectorRequest = PutAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putAnomalyDetector(anomalyDetectorRequest)
        println("Added anomaly detector for metric $customMetricName.")
    }
}

suspend fun getAlarmHistory(fileName: String, date: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val start = Instant.parse(date)
    val endDateVal = Instant.now()

    val historyRequest = DescribeAlarmHistoryRequest {
        startDate = aws.smithy.kotlin.runtime.time.Instant(start)
```

```
        endDate = aws.smithy.kotlin.runtime.time.Instant(endDateVal)
        alarmName = alarmNameVal
        historyItemType = HistoryItemType.Action
    }

    CloudWatchClient { credentialsProvider = EnvironmentCredentialsProvider();
 region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarmHistory(historyRequest)
        val historyItems = response.alarmHistoryItems
        if (historyItems != null) {
            if (historyItems.isEmpty()) {
                println("No alarm history data found for $alarmNameVal.")
            } else {
                for (item in historyItems) {
                    println("History summary ${item.historySummary}")
                    println("Time stamp: ${item.timestamp}")
                }
            }
        }
    }
}

suspend fun checkForMetricAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    var hasAlarm = false
    var retries = 10

    val metricRequest = DescribeAlarmsForMetricRequest {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        while (!hasAlarm && retries > 0) {
            val response = cwClient.describeAlarmsForMetric(metricRequest)
            if (response.metricAlarms?.count()!! > 0) {
                hasAlarm = true
            }
            retries--
            delay(20000)
```

```
            println(".")
        }
        if (!hasAlarm) println("No Alarm state found for $customMetricName after
 10 retries.") else println("Alarm state found for $customMetricName.")
    }
}

suspend fun addMetricDataForAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1001.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val datum2 = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1002.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val metricDataList = ArrayList<MetricDatum>()
    metricDataList.add(datum)
    metricDataList.add(datum2)

    val request = PutMetricDataRequest {
        namespace = customMetricNamespace
        metricData = metricDataList
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
```

```
        println("Added metric values for for metric $customMetricName")
    }
}

suspend fun getCustomMetricData(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set the date.
    val nowDate = Instant.now()
    val hours: Long = 1
    val minutes: Long = 30
    val date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(
        minutes,
        ChronoUnit.MINUTES
    )

    val met = Metric {
        metricName = customMetricName
        namespace = customMetricNamespace
    }

    val metStat = MetricStat {
        stat = "Maximum"
        period = 1
        metric = met
    }

    val dataQUery = MetricDataQuery {
        metricStat = metStat
        id = "foo2"
        returnData = true
    }

    val dq = ArrayList<MetricDataQuery>()
    dq.add(dataQUery)
    val getMetReq = GetMetricDataRequest {
        maxDatapoints = 10
        scanBy = ScanBy.TimestampDescending
        startTime = aws.smithy.kotlin.runtime.time.Instant(nowDate)
```

```
        endTime = aws.smithy.kotlin.runtime.time.Instant(date2)
        metricDataQueries = dq
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricData(getMetReq)
        response.metricDataResults?.forEach { item ->
            println("The label is ${item.label}")
            println("The status code is ${item.statusCode}")
        }
    }
}

suspend fun describeAlarms() {
    val typeList = ArrayList<AlarmType>()
    typeList.add(AlarmType.MetricAlarm)
    val alarmsRequest = DescribeAlarmsRequest {
        alarmTypes = typeList
        maxRecords = 10
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarms(alarmsRequest)
        response.metricAlarms?.forEach { alarm ->
            println("Alarm name: ${alarm.alarmName}")
            println("Alarm description: ${alarm.alarmDescription}")
        }
    }
}

suspend fun createAlarm(fileName: String): String {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode: JsonNode = ObjectMapper().readTree(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val emailTopic = rootNode.findValue("emailTopic").asText()
    val accountId = rootNode.findValue("accountId").asText()
    val region2 = rootNode.findValue("region").asText()

    // Create a List for alarm actions.
    val alarmActionObs: MutableList<String> = ArrayList()
```

```
    alarmActionObs.add("arn:aws:sns:$region2:$accountId:$emailTopic")
    val alarmRequest = PutMetricAlarmRequest {
        alarmActions = alarmActionObs
        alarmDescription = "Example metric alarm"
        alarmName = alarmNameVal
        comparisonOperator = ComparisonOperator.GreaterThanOrEqualToThreshold
        threshold = 100.00
        metricName = customMetricName
        namespace = customMetricNamespace
        evaluationPeriods = 1
        period = 10
        statistic = Statistic.Maximum
        datapointsToAlarm = 1
        treatMissingData = "ignore"
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricAlarm(alarmRequest)
        println("$alarmNameVal was successfully created!")
        return alarmNameVal
    }
}

suspend fun addMetricToDashboard(fileNameVal: String, dashboardNameVal: String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully updated.")
    }
}

suspend fun createNewCustomMetric(dataPoint: Double) {
    val dimension = Dimension {
        name = "UNIQUE_PAGES"
        value = "URLS"
    }

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
```

```
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = "PAGES_VISITED"
        unit = StandardUnit.None
        value = dataPoint
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
        dimensions = listOf(dimension)
    }

    val request = PutMetricDataRequest {
        namespace = "SITE/TRAFFIC"
        metricData = listOf(datum)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric PAGES_VISITED")
    }
}

suspend fun listDashboards() {
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listDashboardsPaginated({})
            .transform { it.dashboardEntries?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.dashboardName}")
                println("Dashboard ARN is ${obj.dashboardArn}")
            }
    }
}

suspend fun createDashboardWithMetrics(dashboardNameVal: String, fileNameVal:
 String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully created.")
        val messages = response.dashboardValidationMessages
        if (messages != null) {
            if (messages.isEmpty()) {
```

```
                println("There are no messages in the new Dashboard")
            } else {
                for (message in messages) {
                    println("Message is: ${message.message}")
                }
            }
        }
    }
}

fun readFileAsString(file: String): String {
    return String(Files.readAllBytes(Paths.get(file)))
}

suspend fun getMetricStatistics(costDateWeek: String?) {
    val start = Instant.parse(costDateWeek)
    val endDate = Instant.now()
    val dimension = Dimension {
        name = "Currency"
        value = "USD"
    }

    val dimensionList: MutableList<Dimension> = ArrayList()
    dimensionList.add(dimension)

    val statisticsRequest = GetMetricStatisticsRequest {
        metricName = "EstimatedCharges"
        namespace = "AWS/Billing"
        dimensions = dimensionList
        statistics = listOf(Statistic.Maximum)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        period = 86400
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data: List<Datapoint>? = response.datapoints
        if (data != null) {
            if (!data.isEmpty()) {
                for (datapoint in data) {
                    println("Timestamp:  ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
```

```
                println("The returned data list is empty")
            }
        }
    }
}

suspend fun getAndDisplayMetricStatistics(nameSpaceVal: String, metVal: String,
 metricOption: String, date: String, myDimension: Dimension) {
    val start = Instant.parse(date)
    val endDate = Instant.now()
    val statisticsRequest = GetMetricStatisticsRequest {
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        dimensions = listOf(myDimension)
        metricName = metVal
        namespace = nameSpaceVal
        period = 86400
        statistics = listOf(Statistic.fromValue(metricOption))
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data = response.datapoints
        if (data != null) {
            if (data.isNotEmpty()) {
                for (datapoint in data) {
                    println("Timestamp: ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}

suspend fun listMets(namespaceVal: String?): ArrayList<String>? {
    val metList = ArrayList<String>()
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val reponse = cwClient.listMetrics(request)
        reponse.metrics?.forEach { metrics ->
```

```
            val data = metrics.metricName
            if (!metList.contains(data)) {
                metList.add(data!!)
            }
        }
    }
    return metList
}

suspend fun getSpecificMet(namespaceVal: String?): Dimension? {
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.listMetrics(request)
        val myList = response.metrics
        if (myList != null) {
            return myList[0].dimensions?.get(0)
        }
    }
    return null
}

suspend fun listNameSpaces(): ArrayList<String> {
    val nameSpaceList = ArrayList<String>()
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.listMetrics(ListMetricsRequest {})
        response.metrics?.forEach { metrics ->
            val data = metrics.namespace
            if (!nameSpaceList.contains(data)) {
                nameSpaceList.add(data!!)
            }
        }
    }
    return nameSpaceList
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Kotlin.
  - DeleteAlarms
  - DeleteAnomalyDetector

- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Gerenciar métricas e alarmes do CloudWatch usando um AWS SDK

O exemplo de código a seguir mostra como:

- Criar um alarme para observar uma métrica do CloudWatch.
- Inserir dados em uma métrica e acionar o alarme.
- Obter dados do alarme.
- Excluir o alarme.

Python

SDK para Python (Boto3).

> **Note**
>
> Há mais no GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Crie uma classe que envolva operações do CloudWatch.

```
from datetime import datetime, timedelta
import logging
from pprint import pprint
import random
import time
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data_set(self, namespace, name, timestamp, unit, data_set):
        """
        Sends a set of data to CloudWatch for a metric. All of the data in the
 set
        have the same timestamp and unit.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param timestamp: The UTC timestamp for the metric.
        :param unit: The unit of the metric.
```

```
        :param data_set: The set of data to send. This set is a dictionary that
                         contains a list of values and a list of corresponding
counts.
                         The value and count lists must be the same length.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[
                    {
                        "MetricName": name,
                        "Timestamp": timestamp,
                        "Values": data_set["values"],
                        "Counts": data_set["counts"],
                        "Unit": unit,
                    }
                ],
            )
            logger.info("Put data set for metric %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data set for metric %s.%s.",
namespace, name)
            raise


    def create_metric_alarm(
        self,
        metric_namespace,
        metric_name,
        alarm_name,
        stat_type,
        period,
        eval_periods,
        threshold,
        comparison_op,
    ):
        """
        Creates an alarm that watches a metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :param alarm_name: The name of the alarm.
        :param stat_type: The type of statistic the alarm watches.
```

```
        :param period: The period in which metric data are grouped to calculate
                       statistics.
        :param eval_periods: The number of periods that the metric must be over
the
                             alarm threshold before the alarm is set into an
alarmed
                             state.
        :param threshold: The threshold value to compare against the metric
statistic.
        :param comparison_op: The comparison operation used to compare the
threshold
                              against the metric.
        :return: The newly created alarm.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            alarm = metric.put_alarm(
                AlarmName=alarm_name,
                Statistic=stat_type,
                Period=period,
                EvaluationPeriods=eval_periods,
                Threshold=threshold,
                ComparisonOperator=comparison_op,
            )
            logger.info(
                "Added alarm %s to track metric %s.%s.",
                alarm_name,
                metric_namespace,
                metric_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't add alarm %s to metric %s.%s",
                alarm_name,
                metric_namespace,
                metric_name,
            )
            raise
        else:
            return alarm


    def put_metric_data(self, namespace, name, value, unit):
```

```
        """
        Sends a single data value to CloudWatch for a metric. This metric is
given
        a timestamp of the current UTC time.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param value: The value of the metric.
        :param unit: The unit of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{"MetricName": name, "Value": value, "Unit": unit}],
            )
            logger.info("Put data for metric %s.%s", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data for metric %s.%s", namespace,
name)
            raise


    def get_metric_statistics(self, namespace, name, start, end, period,
stat_types):
        """
        Gets statistics for a metric within a specified time span. Metrics are
grouped
        into the specified period.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param start: The UTC start time of the time span to retrieve.
        :param end: The UTC end time of the time span to retrieve.
        :param period: The period, in seconds, in which to group metrics. The
period
                       must match the granularity of the metric, which depends on
                       the metric's age. For example, metrics that are older than
                       three hours have a one-minute granularity, so the period
must
                       be at least 60 and must be a multiple of 60.
        :param stat_types: The type of statistics to retrieve, such as average
value
                           or maximum value.
```

```
        :return: The retrieved statistics for the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            stats = metric.get_statistics(
                StartTime=start, EndTime=end, Period=period,
Statistics=stat_types
            )
            logger.info(
                "Got %s statistics for %s.", len(stats["Datapoints"]),
stats["Label"]
            )
        except ClientError:
            logger.exception("Couldn't get statistics for %s.%s.", namespace,
name)
            raise
        else:
            return stats


    def get_metric_alarms(self, metric_namespace, metric_name):
        """
        Gets the alarms that are currently watching the specified metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :returns: An iterator that yields the alarms.
        """
        metric = self.cloudwatch_resource.Metric(metric_namespace, metric_name)
        alarm_iter = metric.alarms.all()
        logger.info("Got alarms for metric %s.%s.", metric_namespace,
metric_name)
        return alarm_iter


    def delete_metric_alarms(self, metric_namespace, metric_name):
        """
        Deletes all of the alarms that are currently watching the specified
metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        """
        try:
```

```
            metric = self.cloudwatch_resource.Metric(metric_namespace,
 metric_name)
            metric.alarms.delete()
            logger.info(
                "Deleted alarms for metric %s.%s.", metric_namespace, metric_name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete alarms for metric %s.%s.",
                metric_namespace,
                metric_name,
            )
            raise
```

Use a classe criada para colocar dados em uma métrica, acionar um alarme que observa a métrica e obter dados do alarme.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon CloudWatch metrics and alarms demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    cw_wrapper = CloudWatchWrapper(boto3.resource("cloudwatch"))

    minutes = 20
    metric_namespace = "doc-example-metric"
    metric_name = "page_views"
    start = datetime.utcnow() - timedelta(minutes=minutes)
    print(
        f"Putting data into metric {metric_namespace}.{metric_name} spanning the
 "
        f"last {minutes} minutes."
    )
    for offset in range(0, minutes):
        stamp = start + timedelta(minutes=offset)
        cw_wrapper.put_metric_data_set(
            metric_namespace,
            metric_name,
```

```
            stamp,
            "Count",
            {
                "values": [
                    random.randint(bound, bound * 2)
                    for bound in range(offset + 1, offset + 11)
                ],
                "counts": [random.randint(1, offset + 1) for _ in range(10)],
            },
        )

    alarm_name = "high_page_views"
    period = 60
    eval_periods = 2
    print(f"Creating alarm {alarm_name} for metric {metric_name}.")
    alarm = cw_wrapper.create_metric_alarm(
        metric_namespace,
        metric_name,
        alarm_name,
        "Maximum",
        period,
        eval_periods,
        100,
        "GreaterThanThreshold",
    )
    print(f"Alarm ARN is {alarm.alarm_arn}.")
    print(f"Current alarm state is: {alarm.state_value}.")

    print(
        f"Sending data to trigger the alarm. This requires data over the
threshold "
        f"for {eval_periods} periods of {period} seconds each."
    )
    while alarm.state_value == "INSUFFICIENT_DATA":
        print("Sending data for the metric.")
        cw_wrapper.put_metric_data(
            metric_namespace, metric_name, random.randint(100, 200), "Count"
        )
        alarm.load()
        print(f"Current alarm state is: {alarm.state_value}.")
        if alarm.state_value == "INSUFFICIENT_DATA":
            print(f"Waiting for {period} seconds...")
            time.sleep(period)
        else:
```

```
            print("Wait for a minute for eventual consistency of metric data.")
            time.sleep(period)
            if alarm.state_value == "OK":
                alarm.load()
                print(f"Current alarm state is: {alarm.state_value}.")

    print(
        f"Getting data for metric {metric_namespace}.{metric_name} during
timespan "
        f"of {start} to {datetime.utcnow()} (times are UTC)."
    )
    stats = cw_wrapper.get_metric_statistics(
        metric_namespace,
        metric_name,
        start,
        datetime.utcnow(),
        60,
        ["Average", "Minimum", "Maximum"],
    )
    print(
        f"Got {len(stats['Datapoints'])} data points for metric "
        f"{metric_namespace}.{metric_name}."
    )
    pprint(sorted(stats["Datapoints"], key=lambda x: x["Timestamp"]))

    print(f"Getting alarms for metric {metric_name}.")
    alarms = cw_wrapper.get_metric_alarms(metric_namespace, metric_name)
    for alarm in alarms:
        print(f"Alarm {alarm.name} is currently in state {alarm.state_value}.")

    print(f"Deleting alarms for metric {metric_name}.")
    cw_wrapper.delete_metric_alarms(metric_namespace, metric_name)

    print("Thanks for watching!")
    print("-" * 88)
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Python (Boto3).
  - DeleteAlarms

- DescribeAlarmsForMetric
- DisableAlarmActions
- EnableAlarmActions
- GetMetricStatistics
- ListMetrics
- PutMetricAlarm
- PutMetricData

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de serviços cruzados para o CloudWatch Logs usando AWS SDKs

Os exemplos de aplicações, apresentados a seguir, usam AWS SDKs para combinar o CloudWatch Logs com outros Serviços da AWS. Cada exemplo inclui um link para o GitHub, em que é possível encontrar instruções sobre como configurar e executar a aplicação.

Exemplos

- Monitoramento do desempenho do Amazon DynamoDB usando um AWS SDK

Monitoramento do desempenho do Amazon DynamoDB usando um AWS SDK

O exemplo de código, apresentado a seguir, mostra como configurar o uso do DynamoDB por uma aplicação para monitorar o desempenho.

Java

SDK para Java 2.x

Este exemplo mostra como configurar uma aplicação em Java para monitorar o desempenho do DynamoDB. A aplicação envia dados de métricas para o CloudWatch, que é um local em que você pode monitorar o desempenho.

Para obter o código-fonte completo e instruções sobre como configurar e executar o exemplo, consulte o exemplo completo no GitHub.

Serviços utilizados neste exemplo

- CloudWatch
- DynamoDB

Para ver uma lista completa dos Guias do desenvolvedor de SDK da AWS e exemplos de código, consulte Como usar o CloudWatch com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Segurança no Amazon CloudWatch

A segurança na nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você contará com um datacenter e uma arquitetura de rede criados para atender aos requisitos das organizações com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O modelo de responsabilidade compartilhada descreve isso como a segurança da nuvem e a segurança na nuvem:

- Segurança da nuvem –- a AWS é responsável pela proteção da infraestrutura que executa serviços AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos AWSProgramas de Conformidade. Para saber mais sobre os programas de conformidade aplicáveis ao CloudWatch, consulte Serviços da AWS no escopo por programa de conformidade.
- Segurança na nuvem: sua responsabilidade é determinada pelo 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 os regulamentos aplicáveis

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon CloudWatch. Mostra como configurar o Amazon CloudWatch para atender aos objetivos de segurança e compatibilidade. Você também saberá mais sobre como usar outros produtos da AWS que ajudam a monitorar e proteger os recursos do CloudWatch.

Conteúdos

Proteção de dados no Amazon CloudWatch

O modelo de responsabilidade compartilhada da AWS se aplica à proteção de dados no Amazon CloudWatch. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para mais informações sobre a proteção de dados na Europa, consulte o artigo AWS Shared Responsibility Model and GDPR no Blog de segurança da AWS.

Para fins de proteção de dados, recomendamos que você proteja asConta da AWS credenciais da e configure as contas de usuário individuais com o AWS IAM Identity Center ou o AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA [multi-factor authentication]) com cada conta.
- Use SSL/TLS para se comunicar com os atributos da AWS. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure o registro em log das atividades da API e do usuário com o .AWS CloudTrail
- Use AWS as soluções de criptografia da , juntamente com todos os controles de segurança padrão dos Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comandos ou uma API, use um endpoint do FIPS. Para ter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso vale para quando você trabalha com o CloudWatch ou outros Serviços da AWS que usam o console, a API, a AWS CLI ou os SDKs da AWS. Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendemos

fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Criptografia em trânsito

O CloudWatch usa criptografia de ponta a ponta de dados em trânsito.

Gerenciamento de Identidade e Acesso para o Amazon CloudWatch

O AWS Identity and Access Management (IAM) é um serviço da AWS service (Serviço da AWS) que ajuda a controlar o acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode ser autenticado (fazer login) e autorizado (ter permissões) para usar os recursos do CloudWatch. O IAM é um AWS service (Serviço da AWS) que pode ser usado sem custo adicional.

Tópicos

- Políticas gerenciadas pela AWS para o Amazon CloudWatch Application Insights
- Referência de permissões do Amazon CloudWatch

Público

O seu uso do AWS Identity and Access Management (IAM) varia, dependendo do trabalho realizado no CloudWatch.

Usuário do serviço: se você usar o serviço CloudWatch para fazer o trabalho, o administrador fornecerá as credenciais e as permissões necessárias. À medida que usar mais recursos do CloudWatch para fazer seu trabalho, será possível precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no CloudWatch, consulte Solução de problemas de identidade e acesso da Amazon CloudWatch.

Administrador do serviço: se você for o responsável pelos recursos do CloudWatch na empresa, provavelmente terá acesso total ao CloudWatch. É seu trabalho determinar quais funcionalidades e recursos do CloudWatch seus usuários de serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como a empresa pode usar o IAM com o CloudWatch, consulte Como o Amazon CloudWatch funciona com o IAM.

Administrador do IAM: se você for um administrador do IAM, talvez queira saber detalhes sobre a criação de políticas para gerenciar o acesso ao CloudWatch. Para visualizar exemplos de políticas baseadas em identidade do CloudWatch que podem ser usadas no IAM, consulte Exemplos de políticas baseadas em identidade do Amazon CloudWatch.

Autenticando com identidades

A autenticação é a forma como você faz login na AWS usando suas credenciais de identidade. É necessário ser autenticado (fazer login na AWS) como o usuário-raiz da Usuário raiz da conta da AWS, como usuário do IAM ou assumindo um perfil do IAM.

Você pode fazer login na AWS como uma identidade federada usando credenciais fornecidas por uma fonte de identidades. AWS IAM Identity Center Aos usuários do IAM Identity Center, a autenticação única da empresa e as suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como uma identidade federada, o administrador já

configurou anteriormente a federação de identidades usando perfis do IAM. Quando você acessa a AWS usando a federação, está indiretamente assumindo um perfil.

É possível fazer login no ou no portal de acesso da AWS Management Console dependendo do tipo de usuário que você é.AWS Para obter mais informações sobre como fazer login na AWS, consulte How to sign in to your (Como fazer login na conta da) Conta da AWS no Início de Sessão da AWS User Guide (Guia do usuário do)..

Se você acessar a AWS programaticamente, a AWS fornecerá um kit de desenvolvimento de software (SDK) e uma interface da linha de comando (CLI) para você assinar criptograficamente as solicitações usando as suas credenciais. Se você não utilizar as ferramentas da AWS, deverá assinar as solicitações por conta própria. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte Assinar solicitações de API da AWS no Guia do usuário do IAM.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, a AWS recomenda o uso da autenticação multifator (MFA) para aumentar a segurança de sua conta. Para saber mais, consulte Autenticação multifator no Guia do usuário do AWS IAM Identity Center. Usar a autenticação multifator (MFA) na AWS no Guia do usuário do IAM.

Usuário-raiz da Conta da AWS

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos os atributos e Serviços da AWS na conta. Essa identidade, denominada usuário-raiz da Conta da AWS, e é acessada por login com o endereço de email e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele pode executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte Tarefas que exigem credenciais de usuário-raiz no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários, inclusive os que precisam de acesso de administrador, usem a federação com um provedor de identidades para acessar Serviços da AWS usando credenciais temporárias.

Identidade federada é um usuário de seu diretório de usuários corporativos, um provedor de identidades da web AWS Directory Service, o , o diretório do Centro de Identidade ou qualquer

usuário que acesse os Serviços da AWS usando credenciais fornecidas por meio de uma fonte de identidade. Quando as identidades federadas acessam Contas da AWS, elas assumem perfis que fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o .AWS IAM Identity Center Você pode criar usuários e grupos no Centro de Identidade do IAM ou se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todas as suas Contas da AWS e aplicações. Para obter mais informações sobre o Centro de Identidade do IAM, consulte "What is IAM Identity Center?" (O que é o Centro de Identidade do IAM?) noAWS IAM Identity Center Guia do usuário do .

Grupos e usuários do IAM

Um usuário do IAM é uma identidade dentro da Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos depender de credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte Alterne as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e atribuir a esse grupo permissões para administrar atributos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando criar um usuário do IAM (em vez de um perfil) no Guia do usuário do IAM.

Perfis do IAM

um perfil do IAMhttps://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html é uma identidade dentro da Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. É possível assumir temporariamente um perfil do IAM no AWS Management Console alternando perfis. É possível assumir um perfil chamando

uma operação de API da AWS CLI ou da AWS, ou usando um URL personalizado. Para obter mais informações sobre métodos para o uso de perfis, consulte Usar perfis do IAM no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidades de terceiros no Guia do usuário do IAM. Se você usar o IAM Identity Center, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de permissões noAWS IAM Identity Center Guia do usuário do .
- Permissões temporárias para usuários do IAM: um usuário ou um perfil do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, alguns Serviços da AWS permitem que você anexe uma política diretamente a um atributo (em vez de usar um perfil como proxy). Para saber a diferença entre perfis e políticas baseadas em atributo para acesso entre contas, consulte Como os perfis do IAM diferem das políticas baseadas em atributo no Guia do usuário do IAM.
- Acesso entre serviços: alguns Serviços da AWS usam atributos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicações no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal de chamada, usando um perfil de serviço ou uma função vinculada ao serviço.
  - Encaminhamento de sessões de acesso (FAS): qualquer pessoa que utilizar uma função ou usuário do IAM para realizar ações na AWS é considerada uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações.

Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

- Perfil de serviço: um perfil de serviço é um perfil do IAMhttps://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do usuário do IAM.
- Perfil vinculado a serviço: um perfil vinculado a serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode assumir um perfil para executar uma ação em seu nome. Os perfis vinculados ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

- Aplicações em execução no Amazon EC2: é possível usar um perfil do IAM para gerenciar credenciais temporárias para aplicações em execução em uma instância do EC2 e fazer solicitações da AWS CLI ou da AWS API. É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir um perfil da AWS a uma instância do EC2 e disponibilizá-la para todas as suas aplicações, crie um perfil de instância que esteja anexado a ela. Um perfil de instância contém a perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Usar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para saber se deseja usar as funções do IAM, consulte Quando criar um perfil do IAM (em vez de um usuário) no Guia do usuário do IAM.

Gerenciamento do acesso usando políticas

Você controla o acesso na AWS criando políticas e anexando-as a identidades ou atributos da AWS. Uma política é um objeto na AWS que, quando associado a uma identidade ou recurso, define suas permissões. A AWS avalia essas políticas quando uma entidade principal (usuário, usuário-raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada na AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM a perfis, e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de perfis do AWS Management Console, da AWS CLI ou da API da AWS.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda mais como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas independentes que podem ser anexadas a vários usuários, grupos e perfis na Conta da AWS. As políticas gerenciadas incluem políticas gerenciadas pela AWS e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. As entidades principais podem incluir contas, usuários, perfis, usuários federados ou Serviços da AWS.

Políticas baseadas em atributos são políticas em linha que estão localizadas nesse serviço. Não é possível usar as políticas gerenciadas da AWS do IAM em uma política baseada em atributos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Amazon S3, AWS WAF e Amazon VPC são exemplos de serviços compatíveis ACLs. Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

A AWS é compatível com tipos de política menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (perfil ou usuário do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade e dos seus limites de permissões. As políticas baseadas em atributo que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs): SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (UO) no AWS Organizations. O AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS pertencentes à empresa. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades em contas-membro, incluindo cada .Usuário raiz da conta da AWS Para obter mais informações sobre o Organizações e SCPs, consulte Como os SCPs funcionam no Guia do usuário do AWS Organizations.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas

substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como a AWS determina se deve permitir uma solicitação quando há vários tipos de política envolvidos, consulte Lógica da avaliaçãode políticas no Guia do usuário do IAM.

Como o Amazon CloudWatch funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao CloudWatch, saiba quais recursos do IAM estão disponíveis para uso com o CloudWatch.

Recursos do IAM que podem ser usados com o Amazon CloudWatch

Atributo do IAM	Suporte ao CloudWatch
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não
Ações de políticas	Sim
atributos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim

Atributo do IAM	Suporte ao CloudWatch
Perfis vinculados ao serviço	Não

Para obter uma visão geral de como o CloudWatch e outros serviços da AWS funcionam com a maioria dos recursos do IAM, consulte Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Políticas baseadas em identidade para o CloudWatch

É compatível com políticas baseadas em identidade	Sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou atributos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Não é possível especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou função à qual ela está anexado. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elementos da política JSON do IAM no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para o CloudWatch

Para visualizar exemplos de políticas baseadas em identidade do CloudWatch, consulte Exemplos de políticas baseadas em identidade do Amazon CloudWatch.

Políticas baseadas em recursos no CloudWatch

Oferece suporte a políticas baseadas em recursos	Não

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse atributo e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. As entidades principais podem incluir contas, usuários, funções, usuários federados ou Serviços da AWS.

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em atributo. Adicionar uma entidade principal entre contas à política baseada em atributo é apenas metade da tarefa de estabelecimento da relação de confiança. Quando a entidade principal e o atributo estão em diferentes Contas da AWS, um administrador do IAM da conta confiável também deve conceder à entidade principal (usuário ou perfil) permissão para acessar o atributo. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em atributo conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Ações de políticas do CloudWatch

Oferece suporte a ações de políticas	Sim

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista das ações do CloudWatch, consulte Ações definidas pelo Amazon CloudWatch na Referência de autorização de serviço.

As ações de políticas no CloudWatch usam o prefixo a seguir antes da ação:

```
cloudwatch
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [
      "cloudwatch:action1",
      "cloudwatch:action2"
         ]
```

Para visualizar exemplos de políticas baseadas em identidade do CloudWatch, consulte Exemplos de políticas baseadas em identidade do Amazon CloudWatch.

Recursos de políticas do CloudWatch

Oferece suporte a atributos de políticas	Sim

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando Nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem suporte a um tipo de atributo específico, conhecido como permissões em nível de atributo.

Para ações não compatíveis com permissões no nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de recursos do CloudWatch e seus ARNs, consulte Tipos de recursos definidos pelo Amazon CloudWatch na Referência de autorização do serviço. Para saber com quais ações é possível especificar o ARN de cada recurso, consulte Ações definidas pelo Amazon CloudWatch.

Para visualizar exemplos de políticas baseadas em identidade do CloudWatch, consulte Exemplos de políticas baseadas em identidade do Amazon CloudWatch.

Chaves de condição de políticas para o CloudWatch

Compatível com chaves de condição de política específicas do serviço	Sim

Os administradores podem usar as políticas da JSON AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou `Condition` bloco de) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação lógica `OR`. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos de política do IAM: variáveis e tags no Guia do usuário do IAM.

A AWS é compatível com chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as AWS chaves de condição globais da , consulte AWSChaves de contexto de condição globais da no Guia do usuário do IAM.

Para ver uma lista de chaves de condição do CloudWatch, consulte Chaves de condição do Amazon CloudWatch na Referência de autorização do serviço. Para saber com quais ações e recursos é possível usar a chave de condição, consulte Ações definidas pelo Amazon CloudWatch.

Para visualizar exemplos de políticas baseadas em identidade do CloudWatch, consulte Exemplos de políticas baseadas em identidade do Amazon CloudWatch.

ACLs no CloudWatch

Oferece suporte a ACLs	Não

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com o CloudWatch

Oferece suporte a ABAC (tags em políticas)	Parcial

O controle de acesso baseado em recurso (ABAC) é uma estratégia de autorização que define permissões com base em recursos. Na AWS, esses atributos são chamados de tags. É possível anexar tags a entidades do IAM (usuários ou perfis) e a muitos atributos da AWS. A marcação de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do atributo que ela está tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as chaves de condição `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial.

Para obter mais informações sobre o ABAC, consulte O que é ABAC? no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do usuário do IAM.

Uso de credenciais temporárias com o CloudWatch

Oferece suporte a credenciais temporárias	Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Você está usando credenciais temporárias se faz login no AWS Management Console usando qualquer método, exceto um nome de usuário e uma senha. Por exemplo, quando você acessa a AWS usando o link de autenticação única (SSO) da empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar perfis, consulte Alternar para um perfil (console) no Guia do usuário do IAM.

Você pode criar credenciais temporárias manualmente usando a AWS CLI ou a API da AWS. Em seguida, você pode usar essas credenciais temporárias para acessar aAWS .AWS A recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para mais informações, consulte Credenciais de segurança temporárias no IAM.

Permissões de entidade principal entre serviços do CloudWatch

Suporte para o recurso Encaminhamento de sessões de acesso (FAS)	Sim

Quando você usa um usuário do IAM ou um perfil para executar ações na AWS, você é considerado uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

Perfis de serviço para o CloudWatch

Oferece suporte a perfis de serviço	Sim

Um perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do usuário do IAM.

> **Warning**
>
> Alterar as permissões de um perfil de serviço pode interromper a funcionalidade do CloudWatch. Edite perfis de serviço somente quando o CloudWatch fornecer orientação para isso.

Exemplos de políticas baseadas em identidade do Amazon CloudWatch

Por padrão, usuários e funções não têm permissão para criar nem modificar recursos do CloudWatch. Eles também não podem executar tarefas usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou a API AWS. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis, e os usuários podem assumir os perfis.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documento de política JSON, consulte Criação de políticas do IAM no Guia do Usuário do IAM.

Para obter detalhes sobre ações e tipos de recurso definidos pelo CloudWatch, por exemplo, o formato dos ARNs para cada um dos tipos de recurso, consulte Ações, recursos e chaves de condição do Amazon CloudWatch na Referência de autorização de serviço.

Tópicos

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do CloudWatch em sua conta. Essas ações podem incorrer em custos para a Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas gerenciadas pela AWS e avance para as permissões de privilégio mínimo: para começar a conceder permissões a seus usuários e workloads, use as políticas gerenciadas pela AWS que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis na sua Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo cliente AWS específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para perfis de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e atributos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso a ações de serviço, se elas forem usadas por meio de um AWS service (Serviço da AWS) específico, como o AWS CloudFormation. Para obter mais informações, consulte Elementos de política JSON do IAM: condições no Manual do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações acionáveis para ajudar você a criar políticas seguras e funcionais. Para obter mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do usuário do IAM.
- Exigir autenticação multifator (MFA): se houver um cenário que exija usuários do IAM ou um usuário raiz em sua Conta da AWS, ative a MFA para obter segurança adicional. Para exigir a MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do usuário do IAM.

Para mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usando o console do CloudWatch

Para acessar o console do Amazon CloudWatch, você deve ter um conjunto mínimo de permissões. Essas permissões devem dar autorização para que você liste e visualize detalhes sobre os recursos do CloudWatch em sua Conta da AWS. Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Não é necessário conceder permissões mínimas do console para usuários que fazem chamadas somente à AWS CLI ou à AWS API. Em vez disso, permita o acesso somente a ações que correspondam a operação de API que estiverem tentando executar.

Para garantir que usuários e perfis ainda possam usar o console do CloCloudWatchudFront, anexe também a política *ConsoleAccess* ou *ReadOnly* gerenciada pela AWS do CloudWatch às entidades. Para obter mais informações, consulte Adicionando Permissões a um Usuário no Guia do Usuário do IAM.

Permissões necessárias para o console CloudWatch

O conjunto completo de permissões necessárias para trabalhar com o console do CloudWatch está listado abaixo. Estas permissões fornecem acesso total de gravação e leitura ao console do CloudWatch.

- application-autoscaling:DescribeScalingPolicies
- autoscaling:DescribeAutoScalingGroups
- autoscaling:DescribePolicies
- cloudtrail:DescribeTrails
- cloudwatch:DeleteAlarms
- cloudwatch:DescribeAlarmHistory
- cloudwatch:DescribeAlarms
- cloudwatch:GetMetricData
- cloudwatch:GetMetricStatistics
- cloudwatch:ListMetrics
- cloudwatch:PutMetricAlarm
- cloudwatch:PutMetricData
- ec2:DescribeInstances

- ec2:DescribeTags
- ec2:DescribeVolumes
- es:DescribeElasticsearchDomain
- es:ListDomainNames
- events:DeleteRule
- events:DescribeRule
- events:DisableRule
- events:EnableRule
- events:ListRules
- events:PutRule
- iam:AttachRolePolicy
- iam:CreateRole
- iam:GetPolicy
- iam:GetPolicyVersion
- iam:GetRole
- iam:ListAttachedRolePolicies
- iam:ListRoles
- kinesis:DescribeStream
- kinesis:ListStreams
- lambda:AddPermission
- lambda:CreateFunction
- lambda:GetFunctionConfiguration
- lambda:ListAliases
- lambda:ListFunctions
- lambda:ListVersionsByFunction
- lambda:RemovePermission
- logs:CancelExportTask
- logs:CreateExportTask
- logs:CreateLogGroup
- logs:CreateLogStream

- logs:DeleteLogGroup
- logs:DeleteLogStream
- logs:DeleteMetricFilter
- logs:DeleteRetentionPolicy
- logs:DeleteSubscriptionFilter
- logs:DescribeExportTasks
- logs:DescribeLogGroups
- logs:DescribeLogStreams
- logs:DescribeMetricFilters
- logs:DescribeQueries
- logs:DescribeSubscriptionFilters
- logs:FilterLogEvents
- logs:GetLogGroupFields
- logs:GetLogRecord
- logs:GetLogEvents
- logs:GetQueryResults
- logs:PutMetricFilter
- logs:PutRetentionPolicy
- logs:PutSubscriptionFilter
- logs:StartQuery
- logs:StopQuery
- logs:TestMetricFilter
- s3:CreateBucket
- s3:ListBucket
- sns:CreateTopic
- sns:GetTopicAttributes
- sns:ListSubscriptions
- sns:ListTopics
- sns:SetTopicAttributes
- sns:Subscribe

- sns:Unsubscribe
- sqs:GetQueueAttributes
- sqs:GetQueueUrl
- sqs:ListQueues
- sqs:SetQueueAttributes
- swf:CreateAction
- swf:DescribeAction
- swf:ListActionTemplates
- swf:RegisterAction
- swf:RegisterDomain
- swf:UpdateAction

Além disso, para visualizar o mapa de rastreamento do X-Ray, você precisa da permissão `AWSXrayReadOnlyAccess`.

Solução de problemas de identidade e acesso da Amazon CloudWatch

Use as informações a seguir para ajudar a diagnosticar e corrigir problemas comuns que você possa encontrar ao trabalhar com o CloudWatch e o IAM.

Tópicos

Não tenho autorização para executar uma ação no CloudWatch

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo *`my-example-widget`* fictício, mas não tem as permissões `cloudwatch:`*`GetWidget`* fictícias.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 cloudwatch:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso *`my-example-widget`* usando a ação `cloudwatch:`*`GetWidget`*.

Se você precisar de ajuda, entre em contato com seu administrador da AWS. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a executar iam:PassRole

Se você receber uma mensagem de erro informando que não tem autorização para realizar a ação `iam:PassRole`, as suas políticas deverão ser atualizadas para permitir que você passe um perfil para o CloudWatch.

Alguns Serviços da AWS permitem que você passe um perfil existente para o serviço, em vez de criar um novo perfil de serviço ou perfil vinculado ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O erro de exemplo a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para realizar uma ação no CloudWatch. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se você precisar de ajuda, entre em contato com seu administrador da AWS. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que as pessoas fora da minha Conta da AWS acessem meus recursos do CloudWatch

Você pode criar uma função que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços compatíveis com políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o CloudWatch oferece suporte a esses recursos, consulte Como o Amazon CloudWatch funciona com o IAM.
- Para saber como conceder acesso a seus recursos em todas as Contas da AWS pertencentes a você, consulte Fornecimento de acesso a um usuário do IAM em outra Conta da AWS pertencente a você no Guia de usuário do IAM.
- Para saber como conceder acesso a seus atributos para terceiros Contas da AWS, consulte Fornecimento de acesso a Contas da AWS pertencentes a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte Como os perfis do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Atualização de permissões do painel do CloudWatch

Em 1.º de maio de 2018, a AWS alterou as permissões necessárias para acessar painéis do CloudWatch. O acesso ao painel no console do CloudWatch agora necessita de permissões que foram apresentadas em 2017 para oferecer suporte a operações de API no painel:

- cloudwatch:GetDashboard
- cloudwatch:ListDashboards
- cloudwatch:PutDashboard
- cloudwatch:DeleteDashboards

Para acessar os painéis do CloudWatch, você precisa de um destes itens:

- A política AdministratorAccess.
- A política CloudWatchFullAccess.
- Uma política personalizada que inclui uma ou mais destas permissões específicas:
  - `cloudwatch:GetDashboard` e `cloudwatch:ListDashboards` para poder visualizar painéis
  - `cloudwatch:PutDashboard` para poder criar ou modificar painéis

- `cloudwatch:DeleteDashboards` para poder excluir painéis

Para obter mais informações sobre o uso de políticas para alterar as permissões de um usuário do IAM, consulte Alteração de permissões para um usuário do IAM.

Para obter mais informações sobre permissões do CloudWatch, consulte Referência de permissões do Amazon CloudWatch.

Para obter mais informações sobre as operações da API do painel, consulte PutDashboard na Referência da API do Amazon CloudWatch.

Políticas gerenciadas (predefinidas) pela AWS para o CloudWatch

A AWS aborda muitos casos de uso comuns fornecendo políticas autônomas do IAM que são criadas e administradas pela AWS. Essas políticas gerenciadas pela AWS concedem as permissões indispensáveis para casos de uso comuns para que você não precise investigar quais permissões são necessárias. Para mais informações, consulte Políticas gerenciadas pela AWS no Manual do usuário do IAM.

As seguintes políticas gerenciadas pelaAWS , que é possível associar a usuários na sua conta, são específicas do CloudWatch.

Tópicos

CloudWatchFullAccessV2

A AWS adicionou recentemente a política do IAM gerenciada CloudWatchFullAccessV2. Essa política concede acesso total às ações e recursos do CloudWatch, ao mesmo tempo em que define melhor o escopo das permissões concedidas para outros serviços, como o Amazon SNS e o Amazon EC2 Auto Scaling. Recomendamos que você comece a usar essa política em vez de usar CloudWatchFullAccess. A AWS planeja descontinuar o uso de CloudWatchFullAccess em um futuro próximo.

A política inclui permissões `application-signals:` para que os usuários possam acessar todas as funcionalidades do console do CloudWatch no Application Signals. Isso inclui algumas permissões `autoscaling:Describe` para que os usuários com essa política possam ver as ações do Auto Scaling associadas aos alarmes do CloudWatch. Isso inclui algumas permissões `sns` para que os usuários com essa política possam recuperar tópicos criados do Amazon SNS e associá-los a alarmes do CloudWatch. Inclui permissões do IAM para que os usuários com essa política possam visualizar informações sobre perfis vinculados a serviço associados ao CloudWatch. Inclui as permissões `oam:ListSinks` e `oam:ListAttachedLinks` para que os usuários com essa política possam usar o console para visualizar dados compartilhados das contas de origem na observabilidade entre contas do CloudWatch.

Inclui as permissões `rum`, `synthetics` e `xray` para que os usuários possam ter acesso total ao CloudWatch Synthetics, ao AWS X-Ray e ao CloudWatch RUM usando o serviço CloudWatch.

O conteúdo da política CloudWatchFullAccessV2 é o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchFullAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalingPolicies",
                "application-signals:*",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribePolicies",
                "cloudwatch:*",
                "logs:*",
                "sns:CreateTopic",
                "sns:ListSubscriptions",
                "sns:ListSubscriptionsByTopic",
```

```
                "sns:ListTopics",
                "sns:Subscribe",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "oam:ListSinks",
                "rum:*",
                "synthetics:*",
                "xray:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsServiceLinkedRolePermissions",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "application-
signals.cloudwatch.amazonaws.com"
                }
            }
        },
        {
            "Sid": "EventsServicePermissions",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/events.amazonaws.com/
AWSServiceRoleForCloudWatchEvents*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "events.amazonaws.com"
                }
            }
        },
         {
            "Sid": "OAMReadPermissions",
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
```

```
        }
    ]
}
```

CloudWatchFullAccess

A política CloudWatchFullAccess está prestes a ser descontinuada. Recomendamos que você pare de usá-la e use CloudWatchFullAccessV2 em seu lugar.

O conteúdo da política CloudWatchFullAccess é o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:Describe*",
                "cloudwatch:*",
                "logs:*",
                "sns:*",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "oam:ListSinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/events.amazonaws.com/
AWSServiceRoleForCloudWatchEvents*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "events.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
```

```
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        }
    ]
}
```

CloudWatchReadOnlyAccess

A política CloudWatchReadOnlyAccess concede acesso somente leitura ao CloudWatch.

A política inclui algumas permissões `logs:`, portanto, os usuários com essa política podem usar o console para visualizar as informações do CloudWatch Logs e as consultas do CloudWatch Logs Insights. Ela inclui `autoscaling:Describe*` para que os usuários com essa política possam visualizar as ações do Auto Scaling associadas aos alarmes do CloudWatch. A política inclui permissões `application-signals:` para que os usuários possam usar o Application Signals para monitorar a integridade dos serviços. Ela inclui `application-autoscaling:DescribeScalingPolicies` para que os usuários com essa política possam acessar informações sobre as políticas do Application Auto Scaling. Ela inclui `sns:Get*` e `sns:List*` para que os usuários com essa política possam recuperar informações sobre os tópicos do Amazon SNS que recebem notificações sobre alarmes do CloudWatch. Ela inclui as permissões `oam:ListSinks` e `oam:ListAttachedLinks` para que os usuários com essa política possam usar o console para visualizar os dados compartilhados de contas de origem na observabilidade entre contas do CloudWatch. Ela inclui as permissões `iam:GetRole` para que os usuários possam verificar se o CloudWatch Application Signals foi configurado.

Ela inclui permissões `rum`, `synthetics` e `xray` para que os usuários possam ter acesso somente leitura ao CloudWatch Synthetics, ao AWS X-Ray e ao CloudWatch RUM usando o serviço CloudWatch.

A seguir, temos o conteúdo da política CloudWatchReadOnlyAccess.

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchReadOnlyAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalingPolicies",
                "application-signals:BatchGet*",
                "application-signals:Get*",
```

```
                "application-signals:List*",
                "autoscaling:Describe*",
                "cloudwatch:BatchGet*",
                "cloudwatch:Describe*",
                "cloudwatch:GenerateQuery",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:Describe*",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "oam:ListSinks",
                "sns:Get*",
                "sns:List*",
                "rum:BatchGet*",
                "rum:Get*",
                "rum:List*",
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*",
                "xray:BatchGet*",
                "xray:Get*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "OAMReadPermissions",
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        },
        {
            "Sid": "CloudWatchReadOnlyGetRolePermissions",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
```

```
        }
     ]
}
```

CloudWatchActionsEC2Access

A política CloudWatchActionsEC2Access concede acesso somente leitura aos alarmes e às métricas do CloudWatch, bem como aos metadados do Amazon EC2. Também concede acesso às ações da API de interromper, terminar e reinicializar para instâncias do EC2.

A seguir, temos o conteúdo da política CloudWatchActionsEC2Access

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:Describe*",
        "ec2:Describe*",
        "ec2:RebootInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

CloudWatchAutomaticDashboardsAccess

A política gerenciada CloudWatch-CrossAccountAccess é usada pela função do IAM CloudWatch-CrossAccountSharingRole. Essa função e política permitem que os usuários de painéis entre contas visualizem painéis automáticos em cada conta que está compartilhando painéis.

Este é o conteúdo da política CloudWatchAutomaticDashboardsAccess:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
```

```
        "autoscaling:DescribeAutoScalingGroups",
        "cloudfront:GetDistribution",
        "cloudfront:ListDistributions",
        "dynamodb:DescribeTable",
        "dynamodb:ListTables",
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListServices",
        "elasticache:DescribeCacheClusters",
        "elasticbeanstalk:DescribeEnvironments",
        "elasticfilesystem:DescribeFileSystems",
        "elasticloadbalancing:DescribeLoadBalancers",
        "kinesis:DescribeStream",
        "kinesis:ListStreams",
        "lambda:GetFunction",
        "lambda:ListFunctions",
        "rds:DescribeDBClusters",
        "rds:DescribeDBInstances",
        "resource-groups:ListGroupResources",
        "resource-groups:ListGroups",
        "route53:GetHealthCheck",
        "route53:ListHealthChecks",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "sns:ListTopics",
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl",
        "sqs:ListQueues",
        "synthetics:DescribeCanariesLastRun",
        "tag:GetResources"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "apigateway:GET"
      ],
      "Effect": "Allow",
      "Resource": [
```

```
        "arn:aws:apigateway:*::/restapis*"
      ]
    }
  ]
```

Política: CloudWatchAgentServerPolicy

A política CloudWatchAgentServerPolicy pode ser usada em funções do IAM anexadas às instâncias do Amazon EC2 para permitir que o atendente do CloudWatch leia informações da instância e as grave no CloudWatch. Contém o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CWACloudWatchServerPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ec2:DescribeVolumes",
                "ec2:DescribeTags",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CWASSMServerPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-*"
        }
```

```
    ]
}
```

CloudWatchAgentAdminPolicy

A política CloudWatchAgentAdminPolicy pode ser usada em funções do IAM anexadas às instâncias do Amazon EC2. Essa política permite que o atendente do CloudWatch leia informações da instância, grave-as no CloudWatch e grave informações no Parameter Store. Contém o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CWACloudWatchPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ec2:DescribeTags",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CWASSMPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-*"
        }
    ]
```

```
}
```

> **Note**
>
> É possível analisar essas políticas de permissões fazendo login no console do IAM e pesquisando políticas específicas.

Você também pode criar as próprias políticas do IAM personalizadas a fim de conceder permissões para ações e recursos do CloudWatch. Você pode anexar essas políticas personalizadas a usuários ou grupos do IAM que exijam essas permissões.

Políticas gerenciadas da (predefinidas) da AWS para a observabilidade entre contas do CloudWatch

As políticas dessa seção concedem permissões relacionadas à observabilidade entre contas do CloudWatch. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.

CloudWatchCrossAccountSharingConfiguration

A política CloudWatchCrossAccountSharingConfiguration dá acesso para criar, gerenciar e exibir links do Observability Access Manager para compartilhar recursos do CloudWatch entre contas. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch. Contém o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Link",
                "oam:ListLinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
```

```
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        }
    ]
}
```

OAMFullAccess

A política OAMFullAccess dá acesso para criar, gerenciar e visualizar coletores e links do Observability Access Manager, que são utilizadas para a observabilidade entre contas do CloudWatch.

A política OAMFullAccess por si só não permite compartilhar dados de observabilidade entre links. Para criar um link para compartilhar métricas do CloudWatch, você também precisa da política CloudWatchFullAccess ou CloudWatchCrossAccountSharingConfiguration. Para criar um link para compartilhar grupos de logs do CloudWatch Logs, você também precisa da política CloudWatchLogsFullAccess ou CloudWatchLogsCrossAccountSharingConfiguration. Para criar um link para compartilhar rastreamentos do X-Ray, você também precisa da política AWSXRayFullAccess ou AWSXRayCrossAccountSharingConfiguration.

Para ter mais informações, consulte Observabilidade entre contas do CloudWatch. Contém o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oam:*"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

OAMReadOnlyAccess

A política OAMReadOnlyAccess dá acesso somente leitura aos recursos do Observability Access Manager, que são utilizados para a observabilidade entre contas do CloudWatch. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch. Contém o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oam:Get*",
                "oam:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Políticas gerenciadas (predefinidas) pela AWS para o CloudWatch Synthetics

As políticas CloudWatchSyntheticsFullAccess e CloudWatchSyntheticsReadOnlyAccess gerenciadas pela AWS estão disponíveis para você atribuir aos usuários que vão gerenciar ou usar CloudWatch Synthetics. Estas políticas adicionais também são relevantes:

- AmazonS3ReadOnlyAccess e CloudWatchReadOnlyAccess: são necessárias para poder ler todos os dados do Synthetics no console do CloudWatch.
- AWSLambdaReadOnlyAccess: para ser capaz de visualizar o código-fonte usado pelos canaries.
- CloudWatchSyntheticsFullAccess permite que você crie canaries. Além disso, para criar e excluir canaries que tenham uma nova função do IAM criada para eles, você também precisa da seguinte instrução de política em linha:

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:CreatePolicy",
                "iam:DeletePolicy",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy",
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

Important

A concessão das permissões `iam:CreateRole`, `iam:DeleteRole`, `iam:CreatePolicy`, `iam:DeletePolicy`, `iam:AttachRolePolicy` e `iam:DetachRolePolicy` a um usuário proporciona acesso administrativo total para esse usuário criar, anexar e excluir funções e políticas que tenham ARNs correspondentes a `arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*` e `arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*`. Por exemplo, um usuário com essas permissões pode criar uma política com permissões completas para todos os recursos e anexar essa política a qualquer função correspondente ao padrão de ARN. Seja muito cuidadoso a quem você concede essas permissões.

Para obter informações sobre como anexar políticas e conceder permissões a usuários, consulte Alterar permissões para um usuário do IAM e Incorporar uma política em linha para um usuário ou para uma função.

CloudWatchSyntheticsFullAccess

A seguir está o conteúdo da política CloudWatchSyntheticsFullAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutEncryptionConfiguration"
            ],
            "Resource": [
                "arn:aws:s3:::cw-syn-results-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles",
                "s3:ListAllMyBuckets",
                "xray:GetTraceSummaries",
                "xray:BatchGetTraces",
                "apigateway:GET"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
```

```
            ],
            "Resource": "arn:aws:s3:::cw-syn-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::aws-synthetics-library-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "lambda.amazonaws.com",
                        "synthetics.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:ListAttachedRolePolicies"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:CreateFunction",
                "lambda:AddPermission",
                "lambda:PublishVersion",
                "lambda:UpdateFunctionCode",
                "lambda:UpdateFunctionConfiguration",
                "lambda:GetFunctionConfiguration",
                "lambda:DeleteFunction"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:cwsyn-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:GetLayerVersion",
                "lambda:PublishLayerVersion",
                "lambda:DeleteLayerVersion"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:layer:cwsyn-*",
```

```
                "arn:aws:lambda:*:*:layer:Synthetics:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": [
                "arn:*:sns:*:*:Synthetics-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:*:*:key/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:*:*:key/*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "s3.*.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

CloudWatchSyntheticsReadOnlyAccess

A seguir está o conteúdo da política CloudWatchSyntheticsReadOnlyAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

Políticas gerenciadas (predefinidas) da AWS para o Amazon CloudWatch RUM

As políticas gerenciadas da AWSAmazonCloudWatchRUMFullAccess e AmazonCloudWatchRUMReadOnlyAccess estão disponíveis para que você as atribua a usuários que gerenciarão ou usarão o CloudWatch RUM.

AmazonCloudWatchRUMFullAccess

Veja a seguir o conteúdo da política AmazonCloudWatchRUMFullAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rum:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-service-role/rum.amazonaws.com/
AWSServiceRoleForRealUserMonitoring"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/RUM-Monitor*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "cognito-identity.amazonaws.com"
```

```
                ]
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "cloudwatch:GetMetricData",
            "cloudwatch:GetMetricStatistics",
            "cloudwatch:ListMetrics"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "cloudwatch:DescribeAlarms"
        ],
        "Resource": "arn:aws:cloudwatch:*:*:alarm:*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "cognito-identity:CreateIdentityPool",
            "cognito-identity:ListIdentityPools",
            "cognito-identity:DescribeIdentityPool",
            "cognito-identity:GetIdentityPoolRoles",
            "cognito-identity:SetIdentityPoolRoles"
        ],
        "Resource": "arn:aws:cognito-identity:*:*:identitypool/*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogGroup",
            "logs:DeleteLogGroup",
            "logs:PutRetentionPolicy",
            "logs:CreateLogStream"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:*RUMService*"
    },
    {
        "Effect": "Allow",
        "Action": [
```

```
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": "arn:aws:logs:*:*:log-group::log-stream:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:describeCanaries",
                "synthetics:describeCanariesLastRun"
            ],
            "Resource": "arn:aws:synthetics:*:*:canary:*"
        }
    ]
}
```

AmazonCloudWatchRUMReadOnlyAccess

Veja a seguir o conteúdo da política AmazonCloudWatchRUMReadOnlyAccess.

```
{

 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "rum:GetAppMonitor",
    "rum:GetAppMonitorData",
    "rum:ListAppMonitors",
    "rum:ListRumMetricsDestinations",
    "rum:BatchGetRumMetricDefinitions"
   ],
```

```
    "Resource": "*"
   }
  ]
}
```

AmazonCloudWatchRUMServiceRolePolicy

Não é possível anexar o AmazonCloudWatchRUMServiceRolePolicy em suas entidades do IAM. Essa política é anexada a uma função vinculada a serviços que permite que o CloudWatch RUM publique dados de monitoramento para outros serviços relevantes da AWS. Para obter mais informações sobre essa função vinculada ao serviço, consulte Usar funções vinculadas ao serviço para o CloudWatch RUM.

Este é o conteúdo completo da política AmazonCloudWatchRUMServiceRolePolicy.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "xray:PutTraceSegments"
   ],
   "Resource": [
    "*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "cloudwatch:namespace": [
      "RUM/CustomMetrics/*",
      "AWS/RUM"
     ]
    }
   }
  }
 ]
}
```

Políticas gerenciadas (predefinidas) pela AWS para o CloudWatch Evidently

As políticas CloudWatchSyntheticsFullAccess e CloudWatchSyntheticsReadOnlyAccess gerenciadas pela AWS estão disponíveis para você atribuir aos usuários que vão gerenciar ou usar o CloudWatch Evidently.

CloudWatChevidentlyFullAccess

A seguir está o conteúdo da política CloudWatchEvidentlyFullAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "evidently:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchRUMEvidentlyRole-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:ListTagsForResource"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms",
                "cloudwatch:TagResource",
                "cloudwatch:UnTagResource"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudtrail:LookupEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:Evidently-Alarm-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": [
                "arn:*:sns:*:*:Evidently-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

CloudWatchEvidentlyReadOnlyAccess

A seguir está o conteúdo da política CloudWatchEvidentlyReadOnlyAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "evidently:GetExperiment",
                "evidently:GetFeature",
                "evidently:GetLaunch",
                "evidently:GetProject",
                "evidently:GetSegment",
                "evidently:ListExperiments",
                "evidently:ListFeatures",
```

```
                "evidently:ListLaunches",
                "evidently:ListProjects",
                "evidently:ListSegments",
                "evidently:ListSegmentReferencs"
            ],
            "Resource": "*"
        }
    ]
}
```

Política gerenciada pela AWS para o AWS Systems Manager Incident Manager

A política AWSCloudWatchAlarms_ActionSSMIncidentsServiceRolePolicy está anexada a uma função vinculada ao serviço que permite que o CloudWatch inicie incidentes no AWSSystems Manager Incident Manager em seu nome. Para ter mais informações, consulte Permissões de função vinculada ao serviço para ações do Systems Manager Incident Manager de alarmes do CloudWatch.

A política tem a seguinte permissão:

- ssm-incidents:StartIncident

Exemplos de política gerenciada pelo cliente

Nesta seção, você encontrará exemplos de políticas de usuário que concedem permissões para diversas ações do CloudWatch. Essas políticas funcionam quando você está usando a API do CloudWatch, os SDKs da AWS ou a AWS CLI.

Exemplos

- Exemplo 1: Permitir acesso total ao CloudWatch
- Exemplo 2: Permitir acesso somente leitura ao CloudWatch
- Exemplo 3: Interromper ou encerrar uma instância do instância do Amazon EC2

Exemplo 1: Permitir acesso total ao CloudWatch

Para conceder a um usuário acesso total ao CloudWatch, você pode usar a política gerenciada CloudWatchFullAccess em vez de criar uma política gerenciada pelo cliente. O conteúdo de CloudWatchFullAccess está listado em CloudWatchFullAccess.

Exemplo 2: Permitir acesso somente leitura ao CloudWatch

A política a seguir permite a um usuário acesso somente leitura ao CloudWatch e visualização de ações do Amazon EC2 Auto Scaling, métricas do CloudWatch, dados do CloudWatch Logs e dados do Amazon SNS relacionados a alarmes.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:Describe*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:List*",
        "logs:Get*",
        "logs:Describe*",
        "logs:StartQuery",
        "logs:StopQuery",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "logs:StartLiveTail",
        "logs:StopLiveTail",
        "sns:Get*",
        "sns:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Exemplo 3: Interromper ou encerrar uma instância do instância do Amazon EC2

A política a seguir permite que uma ação de alarme do CloudWatch interrompa ou encerre uma instância do EC2. No exemplo abaixo, as ações GetMetricData, ListMetrics e DescribeAlarms são opcionais. Recomendamos que você inclua essas ações para interromper ou encerrar corretamente a instância.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Action": [
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

Atualização do CloudWatch para políticas gerenciadas pela AWS

Veja detalhes sobre atualizações em políticas gerenciadas pela AWS para o CloudWatch desde que esse serviço começou a monitorar essas alterações. Para obter alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed RSS na página Document history (Histórico de documentos) do CloudWatch.

Alteração	Descrição	Data
CloudWatchFullAccessV2: atualização para uma política existente	O CloudWatch atualizou a política chamada CloudWatchFullAccessV2.	20 de maio de 2024

Alteração	Descrição	Data
	O escopo da política `CloudWatchFullAccessPermissions` foi atualizado para adicionar `application-signals:*` com a finalidade de que os usuários possam usar o CloudWatch Application Signals para visualizar, investigar e diagnosticar problemas relacionados com a integridade de seus serviços.	

Alteração	Descrição	Data
CloudWatchReadOnlyAccess – Atualização de política existente	O CloudWatch atualizou a política chamada CloudWatchReadOnlyAccess. O escopo da política `CloudWatchReadOnlyAccessPermissions` foi atualizado para adicionar `application-signals:BatchGet*` , `application-signals:List*` e `application-signals:Get*` com a finalidade de que os usuários possam usar o CloudWatch Application Signals para visualizar, investigar e diagnosticar problemas relacionados com a integridade de seus serviços. O escopo de `CloudWatchReadOnlyGetRolePermissions` foi atualizado para adicionar a ação `iam:GetRole` com a finalidade de que os usuários possam verificar se o CloudWatch Application Signals está configurado.	20 de maio de 2024

Alteração	Descrição	Data
CloudWatchApplicationSignalsServiceRolePolicy: atualização para uma política existente	O CloudWatch atualizou a política denominada CloudWatchApplicationSignalsServiceRolePolicy. O escopo das permissões `logs:StartQuery` e `logs:GetQueryResults` foi alterado para adicionar os ARNs `arn:aws:logs:*:*:log-group:/aws/appsignals/*:*` e `arn:aws:logs:*:*:log-group:/aws/application-signals/data:*` com a finalidade de habilitar o Application Signals em mais arquiteturas.	18 de abril de 2024
CloudWatchApplicationSignalsServiceRolePolicy: atualização para uma política existente	O CloudWatch alterou o escopo de uma permissão em CloudWatchApplicationSignalsServiceRolePolicy. O escopo da permissão `cloudwatch:GetMetricData` foi alterado para `*` com a finalidade de que o Application Signals possa recuperar métricas de origens em contas vinculadas.	8 de abril de 2024

Alteração	Descrição	Data
CloudWatchAgentServerPolicy : atualização para uma política existente	O CloudWatch adicionou permissões à CloudWatchAgentServerPolicy. As permissões `xray:PutTraceSegments` , `xray:PutTelemetryRecords` , `xray:GetSamplingRules` , `xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` e `logs:PutRetentionPolicy` foram adicionadas para que o agente do CloudWatch possa publicar rastreamentos do X-Ray e modificar os períodos de retenção do grupo de logs.	12 de fevereiro de 2024

Alteração	Descrição	Data
CloudWatchAgentAdminPolicy: atualização para uma política existente	O CloudWatch adicionou permissões à CloudWatchAgentAdminPolicy. As permissões `xray:PutTraceSegments` , `xray:PutTelemetryRecords` , `xray:GetSamplingRules` , `xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` e `logs:PutRetentionPolicy` foram adicionadas para que o agente do CloudWatch possa publicar rastreamentos do X-Ray e modificar os períodos de retenção do grupo de logs.	12 de fevereiro de 2024

Alteração	Descrição	Data
CloudWatchFullAccessV2: atualização para uma política existente	O CloudWatch adicionou permissões para CloudWatchFullAccessV2. Permissões existentes para as ações do CloudWatch Synthetics, do X-Ray e do CloudWatch RUM e novas permissões para o CloudWatch Application Signals foram adicionadas com a finalidade de que os usuários com essa política possam gerenciar o CloudWatch Application Signals. A permissão para a criação de um perfil vinculado ao serviço CloudWatch Application Signals foi adicionada com a finalidade de permitir que o CloudWatch Application Signals descubra dados de telemetria em logs, métricas, rastreamentos e tags.	5 de dezembro de 2023

Alteração	Descrição	Data
CloudWatchReadOnlyAccess – Atualização de política existente	O CloudWatch adicionou permissões ao CloudWatchReadOnlyAccess. Permissões somente leitura existentes para as ações do CloudWatch Synthetics, do X-Ray e do CloudWatch RUM e novas permissões somente leitura para o CloudWatch Application Signals foram adicionadas com a finalidade de que os usuários com essa política possam realizar a triagem e avaliar os problemas de integridade de serviço, conforme relatado pelo CloudWatch Application Signals. A permissão `cloudwatch:GenerateQuery` foi adicionada para que os usuários com essa política possam gerar uma string de consulta do CloudWatch Metrics Insights de uma solicitação em linguagem natural.	5 de dezembro de 2023

Alteração	Descrição	Data
CloudWatchApplicationSignalsServiceRolePolicy: nova política	O CloudWatch adicionou uma nova política CloudWatchApplicationSignalsServiceRolePolicy. O CloudWatchApplicationSignalsServiceRolePolicy concede a um futuro recurso permissões para coletar dados de registros do CloudWatch, dados de rastreamento do X-Ray, dados de métricas do CloudWatch e dados de marcação.	9 de novembro de 2023
AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy: nova política	O CloudWatch adicionou uma nova política AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy. AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy concede permissão ao CloudWatch para buscar métricas do Insights de Performance de bancos de dados em seu nome.	20 de setembro de 2023

Alteração	Descrição	Data
CloudWatchReadOnlyAccess – Atualização de política existente	O CloudWatch adicionou uma permissão a CloudWatchReadOnlyAccess. A permissão `application-autoscaling:DescribeScalingPolicies` foi adicionada de forma que os usuários com essa política possam acessar informações sobre as políticas do Application Auto Scaling.	14 de setembro de 2023
CloudWatchFullAccessV2: nova política	O CloudWatch adicionou uma nova política: CloudWatchFullAccessV2. CloudWatchFullAccessV2 concede acesso total às ações e recursos do CloudWatch, ao mesmo tempo em que define melhor o escopo das permissões concedidas a outros serviços, como o Amazon SNS e o Amazon EC2 Auto Scaling. Para obter mais informações, consulte CloudWatchFullAccessV2.	1º de agosto de 2023

Alteração	Descrição	Data
AWSServiceRoleForInternetMonitor: atualização de uma política existente	O Monitor de Internet do Amazon CloudWatch adicionou novas permissões para monitorar recursos do Network Load Balancer. As permissões `elasticloadbalancing:DescribeLoadBalancers` e `ec2:DescribeNetworkInterfaces` são necessárias para que o Monitor de Internet possa monitorar o tráfego do Network Load Balancer dos clientes analisando os logs de fluxo dos recursos do NLB. Para ter mais informações, consulte Uso do Monitor de Internet do Amazon CloudWatch.	15 de julho de 2023

Alteração	Descrição	Data
CloudWatchReadOnlyAccess – Atualização de política existente	O CloudWatch adicionou permissões ao CloudWatchReadOnlyAccess. As permissões `logs:StartLiveTail` e `logs:StopLiveTail` foram adicionadas para que os usuários com essa política possam usar o console para iniciar e interromper as sessões de teste ao vivo do CloudWatch Logs. Para obter mais informações, consulte Usar o Live Tail para visualizar registros quase em tempo real.	6 de junho de 2023
CloudWatchCrossAccountSharingConfiguration – Nova política	O CloudWatch adicionou uma nova política para possibilitar o gerenciamento dos links de observabilidade entre contas do CloudWatch que compartilham métricas do CloudWatch. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.	27 de novembro de 2022

Alteração	Descrição	Data
OAMFullAccess – Nova política	O CloudWatch adicionou uma nova política para possibilitar o gerenciamento completo dos links e coletores de observabilidade entre contas do CloudWatch. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.	27 de novembro de 2022
OAMReadOnlyAccess – Nova política	O CloudWatch adicionou uma nova política para possibilitar que você visualize informaçõ es sobre links e coletores de observabilidade entre contas do CloudWatch. Para ter mais informações, consulte Observabilidade entre contas do CloudWatch.	27 de novembro de 2022
CloudWatchFullAccess – Atualização de política existente	O CloudWatch adicionou permissões a CloudWatc hFullAccess. As permissões `oam:ListS inks` e `oam:ListA ttachedLinks` foram adicionadas para que os usuários com essa política possam usar o console para visualizar dados compartil hados das contas de origem na observabilidade entre contas do CloudWatch.	27 de novembro de 2022

Alteração	Descrição	Data
CloudWatchReadOnlyAccess – Atualização de política existente	O CloudWatch adicionou permissões ao CloudWatchReadOnlyAccess. As permissões `oam:ListSinks` e `oam:ListAttachedLinks` foram adicionadas para que os usuários com essa política possam usar o console para visualizar dados compartilhados das contas de origem na observabilidade entre contas do CloudWatch.	27 de novembro de 2022

Alteração	Descrição	Data
AmazonCloudWatchRUMServiceRolePolicy - Atualização de uma política existente	O CloudWatch RUM atualizou uma chave de condição em AmazonCloudWatchRUMServiceRolePolicy. A chave de condição `"Condition": { "StringEquals": { "cloudwatch:namespace": "AWS/RUM" } }` foi alterada para a seguinte, de forma que o CloudWatch RUM possa enviar métricas personalizadas para namespaces de métricas personalizadas. `"Condition": { "StringLike": { "cloudwatch:namespace": [ "RUM/CustomMetrics/*", "AWS/RUM" ] } }`	2 de fevereiro de 2023

Alteração	Descrição	Data
AmazonCloudWatchRUMReadOnlyAccess - Política atualizada	O CloudWatch adicionou permissões à política AmazonCloudWatchRUMReadOnlyAccess. As permissões `rum:ListRumMetricsDestinations` e `rum:BatchGetRumMetricsDefinitions` foram adicionadas para que o CloudWatch RUM possa enviar métricas estendidas ao CloudWatch e ao Evidently.	27 de outubro de 2022
AmazonCloudWatchRUMServiceRolePolicy - Atualização de uma política existente	O CloudWatch RUM adicionou permissões à AmazonCloudWatchRUMServiceRolePolicy. A permissão `cloudwatch:PutMetricData` foi adicionada para que o CloudWatch RUM possa enviar métricas estendidas para o CloudWatch.	26 de outubro de 2022

Alteração	Descrição	Data
CloudWatchEvidentlyReadOnlyAccess: atualizar para uma política existente	O CloudWatch Evidently adicionou permissões para CloudWatchEvidentlyReadOnlyAccess. Foram adicionadas as permissões `evidently:GetSegment` , `evidently:ListSegments` e `evidently:ListSegmentReferences` para que os usuários com esta política possam ver os segmentos de público do Evidently que foram criados.	12 de agosto de 2022
CloudWatchSyntheticsFullAccess: atualização para uma política existente	O CloudWatch Synthetics adicionou permissões para CloudWatchSyntheticsFullAccess. As permissões `lambda:DeleteFunction` e `lambda:DeleteLayerVersion` foram adicionadas para que o CloudWatch Synthetics possa excluir recursos relacionados quando um canário for excluído. `iam:ListAttachedRolePolicies` foi adicionada para que os clientes possam visualizar as políticas que estão anexadas à função do IAM de um canário.	6 de maio de 2022

Alteração	Descrição	Data
AmazonCloudWatchRUMFullAccess: nova política	O CloudWatch adicionou uma nova política para permitir o gerenciamento completo do CloudWatch RUM. O CloudWatch RUM permite que você execute o monitoramento real do usuário de sua aplicação Web. Para ter mais informações, consulte Usar o CloudWatch RUM.	29 de novembro de 2021
AmazonCloudWatchRUMReadOnlyAccess: nova política	O CloudWatch adicionou uma nova política para permitir o acesso somente leitura ao CloudWatch RUM. O CloudWatch RUM permite que você execute o monitoramento real do usuário de sua aplicação Web. Para ter mais informações, consulte Usar o CloudWatch RUM.	29 de novembro de 2021

Alteração	Descrição	Data
CloudWatchEvidentlyFullAccess: nova política	O CloudWatch adicionou uma nova política para permitir o gerenciamento completo do CloudWatch Evidently. O CloudWatch Evidently permite que você execute experimentos A/B de suas aplicações Web e os implemente gradualmente. Para ter mais informaçõ es, consulte Execução de lançamentos e experimen tos A/B com o CloudWatch Evidently.	29 de novembro de 2021
CloudWatchEvidentl yReadOnlyAccess: nova política	O CloudWatch adicionou uma nova política para permitir o acesso somente leitura ao CloudWatch Evidently. O CloudWatch Evidently permite que você execute experimentos A/B de suas aplicações Web e os implemente gradualmente. Para ter mais informaçõ es, consulte Execução de lançamentos e experimen tos A/B com o CloudWatch Evidently.	29 de novembro de 2021

Alteração	Descrição	Data
AWSServiceRoleForCloudWatchRUM: nova política gerenciada	O CloudWatch adicionou uma política voltada para uma nova função vinculada a serviços para permitir que o CloudWatch RUM substitua dados de monitoramento para outros serviços relevantes da AWS.	29 de novembro de 2021

Alteração	Descrição	Data
CloudWatchSyntheticsFullAccess: atualização para uma política existente	O CloudWatch Synthetics adicionou permissões ao CloudWatchSyntheticsFullAccess, e também alterou o escopo de uma permissão. A permissão `kms:ListAliases` foi adicionada para que os usuários possam listar chaves KMS AWS KMS que podem ser usadas para criptografar artefatos do canário. A permissão `kms:DescribeKey` foi adicionada para que os usuários possam ver os detalhes das chaves que serão usadas para criptografar artefatos do canário. E a permissão `kms:Decrypt` foi adicionada para permitir que os usuários descriptografassem artefatos do canário. Essa capacidade de descriptografia é limitada ao uso em recursos dentro de buckets do Amazon S3. O escopo do `Resource` da permissão `s3:GetBucketLocation` foi alterado de `*` para `arn:aws:s3:::*` .	29 de setembro de 2021

Alteração	Descrição	Data
CloudWatchSyntheticsFullAccess: atualização para uma política existente	O CloudWatch Synthetics adicionou uma permissão para CloudWatchSyntheticsFullAccess. A permissão `lambda:UpdateFunctionCode` foi adicionada para que os usuários com esta política possam alterar a versão de runtime dos canaries.	20 de julho de 2021
AWSCloudWatchAlarms_ActionSSMIncidentsServiceRolePolicy: nova política gerenciada	O CloudWatch adicionou uma nova política do IAM gerenciada para permitir que o CloudWatch crie incidentes no AWS Systems Manager Incident Manager.	10 de maio de 2021
CloudWatchAutomaticDashboardsAccess: atualização para uma política existente	O CloudWatch adicionou uma permissão à política gerenciada CloudWatchAutomaticDashboardsAccess. A permissão `synthetics:DescribeCanariesLastRun` foi adicionada a esta política para permitir que os usuários de painel entre contas visualizem detalhes sobre execuções do canário do CloudWatch Synthetics.	20 de abril de 2021

Alteração	Descrição	Data
O CloudWatch começou a monitorar alterações	O CloudWatch começou a rastrear alterações para as políticas gerenciadas pela AWS.	14 de abril de 2021

Usar chaves de condição para limitar o acesso a namespaces do CloudWatch

Use chaves de condição do IAM para limitar o usuário a publicar métricas somente nos namespaces do CloudWatch especificados.

Permitir a publicação em apenas um namespace

A política a seguir limita o usuário a publicar métricas somente no namespace chamado `MyCustomNamespace`.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Resource": "*",
        "Action": "cloudwatch:PutMetricData",
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": "MyCustomNamespace"
            }
        }
    }
}
```

Excluir publicação de um namespace

A política a seguir permite que o usuário publique métricas em qualquer namespace, exceto para `CustomNamespace2`.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData"
        },
        {
            "Effect": "Deny",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CustomNamespace2"
                }
            }
        }
    ]
}
```

Usar chaves de condição para limitar o acesso dos usuários do Contributor Insights aos grupos de log

Para criar uma regra no Contributor Insights e visualizar seus resultados, o usuário deve ter a permissão `cloudwatch:PutInsightRule`. Por padrão, um usuário com essa permissão pode criar uma regra do Contributor Insights que avalia qualquer grupo de log no CloudWatch Logs e ver os resultados. Os resultados podem conter dados de colaborador para esses grupos de log.

É possível criar políticas do IAM com chaves de condição para conceder aos usuários a permissão para gravar regras do Contributor Insights para alguns grupos de log, impedindo que eles gravem regras e visualizem esses dados de outros grupos de log.

Para obter mais informações sobre o elemento `Condition` em políticas do IAM, consulte Elementos de políticas JSON do IAM: condição.

Permitir acesso para gravar regras e exibir resultados apenas a determinados grupos de logs

A política a seguir concede ao usuário acesso para gravar regras e exibir resultados para o grupo de logs chamado `AllowedLogGroup` e todos os grupos de logs que têm nomes começados com `AllowedWildCard`. Não concede acesso para gravar regras ou exibir resultados de regra para quaisquer outros grupos de log.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCertainLogGroups",
            "Effect": "Allow",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*",
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
                    "cloudwatch:requestInsightRuleLogGroups": [
                        "AllowedLogGroup",
                        "AllowedWildcard*"
                    ]
                }
            }
        }
    ]
}
```

Negar gravação de regras para grupos específicos de logs, mas permitir a gravação de regras para todos os outros grupos de logs

A política a seguir nega explicitamente o acesso do usuário para gravar regras e exibir resultados de regra para o grupo de log chamado `ExplicitlyDeniedLogGroup`, mas permite gravar regras e exibir resultados de regra para todos os outros grupos de log.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowInsightRulesOnLogGroupsByDefault",
            "Effect": "Allow",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*"

        },
        {
            "Sid": "ExplicitDenySomeLogGroups",
            "Effect": "Deny",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*",
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
```

```
                    "cloudwatch:requestInsightRuleLogGroups": [
                        "/test/alpine/ExplicitlyDeniedLogGroup"
                    ]
                }
            }
        }
    ]
}
```

Usar chaves de condição para limitar as ações de alarme

Quando os alarmes do CloudWatch mudam de estado, podem executar ações diferentes, como interromper e terminar instâncias do EC2 e executar ações do Systems Manager. Essas ações podem ser iniciadas quando o alarme muda para qualquer estado, inclusive ALARM, OK ou INSUFFICIENT_DATA.

Usar a chave de condição `cloudwatch:AlarmActions` para permitir que um usuário crie alarmes que só possam executar as ações que você especificar quando o estado do alarme mudar. Por exemplo, é possível permitir que um usuário crie alarmes que só possam executar ações que não sejam ações do EC2.

Permitir que um usuário crie alarmes que possam enviar apenas notificações do Amazon SNS ou executar ações do Systems Manager

A política a seguir limita o usuário a criar alarmes que possam apenas enviar notificações do Amazon SNS e executar ações do Systems Manager. O usuário não pode criar alarmes que executem ações do EC2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateAlarmsThatCanPerformOnlySNSandSSMActions",
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricAlarm",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringLike": {
                    "cloudwatch:AlarmActions": [
                        "arn:aws:sns:*",
                        "arn:aws:ssm:*"
```

```
                    ]
                }
            }
        }
    ]
}
```

Usar funções vinculadas ao serviço para o CloudWatch

O Amazon CloudWatch usa funções vinculadas ao serviço do AWS Identity and Access Management (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao CloudWatch. As funções vinculadas ao serviço são predefinidas pelo CloudWatch e incluem todas as permissões que o serviço requer para chamar outros produtos da AWS em seu nome.

A função vinculada ao serviço do CloudWatch facilita as configurações de alarmes do CloudWatch que podem terminar, interromper ou reinicializar uma instância do Amazon EC2 sem a necessidade de adicionar as permissões necessárias manualmente. Outra função vinculada ao serviço permite que uma conta de monitoramento acesse dados do CloudWatch de outras contas especificadas por você, para criar painéis entre contas e entre regiões.

O CloudWatch define as permissões dessas funções vinculadas ao serviço e, a menos que definido de outra forma, somente o CloudWatch poderá assumir a função. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Você pode excluir os perfis somente depois de primeiro excluir seus recursos relacionados. Essa restrição protege seus recursos do CloudWatch, pois não é possível remover acidentalmente as permissões para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com funções vinculadas a serviços, consulte Serviços da AWS compatíveis com o IAM e procure os serviços que contenham Sim na coluna Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

Permissões de função vinculada ao serviço para ações do EC2 de alarmes do CloudWatch

O CloudWatch usa a função vinculada ao serviço chamada AWSServiceRoleForCloudWatchEvents: o CloudWatch usa essa função vinculada ao serviço para executar ações de alarmes do Amazon EC2.

A função vinculada ao serviço AWSServiceRoleForCloudWatchEvents se apoia no serviço do CloudWatch Events abaixo para assumir a função. O CloudWatch Events invoca as ações de terminar, interromper ou reinicializar as instâncias quando chamado pelo alarme.

A política de permissões de função vinculada ao serviço AWSServiceRoleForCloudWatchEvents permite que o CloudWatch Events conclua estas ações nas instâncias do Amazon EC2:

- `ec2:StopInstances`
- `ec2:TerminateInstances`
- `ec2:RecoverInstances`
- `ec2:DescribeInstanceRecoveryAttribute`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceStatus`

A política de permissões de função vinculada ao serviço AWSServiceRoleForCloudWatchCrossAccount permite que o CloudWatch conclua as seguintes ações:

- `sts:AssumeRole`

Permissões de perfis vinculados ao serviço para o CloudWatch Application Signals

O CloudWatch Application Signals usa o perfil vinculado ao serviço denominado AWSServiceRoleForCloudWatchApplicationSignals: o CloudWatch usa esse perfil vinculado ao serviço para coletar dados do CloudWatch Logs, dados de rastreamento do X-Ray, dados de métricas do CloudWatch e dados de marcação de aplicações que você habilitou para o CloudWatch Application Signals.

O perfil vinculado ao serviço AWSServiceRoleForCloudWatchApplicationSignals confia no CloudWatch Application Signals para assumir o perfil. O Application Signals coleta dados de logs, rastreamentos, métricas e tags usando sua conta.

O AWSServiceRoleForCloudWatchApplicationSignals possui uma política do IAM anexada, e essa política é denominada CloudWatchApplicationSignalsServiceRolePolicy. Essa política concede permissão ao CloudWatch Application Signals para coletar dados de monitoramento e de marcação de outros serviços da AWS relevantes. Ela inclui permissões para que o Application Signals conclua as seguintes ações:

- xray:GetServiceGraph
- logs:StartQuery
- logs:GetQueryResults
- cloudwatch:GetMetricData
- cloudwatch:ListMetrics
- tag:GetResources

O conteúdo completo da CloudWatchApplicationSignalsServiceRolePolicy é o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "XRayPermission",
            "Effect": "Allow",
            "Action": [
                "xray:GetServiceGraph"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWLogsPermission",
            "Effect": "Allow",
            "Action": [
                "logs:StartQuery",
                "logs:GetQueryResults"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/appsignals/*:*",
                "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
```

```
                }
            }
        },
        {
            "Sid": "CWListMetricsPermission",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWGetMetricDataPermission",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "TagsPermission",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        }
    ]
```

```
}
```

Permissões de função vinculada ao serviço para ações do Systems Manager OpsCenter de alarmes do CloudWatch

O CloudWatch usa a função vinculada ao serviço chamada AWSServiceRoleForCloudWatchAlarms_ActionSSM: o CloudWatch usa essa função vinculada ao serviço para executar ações do Systems Manager OpsCenter quando o alarme do CloudWatch passa para o estado ALARM.

A função vinculada ao serviço AWSServiceRoleForCloudWatchAlarms_ActionSSM se apoia no serviço do CloudWatch abaixo para assumir a função. Os alarmes do CloudWatch invocam as ações do Systems Manager OpsCenter quando chamado pelo alarme.

A política de permissões de função vinculada ao serviço AWSServiceRoleForCloudWatchAlarms_ActionSSM permite que o Systems Manager conclua as seguintes ações:

- `ssm:CreateOpsItem`

Permissões de função vinculada ao serviço para ações do Systems Manager Incident Manager de alarmes do CloudWatch

O CloudWatch usa a função vinculada ao serviço chamada AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents: o CloudWatch usa essa função vinculada ao serviço para iniciar incidentes do Incident Manager quando o alarme do CloudWatch passa para o estado ALARM.

A função vinculada ao serviço AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents se apoia no serviço do CloudWatch abaixo para assumir a função. Os alarmes do CloudWatch invocam a ação do Systems Manager Incident Manager quando chamado pelo alarme.

A política de permissões de função vinculada ao serviço AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents permite que o Systems Manager conclua as seguintes ações:

- `ssm-incidents:StartIncident`

Permissões de função vinculada ao serviço para o CloudWatch entre contas e entre regiões

O CloudWatch usa a função vinculada ao serviço chamada AWSServiceRoleForCloudWatchCrossAccount: o CloudWatch usa essa função para acessar os dados do CloudWatch em outras contas da AWS especificadas. O SLR fornece apenas a permissão da função assume para permitir que o serviço do CloudWatch assuma a função na conta de compartilhamento. É a função de compartilhamento que fornece acesso aos dados.

A política de permissões de função vinculada ao serviço AWSServiceRoleForCloudWatchCrossAccount permite que o CloudWatch conclua as seguintes ações:

- `sts:AssumeRole`

A função vinculada ao serviço AWSServiceRoleForCloudWatchCrossAccount se apoia no serviço do CloudWatch para assumir a função.

Permissões de perfil vinculado ao serviço do Insights de Performance de banco de dados do CloudWatch

O CloudWatch Application Insights usa o perfil vinculado ao serviço chamado AWSServiceRoleForCloudWatchMetrics_DbPerfInsights. O CloudWatch usa essa perfil para recuperar métricas de Insights de Performance para criar alarmes e capturas instantâneas.

O perfil vinculado ao serviço ao serviço AWSServiceRoleForCloudWatchMetrics_DbPerfInsights tem a política do IAM `AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy` anexada. O conteúdo dessa política está listado como a seguir:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "pi:GetResourceMetrics"
   ],
   "Resource": "*",
   "Condition": {
```

```
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
  }
 ]
}
```

O perfil vinculado ao serviço AWSServiceRoleForCloudWatchMetrics_DbPerfInsights confia no serviço do CloudWatch para assumir o perfil.

Criar uma função vinculada ao serviço para o CloudWatch

Não é necessário criar manualmente nenhuma dessas funções vinculadas ao serviço. A primeira vez que você cria um alarme no AWS Management Console, na CLI do IAM ou na API do IAM, o CloudWatch cria AWSServiceRoleForCloudWatchEvents e AWSServiceRoleForCloudWatchAlarms_ActionSSM para você.

Na primeira vez que você habilita a descoberta de serviços e de topologias, o Application Signals cria um AWSServiceRoleForCloudWatchApplicationSignals para você.

Quando você habilita uma conta para ser uma conta de monitoramento da funcionalidade entre contas e entre regiões, o CloudWatch cria AWSServiceRoleForCloudWatchCrossAccount para você.

Quando você cria pela primeira vez um alarme que usa a função matemática métrica `DB_PERF_INSIGHTS`, o CloudWatch cria AWSServiceRoleForCloudWatchMetrics_DBPerfInsights para você.

Para obter mais informações, consulte Criar uma função vinculada ao serviço no Guia do usuário do IAM.

Editar uma função vinculada ao serviço para o CloudWatch

O CloudWatch não permite editar os perfis AWSServiceRoleForCloudWatchEvents, AWSServiceRoleForCloudWatchAlarms_ActionSSM, AWSServiceRoleForCloudWatchCrossAccount ou AWSServiceRoleForCloudWatchMetrics_DbPerfInsights. Depois de criar essas funções, você não pode alterar seus nomes porque várias entidades podem fazer referência a elas. No entanto, você poderá editar a descrição da função usando o IAM.

Editar a descrição de uma função vinculada ao serviço (console do IAM)

Você pode usar o console do IAM para editar a descrição de uma função vinculada ao serviço.

Para editar a descrição de uma função vinculada ao serviço (console)

1. No painel de navegação do console do IAM, escolha Perfis.
2. Escolha o nome da função a ser modificada.
3. No extremo direito da Descrição da função, escolha Editar.
4. Digite uma nova descrição na caixa e escolha Save (Salvar).

Editar a descrição de uma função vinculada ao serviço (AWS CLI)

Você pode usar comandos do IAM na AWS Command Line Interface para editar a descrição de uma função vinculada ao serviço.

Para alterar a descrição de uma função vinculada ao serviço (AWS CLI)

1. (Opcional) Para visualizar a descrição atual de a uma função, use um dos comandos a seguir:

   ```
   $ aws iam get-role --role-name role-name
   ```

   Use o nome da função, não o ARN, para fazer referência às funções com os comandos da AWS CLI. Por exemplo, se uma função tiver o seguinte nome de recurso da Amazon (ARN): `arn:aws:iam::123456789012:role/myrole`, você fará referência à função como **`myrole`**.
2. Para atualizar a descrição de uma função vinculada ao serviço, use um dos seguintes comandos:

   ```
   $ aws iam update-role-description --role-name role-name --description description
   ```

Editar a descrição de uma função vinculada ao serviço (API do IAM)

Você pode usar a API do IAM para editar a descrição de uma função vinculada ao serviço.

Para alterar a descrição de uma função (API)

1. (Opcional) Para visualizar a descrição atual de uma função, use o comando a seguir:

   GetRole
2. Para atualizar a descrição de uma função, use o comando a seguir:

   UpdateRoleDescription

Excluir uma função vinculada a serviço para o CloudWatch

Se você não tem mais alarmes que automaticamente interrompem, encerram ou reinicializam as instâncias do EC2, recomendamos que você exclua a função AWSServiceRoleForCloudWatchEvents.

Se você não tem mais alarmes que executam ações do Systems Manager OpsCenter, recomendamos excluir a função AWSServiceRoleForCloudWatchAlarms_ActionSSM.

Se você excluir todos os alarmes que usam a função matemática métrica `DB_PERF_INSIGHTS`, é recomendável excluir o perfil vinculado ao serviço AWSServiceRoleForCloudWatchMetrics_DbPerfInsights.

Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar sua função vinculada ao serviço antes de excluí-la.

Limpar uma função vinculada ao serviço

Antes de você poder usar o IAM para excluir uma função vinculada ao serviço, você deve primeiro confirmar que a função não tem sessões ativas e remover quaisquer recursos usados pela função.

Para verificar se a função vinculada ao serviço tem uma sessão ativa no console do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Perfis. Escolha o nome (não a caixa de seleção) da função AWSServiceRoleForCloudWatchEvents.
3. Na página Resumo da função selecionada, escolha Consultor de acesso e analise as atividades recentes para a função vinculada ao serviço.

> **Note**
>
> Se você não tiver certeza se o CloudWatch está usando a função AWSServiceRoleForCloudWatchEvents, tente excluir a função. Se o serviço estiver usando a função, a exclusão falhará e você poderá visualizar as regiões da em que a função está sendo usada. Se a função está sendo usada, você deve aguardar a sessão final antes de excluir a função. Não é possível revogar a sessão de uma função vinculada a um serviço.

Excluir um perfil vinculado ao serviço (console do IAM)

É possível usar o console do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (console)

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Perfis. Marque a caixa de seleção ao lado do nome da função que você deseja excluir, não o nome ou a linha em si.
3. Em Ações de função, escolha Excluir função.
4. Na caixa de diálogo de confirmação, revise os dados do último acesso ao serviço que mostram quando cada uma das funções selecionadas acessou pela última vez um produto da AWS. Isso ajuda você a confirmar se a função está ativo no momento. Para prosseguir, selecione Yes, Delete.
5. Monitore as notificações do console do IAM para progresso da exclusão da função vinculada ao serviço. Como a exclusão da função vinculada ao serviço do IAM é assíncrona, depois que você enviar a função para exclusão, a tarefa de exclusão poderá ser concluída ou falhar. Se a tarefa falhar, escolha Visualizar detalhes ou Visualizar recursos nas notificações para saber por que a exclusão falhou. Se houve falha na exclusão porque há recursos no serviço que estão sendo usados pela função, o motivo da falha incluirá uma lista de recursos.

Excluir uma função vinculada ao serviço (AWS CLI)

Você pode usar comandos do IAM na AWS Command Line Interface para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (AWS CLI)

1. Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tiver recursos associados, você deverá enviar uma solicitação de exclusão. Essa solicitação poderá ser negada se essas condições não forem atendidas. Você deve capturar o `deletion-task-id` da resposta para verificar o status da tarefa de exclusão. Digite o seguinte comando para enviar uma solicitação de exclusão de função vinculada ao serviço:

   ```
   $ aws iam delete-service-linked-role --role-name service-linked-role-name
   ```

2. Digite o seguinte comando para verificar o estado da tarefa de exclusão:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

O status da tarefa de exclusão pode ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, ou `FAILED`. Se a exclusão falhar, a chamada informará o motivo de falha para que você possa solucionar o problema.

Excluir uma função vinculada ao serviço (API do IAM)

É possível usar a API do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (API)

1. Para enviar uma solicitação de exclusão de uma função vinculada ao serviço, chame DeleteServiceLinkedRole. Na solicitação, especifique o nome da função que você deseja excluir.

   Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tiver recursos associados, você deverá enviar uma solicitação de exclusão. Essa solicitação poderá ser negada se essas condições não forem atendidas. Você deve capturar o `DeletionTaskId` da resposta para verificar o status da tarefa de exclusão.
2. Para verificar o status da exclusão, chame GetServiceLinkedRoleDeletionStatus. Na solicitação, especifique o `DeletionTaskId`.

   O status da tarefa de exclusão pode ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, ou `FAILED`. Se a exclusão falhar, a chamada informará o motivo de falha para que você possa solucionar o problema.

Atualizações do CloudWatch para funções vinculadas ao serviço da AWS

Veja detalhes sobre atualizações em políticas gerenciadas pela AWS para o CloudWatch desde que esse serviço começou a monitorar essas alterações. Para obter alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed RSS na página Document history (Histórico de documentos) do CloudWatch.

Alteração	Descrição	Data
AWSServiceRoleForCloudWatchApplicationSignals: atualização nas permissões da política de perfil vinculado ao serviço	O CloudWatch adicionou mais grupos de log ao escopo das permissões `logs:StartQuery` e `logs:GetQueryResults` concedidas por este perfil.	24 de abril de 2024
AWSServiceRoleForCloudWatchApplicationSignals: novo perfil vinculado ao serviço	O CloudWatch adicionou esse novo perfil vinculado ao serviço para permitir que o CloudWatch Application Signals colete dados do CloudWatch Logs, dados de rastreamento do X-Ray, dados de métricas do CloudWatch e dados de marcação de aplicações que você habilitou para o CloudWatch Application Signals.	9 de novembro de 2023
AWSServiceRoleForCloudWatchMetrics_DbPerfInsights: novo perfil vinculado ao serviço	O CloudWatch adicionou esse novo perfil vinculado ao serviço para permitir que o CloudWatch busque métricas de Insights de Performance para alarmes e captura de instantâneos. Uma política do IAM está anexada a esse perfil, e a política concede permissão ao CloudWatch para buscar métricas de	13 de setembro de 2023

Alteração	Descrição	Data
	Insights de Performance em seu nome.	
AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents: nova função vinculada ao serviço	O CloudWatch adicionou uma nova função vinculada ao serviço para permitir que o CloudWatch crie incidentes no AWS Systems Manager Incident Manager.	26 de abril de 2021
O CloudWatch começou a monitorar alterações	O CloudWatch começou a rastrear alterações para seus perfis vinculados ao serviço.	26 de abril de 2021

Usar funções vinculadas ao serviço para o CloudWatch RUM

O CloudWatch RUM usa um perfil vinculado ao serviço do AWS Identity and Access Management (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM ligada diretamente ao RUM. O perfil vinculado ao serviço é predefinido pelo RUM e inclui todas as permissões de que o serviço precisa para chamar outros serviços da AWS em seu nome.

O RUM define as permissões desses perfis vinculados ao serviço e, a menos que definido em contrário, apenas o RUM pode assumir o perfil. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Você só poderá excluir os perfis após excluir os recursos relacionados. Essa restrição protege seus recursos do RUM, pois não é possível remover acidentalmente as permissões para acessá-los.

Para obter informações sobre outros serviços compatíveis com perfis vinculados ao serviço, consulte serviços da AWS que funcionam com o IAM e procure os serviços que apresentam Sim na coluna Perfis vinculados aos serviços. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

Permissões de função vinculada ao serviço no RUM

O RUM usa a função vinculada ao serviço chamada AWSServiceRoleForCloudWatchrum: essa função permite que o RUM envie rastreamento de dados do AWS X-Ray em sua conta, para monitorar aplicações para as quais você habilita o rastreamento de X-Ray.

A função vinculada ao serviço AWSServiceRoleForCloudWatchRUM se apoia no serviço do X-Ray para assumir a função. O X-Ray envia os dados de rastreamento para a sua conta.

A função vinculada ao serviço AWSServiceRoleForCloudWatchRUM tem uma política do IAM anexada chamada AmazonCloudWatchRUMServiceRolePolicy. Essa política concede permissão ao CloudWatch RUM para publicar dados de monitoramento para outros serviços relevantes da AWS. Ela inclui permissões que permitem ao RUM concluir as ações a seguir:

- `xray:PutTraceSegments`
- `cloudwatch:PutMetricData`

Este é o conteúdo completo da política AmazonCloudWatchRUMServiceRolePolicy.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "xray:PutTraceSegments"
   ],
   "Resource": [
    "*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "cloudwatch:namespace": [
      "RUM/CustomMetrics/*",
      "AWS/RUM"
     ]
```

```
      }
    }
   }
  ]
}
```

Criar uma função vinculada ao serviço no RUM

Não é necessário criar manualmente uma função vinculada a serviço para o CloudWatch RUM. Na primeira vez que você cria um monitor de aplicações com o rastreamento de X-Ray habilitado ado ou atualiza um monitor de aplicações para usar o rastreamento de X-Ray, o RUM cria um AWSServiceRoleForCloudWatchRUM para você.

Para obter mais informações, consulte Criar uma função vinculada ao serviço no Guia do usuário do IAM.

Editar uma função vinculada ao serviço no RUM

O CloudWatch RUM não permite que você edite a função AWSServiceRoleForCloudRUM. Depois de criar essas funções, você não pode alterar seus nomes porque várias entidades podem fazer referência a elas. No entanto, você poderá editar a descrição da função usando o IAM.

Editar a descrição de uma função vinculada ao serviço (console do IAM)

Você pode usar o console do IAM para editar a descrição de uma função vinculada ao serviço.

Para editar a descrição de uma função vinculada ao serviço (console)

1. No painel de navegação do console do IAM, escolha Perfis.
2. Escolha o nome da função a ser modificada.
3. No extremo direito da Descrição da função, escolha Editar.
4. Digite uma nova descrição na caixa e escolha Save (Salvar).

Editar a descrição de uma função vinculada ao serviço (AWS CLI)

Você pode usar comandos do IAM na AWS Command Line Interface para editar a descrição de uma função vinculada ao serviço.

Para alterar a descrição de uma função vinculada ao serviço (AWS CLI)

1. (Opcional) Para visualizar a descrição atual de a uma função, use um dos comandos a seguir:

```
$ aws iam get-role --role-name role-name
```

Use o nome da função, não o ARN, para fazer referência às funções com os comandos da AWS CLI. Por exemplo, se uma função tiver o seguinte nome de recurso da Amazon (ARN): `arn:aws:iam::123456789012:role/myrole`, você fará referência à função como **`myrole`**.

2. Para atualizar a descrição de uma função vinculada ao serviço, use um dos seguintes comandos:

```
$ aws iam update-role-description --role-name role-name --description description
```

Editar a descrição de uma função vinculada ao serviço (API do IAM)

Você pode usar a API do IAM para editar a descrição de uma função vinculada ao serviço.

Para alterar a descrição de uma função (API)

1. (Opcional) Para visualizar a descrição atual de uma função, use o comando a seguir:

   GetRole
2. Para atualizar a descrição de uma função, use o comando a seguir:

   UpdateRoleDescription

Excluir uma função vinculada ao serviço no RUM

Se você não tem mais monitores de aplicações com o X-Ray habilitados, recomendamos que você exclua a função AWSServiceRoleForCloudWatchRUM.

Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar sua função vinculada ao serviço antes de excluí-la.

Limpar uma função vinculada ao serviço

Antes de você poder usar o IAM para excluir uma função vinculada ao serviço, você deve primeiro confirmar que a função não tem sessões ativas e remover quaisquer recursos usados pela função.

Para verificar se a função vinculada ao serviço tem uma sessão ativa no console do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Perfis. Escolha o nome (não a caixa de seleção) da função AWSServiceRoleForCloudWatchRUM.
3. Na página Resumo da função selecionada, escolha Consultor de acesso e analise as atividades recentes para a função vinculada ao serviço.

> **Note**
>
> Se você não tiver certeza se o RUM está usando a função AWSServiceRoleForCloudWatchRUM, tente excluí-la. Se o serviço estiver usando a função, a exclusão falhará e você poderá visualizar as regiões da em que a função está sendo usada. Se a função está sendo usada, você deve aguardar a sessão final antes de excluir a função. Não é possível revogar a sessão de uma função vinculada a um serviço.

Excluir um perfil vinculado ao serviço (console do IAM)

É possível usar o console do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (console)

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Perfis. Marque a caixa de seleção ao lado do nome da função que você deseja excluir, não o nome ou a linha em si.
3. Em Ações de função, escolha Excluir função.
4. Na caixa de diálogo de confirmação, revise os dados do último acesso ao serviço que mostram quando cada uma das funções selecionadas acessou pela última vez um produto da AWS. Isso ajuda você a confirmar se a função está ativo no momento. Para prosseguir, selecione Yes, Delete.
5. Monitore as notificações do console do IAM para progresso da exclusão da função vinculada ao serviço. Como a exclusão da função vinculada ao serviço do IAM é assíncrona, depois que você enviar a função para exclusão, a tarefa de exclusão poderá ser concluída ou falhar. Se a tarefa falhar, escolha Visualizar detalhes ou Visualizar recursos nas notificações para saber por que

a exclusão falhou. Se houve falha na exclusão porque há recursos no serviço que estão sendo usados pela função, o motivo da falha incluirá uma lista de recursos.

Excluir uma função vinculada ao serviço (AWS CLI)

Você pode usar comandos do IAM na AWS Command Line Interface para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (AWS CLI)

1. Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tiver recursos associados, você deverá enviar uma solicitação de exclusão. Essa solicitação poderá ser negada se essas condições não forem atendidas. Você deve capturar o `deletion-task-id` da resposta para verificar o status da tarefa de exclusão. Digite o seguinte comando para enviar uma solicitação de exclusão de função vinculada ao serviço:

   ```
   $ aws iam delete-service-linked-role --role-name service-linked-role-name
   ```

2. Digite o seguinte comando para verificar o estado da tarefa de exclusão:

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   O status da tarefa de exclusão pode ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, ou `FAILED`. Se a exclusão falhar, a chamada informará o motivo de falha para que você possa solucionar o problema.

Excluir uma função vinculada ao serviço (API do IAM)

É possível usar a API do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (API)

1. Para enviar uma solicitação de exclusão de uma função vinculada ao serviço, chame DeleteServiceLinkedRole. Na solicitação, especifique o nome da função que você deseja excluir.

   Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tiver recursos associados, você deverá enviar uma solicitação de exclusão. Essa solicitação poderá

ser negada se essas condições não forem atendidas. Você deve capturar o `DeletionTaskId` da resposta para verificar o status da tarefa de exclusão.

2. Para verificar o status da exclusão, chame GetServiceLinkedRoleDeletionStatus. Na solicitação, especifique o `DeletionTaskId`.

   O status da tarefa de exclusão pode ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, ou `FAILED`. Se a exclusão falhar, a chamada informará o motivo de falha para que você possa solucionar o problema.

Usar funções vinculadas ao serviço do CloudWatch Application Insights

O CloudWatch Application Insights usa funções vinculadas ao serviço do AWS Identity and Access Management (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao CloudWatch Application Insights. As funções vinculadas ao serviço são predefinidas pelo CloudWatch Application Insights e incluem todas as permissões que o serviço requer para chamar outros produtos da AWS em seu nome.

Uma função vinculada ao serviço facilita a configuração do CloudWatch Application Insights porque você não precisa adicionar as permissões necessárias manualmente. O CloudWatch Application Insights define as permissões das funções vinculadas ao serviço e, exceto se definido de outra forma, somente o CloudWatch Application Insights pode assumir essas funções. As permissões definidas incluem a política de confiança e a política de permissões, e essa política não pode ser anexada a nenhuma outra entidade do IAM.

Para obter informações sobre outros serviços compatíveis com funções vinculadas aos serviços, consulte Serviços da AWS compatíveis com o IAM e procure os serviços que contenham Sim na coluna Funções vinculadas ao serviço. Selecione o link Yes (Sim) para exibir a documentação da função vinculada ao serviço para esse serviço.

Permissões de função vinculada ao serviço do CloudWatch Application Insights

O CloudWatch Application Insights usa a função vinculada ao serviço chamada AWSServiceRoleForApplicationInsights. O Application Insights usa essa função para executar operações como analisar os grupos de recursos do cliente, criar pilhas do CloudFormation para criar alarmes de métricas e configurar o CloudWatch Agent em instâncias do EC2. Esse perfil vinculado a serviço tem uma política do IAM anexada que se chama `CloudwatchApplicationInsightsServiceLinkedRolePolicy`. Para obter atualizações dessa política, consulte O Application Insights atualiza para políticas gerenciadas pela AWS.

A política de permissões da função permite que o CloudWatch Application Insights realize as ações a seguir nos recursos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:PutAnomalyDetector",
        "cloudwatch:DeleteAnomalyDetector",
        "cloudwatch:DescribeAnomalyDetectors"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:FilterLogEvents",
        "logs:GetLogEvents",
        "logs:DescribeLogStreams",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:DescribeRule"
      ],
      "Resource": [
        "*"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudFormation:CreateStack",
        "cloudFormation:UpdateStack",
        "cloudFormation:DeleteStack",
        "cloudFormation:DescribeStackResources"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudFormation:DescribeStacks",
        "cloudFormation:ListStackResources",
        "cloudFormation:ListStacks"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:ListGroupResources",
        "resource-groups:GetGroupQuery",
        "resource-groups:GetGroup"
      ],
      "Resource": [
        "*"
      ]
    },
```

```
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:CreateGroup",
        "resource-groups:DeleteGroup"
      ],
      "Resource": [
        "arn:aws:resource-groups:*:*:group/ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter",
        "ssm:DeleteParameter",
        "ssm:AddTagsToResource",
        "ssm:RemoveTagsFromResource",
        "ssm:GetParameters"
      ],
      "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-ApplicationInsights-*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "ssm:CreateAssociation",
        "ssm:UpdateAssociation",
        "ssm:DeleteAssociation",
        "ssm:DescribeAssociation"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:association/*",
        "arn:aws:ssm:*:*:managed-instance/*",
        "arn:aws:ssm:*:*:document/AWSEC2-
ApplicationInsightsCloudwatchAgentInstallAndConfigure",
        "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
        "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem",
        "ssm:CreateOpsItem",
        "ssm:DescribeOpsItems",
        "ssm:UpdateOpsItem",
        "ssm:DescribeInstanceInformation"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:AddTagsToResource"
      ],
      "Resource": "arn:aws:ssm:*:*:opsitem/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommandInvocations",
        "ssm:GetCommandInvocation"
      ],
      "Resource": [
        "*"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:document/AWSEC2-CheckPerformanceCounterSets",
        "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
        "arn:aws:ssm:*:*:document/AWSEC2-DetectWorkload",
        "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeNatGateways"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:ListFunctions",
        "lambda:GetFunctionConfiguration",
        "lambda:ListEventSourceMappings"
      ],
      "Resource": [
```

```
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "events:DeleteRule"
      ],
      "Resource": [
        "arn:aws:events:*:*:rule/AmazonCloudWatch-ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "xray:GetServiceGraph",
        "xray:GetTraceSummaries",
        "xray:GetTimeSeriesServiceStatistics",
        "xray:GetTraceGraph"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:ListTables",
        "dynamodb:DescribeTable",
        "dynamodb:DescribeContributorInsights",
        "dynamodb:DescribeTimeToLive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:DescribeScalableTargets"
      ],
```

```
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetMetricsConfiguration",
        "s3:GetReplicationConfiguration"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "states:ListStateMachines",
        "states:DescribeExecution",
        "states:DescribeStateMachine",
        "states:GetExecutionHistory"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:GET"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:DescribeServices",
        "ecs:DescribeTaskDefinition",
        "ecs:DescribeTasks",
```

```
        "ecs:DescribeTaskSets",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListServices",
        "ecs:ListTasks"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:UpdateClusterSettings"
      ],
      "Resource": [
        "arn:aws:ecs:*:*:cluster/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:DescribeFargateProfile",
        "eks:DescribeNodegroup",
        "eks:ListClusters",
        "eks:ListFargateProfiles",
        "eks:ListNodegroups",
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:GetSubscriptionAttributes",
        "sns:GetTopicAttributes",
        "sns:GetSMSAttributes",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics"
      ],
```

```
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ListQueues"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:DeleteSubscriptionFilter"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:PutSubscriptionFilter"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:*",
        "arn:aws:logs:*:*:destination:AmazonCloudWatch-ApplicationInsights-
LogIngestionDestination*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53:GetHostedZone",
```

```
        "route53:GetHealthCheck",
        "route53:ListHostedZones",
        "route53:ListHealthChecks",
        "route53:ListQueryLoggingConfigs"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53resolver:ListFirewallRuleGroupAssociations",
        "route53resolver:GetFirewallRuleGroup",
        "route53resolver:ListFirewallRuleGroups",
        "route53resolver:ListResolverEndpoints",
        "route53resolver:GetResolverQueryLogConfig",
        "route53resolver:ListResolverQueryLogConfigs",
        "route53resolver:ListResolverQueryLogConfigAssociations",
        "route53resolver:GetResolverEndpoint",
        "route53resolver:GetFirewallRuleGroupAssociation"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada ao serviço. Para obter mais informações, consulte Permissões de função vinculada ao serviço no Guia do usuário do IAM.

Criar uma função vinculada ao serviço do CloudWatch Application Insights

Não é necessário criar manualmente uma função vinculada ao serviço. Ao criar uma aplicação com o Application Insights no AWS Management Console, o CloudWatch Application Insights criará a função vinculada ao serviço para você.

Se você excluir essa função vinculada ao serviço e quiser criá-la novamente, poderá usar esse mesmo processo para recriar a função em sua conta. Ao criar uma aplicação com o Application

Insights, o CloudWatch Application Insights criará a função vinculada ao serviço para você novamente.

Editar uma função vinculada ao serviço do CloudWatch Application Insights

O CloudWatch Application Insights não permite editar a função vinculada ao serviço AWSServiceRoleForApplicationInsights. Depois que criar uma função vinculada ao serviço, você não poderá alterar o nome da função, pois várias entidades podem fazer referência a ela. No entanto, será possível editar a descrição da função usando o IAM. Para obter mais informações, consulte Editar uma função vinculada ao serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço do CloudWatch Application Insights

Se você não precisar mais usar um recurso ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. Dessa forma, você evita ter uma entidade não utilizada que não seja monitorada ou mantida ativamente. No entanto, você deve excluir todas as aplicações no Application Insights antes de excluir manualmente a função.

> **Note**
>
> Se o serviço CloudWatch Application Insights estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir os recursos do CloudWatch Application Insights usados por AWSServiceRoleForApplicationInsights

- Exclua todos as suas aplicações do CloudWatch Application Insights. Para obter mais informações, consulte “Excluir suas aplicações” no Manual do usuário do CloudWatch Application Insights.

Como excluir manualmente a função vinculada ao serviço usando o IAM

Use o console do IAM, a AWS CLI ou a API da AWS para excluir a função vinculada a serviço AWSServiceRoleForApplicationInsights. Para obter mais informações, consulte Excluir uma função vinculada ao serviço no Guia do usuário do IAM.

Regiões compatíveis com funções vinculadas ao serviço do CloudWatch Application Insights

O CloudWatch Application Insights oferece suporte a funções vinculadas a serviços em todas as regiões da AWS em que o serviço está disponível. Para obter mais informações, consulte Regiões e endpoints do CloudWatch Application Insights.

Políticas gerenciadas pela AWS para o Amazon CloudWatch Application Insights

Uma política gerenciada pela AWS é uma política independente criada e administrada pela AWS. As políticas gerenciadas pela AWS são criadas para fornecer permissões a vários casos de uso comuns a fim de que você possa começar a atribuir permissões a usuários, grupos e perfis.

Lembre-se de que as políticas gerenciadas pela AWS podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para todos os clientes da AWS usarem. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo cliente específicas para seus casos de uso.

Você não pode alterar as permissões definidas em políticas gerenciadas pela AWS. Se a AWS atualiza as permissões definidas em um política gerenciada pela AWS, a atualização afeta todas as identidades de entidades principais (usuários, grupos e perfis) às quais a política está vinculada. É mais provável que a AWS atualize uma política gerenciada pela AWS quando um novo AWS service (Serviço da AWS) é lançado ou novas operações de API são disponibilizadas para os serviços existentes.

Para obter mais informações, consulte Políticas gerenciadas pela AWS no Manual do usuário do IAM.

Política gerenciada pela AWS: CloudWatchApplicationInsightsFullAccess

É possível anexar a política `CloudWatchApplicationInsightsFullAccess` a suas identidades do IAM.

Essa política concede permissões administrativas que oferecem acesso total à funcionalidade do Application Insights.

Detalhes da permissão

Esta política inclui as seguintes permissões.

- `applicationinsights`: oferece acesso total à funcionalidade do Application Insights.
- `iam`: permite que o Application Insights crie a função vinculada ao serviço AWSServiceRoleForApplicationInsights. Isso é necessário para que o Application Insights execute operações como analisar os grupos de recursos do cliente, criar pilhas do CloudFormation para criar alarmes de métricas e configurar o CloudWatch Agent em instâncias do EC2. Para obter mais informações, consulte Usar funções vinculadas ao serviço do CloudWatch Application Insights.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "applicationinsights:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "sqs:ListQueues",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "autoscaling:DescribeAutoScalingGroups",
        "lambda:ListFunctions",
        "dynamodb:ListTables",
        "s3:ListAllMyBuckets",
```

```
        "sns:ListTopics",
        "states:ListStateMachines",
        "apigateway:GET",
        "ecs:ListClusters",
        "ecs:DescribeTaskDefinition",
        "ecs:ListServices",
        "ecs:ListTasks",
        "eks:ListClusters",
        "eks:ListNodegroups",
        "fsx:DescribeFileSystems",
        "logs:DescribeLogGroups",
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/application-insights.amazonaws.com/
AWSServiceRoleForApplicationInsights"
      ],
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "application-insights.amazonaws.com"
        }
      }
    }
  ]
}
```

Política gerenciada pela AWS: CloudWatchApplicationInsightsReadOnlyAccess

É possível anexar a política `CloudWatchApplicationInsightsReadOnlyAccess` a suas identidades do IAM.

Essa política concede permissões administrativas que oferecem acesso somente para leitura à funcionalidade do Application Insights.

Detalhes da permissão

Esta política inclui as seguintes permissões.

- `applicationinsights`: oferece acesso somente para leitura à funcionalidade do Application Insights.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "applicationinsights:Describe*",
                "applicationinsights:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Política gerenciada pela AWS: CloudwatchApplicationInsightsServiceLinkedRolePolicy

Não é possível anexar CloudwatchApplicationInsightsServiceLinkedRolePolicy a suas entidades do IAM. Essa política é anexada a uma função vinculada ao serviço que permite que o Application Insights monitore os recursos do cliente. Para obter mais informações, consulte Usar funções vinculadas ao serviço do CloudWatch Application Insights.

O Application Insights atualiza para políticas gerenciadas pela AWS

Visualize detalhes sobre atualizações em políticas gerenciadas pela AWS para o Application Insights desde que esse serviço começou a monitorar essas alterações. Para receber alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed de RSS na página Document History (Histórico de documentos) do Application Insights.

Alteração	Descrição	Data
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para listar as pilhas do CloudFormation. Essas permissões são necessárias para que o Amazon CloudWatch Application Insights analise e monitore recursos da AWS aninhados na pilha do CloudFormation.	24 de abril de 2023
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para obter uma lista de recursos da Amazon VPC e do Route 53. Essas permissões são necessárias para o Amazon CloudWatch Application Insights configurar automaticamente o monitoramento de rede de práticas recomendadas com o Amazon CloudWatch.	23 de janeiro de 2023
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para obter resultados de invocação de comandos do SSM. Essas permissões são necessárias para o Amazon CloudWatch Applicati	19 de dezembro de 2022

Alteração	Descrição	Data
	on Insights detectar e monitorar automaticamente as workloads em execução nas instâncias do Amazon EC2.	
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para descrever recursos da Amazon VPC e do Route 53. Essas permissões são necessárias para que o Amazon CloudWatch Application Insights leia as configurações de recursos da Amazon VPC e do Route 53 do cliente e ajude os clientes a configurar automaticamente as práticas recomendadas de monitoramento de rede com o Amazon CloudWatch.	19 de dezembro de 2022

Alteração	Descrição	Data
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para descrever recursos do EFS. Essas permissões são necessárias para que o Amazon CloudWatch Application Insights leia as configurações de recursos do cliente do Amazon EFS e ajude os clientes a configurar automaticamente as práticas recomendadas para monitoramento do EFS com o CloudWatch.	3 de outubro de 2022
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para descrever os recursos do EFS. Essas permissões são necessárias para que o Amazon CloudWatch Application Insights crie aplicações baseadas em conta consultando todos os recursos compatíveis em uma conta.	3 de outubro de 2022

Alteração	Descrição	Data
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para recuperar informações sobre recursos do FSx. Essas permissões são necessárias para o Amazon CloudWatch Application Insights monitorar as workloads recuperando informações suficientes sobre os volumes do FSx subjacentes.	12 de setembro de 2022
Política gerenciada pela AWS: CloudWatchApplicationInsightsFullAccess: atualizar para uma política existente	O Application Insights adicionou novas permissões para descrever grupos de log. Essas permissões são necessárias para que o Amazon CloudWatch Application Insights garanta que as permissões corretas para monitorar grupos de log estejam em uma conta quando uma nova aplicação for criada.	24 de janeiro de 2022

Alteração	Descrição	Data
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para criar e excluir filtros de assinatura do log do CloudWatch. Essas permissões são necessárias para que o Amazon CloudWatch Application Insights crie filtros de assinatura para facilitar o monitoramento de logs de recursos nas aplicações configuradas.	24 de janeiro de 2022
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para descrever grupos de destino e integridade de destino para o Elastic Load Balancers. Essas permissões são necessárias para que o Amazon CloudWatch Application Insights crie aplicações baseadas em conta consultando todos os recursos compatíveis em uma conta.	4 de novembro de 2021

Alteração	Descrição	Data
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	Application Insights adicionou novas permissões para executar o documento SSM `AmazonCloudWatch-ManageAgent` em instâncias do Amazon EC2. Essas permissões são necessárias para que o Amazon CloudWatch Application Insights limpe os arquivos de configuração do agente do CloudWatch criados pelo Application Insights.	30 de setembro de 2021

Alteração	Descrição	Data
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para dar suporte ao monitoramento de aplicações baseado em conta para integrar e monitorar todos os recursos compatíveis em sua conta. Essas permissões são necessárias para o Amazon CloudWatch Application Insights consultar, marcar recursos e criar grupos para esses recursos. O Application Insights adicionou novas permissões para suportar o monitoramento de tópicos do SNS. Essas permissões são necessárias para o Amazon CloudWatch Application Insights coletar metadados de recursos do SNS para configurar o monitoramento de tópicos do SNS.	15 de setembro de 2021

Alteração	Descrição	Data
Política gerenciada pela AWS: CloudWatchApplicationInsightsFullAccess: atualizar para uma política existente	O Application Insights adicionou novas permissões para descrever e listar recursos compatíveis. Essas permissões são necessárias para que o Amazon CloudWatch Application Insights crie aplicações baseadas em conta consultando todos os recursos compatíveis em uma conta.	15 de setembro de 2021
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para descrever recursos do FSX. Essas permissões são necessárias para que o Amazon CloudWatch Application Insights leia as configurações de recursos do FSX do cliente e ajude os clientes a configurar automaticamente o monitoramento do FSX de práticas recomendadas com o CloudWatch.	31 de agosto de 2021

Alteração	Descrição	Data
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para descrever e listar recursos de serviços do ECS e do EKS. Essa permissão é necessária para que o Amazon CloudWatch Application Insights leia as configurações de recursos do contêiner do cliente e ajude os clientes a configurar automaticamente o monitoramento do contêiner de práticas recomendadas com o CloudWatch.	18 de maio de 2021
CloudWatchApplicationInsightsServiceLinkedRolePolicy: atualização de uma política existente	O Application Insights adicionou novas permissões para permitir que o OpsCenter marque OpsItems usando ação `ssm:AddTagsToResource` em recursos com o tipo de recurso `opsitem`. Essa permissão é requerida pelo OpsCenter. O Amazon CloudWatch Application Insights cria OpsItems para que o cliente possa resolver problemas usando o AWS SSM OpsCenter.	13 de abril de 2021

Alteração	Descrição	Data
O Application Insights começou a controlar alteraçõe s	O Application Insights começou a monitorar alterações para políticas gerenciadas pela AWS.	13 de abril de 2021

Referência de permissões do Amazon CloudWatch

A tabela a seguir lista cada operação da API do CloudWatch e as ações correspondentes às quais é possível conceder permissões para executar a ação. Você especifica as ações no campo `Action` da política e especifica um caractere curinga (*) como o valor do recurso no campo `Resource` da política.

É possível usar as chaves de condição de toda a AWS em suas políticas do CloudWatch para expressar condições. Para obter uma lista completa de chaves em toda a AWS, consulte Chaves de contexto de condição globais e do IAM da AWS no Manual do usuário do IAM.

> **Note**
>
> Para especificar uma ação, use o prefixo `cloudwatch:` seguido do nome da operação da API. Por exemplo: `cloudwatch:GetMetricData`, `cloudwatch:ListMetrics` ou `cloudwatch:*` (para todas as ações do CloudWatch).

Tópicos

Operações da API do CloudWatch e permissões necessárias para ações

Operações da API do CloudWatch	Permissões obrigatórias (ações de API)
DeleteAlarms	`cloudwatch:DeleteAlarms` Necessária para excluir um alarme.
DeleteDashboards	`cloudwatch:DeleteDashboards` Necessária para apagar um painel.
DeleteMetricStream	`cloudwatch:DeleteMetricStream` Necessária para excluir um fluxo de métricas.
DescribeAlarmHistory	`cloudwatch:DescribeAlarmHistory` Necessária para visualizar o histórico de alarmes. Para recuperar informações sobre alarmes compostos, sua permissão `cloudwatch:DescribeAlarmHistory` deve ter um escopo `*`. Não é possível retornar informações sobre alarmes compostos se a permissão `cloudwatch:DescribeAlarmHistory` tiver um escopo mais limitado.
DescribeAlarms	`cloudwatch:DescribeAlarms` Necessária para recuperar informações sobre alarmes. Para recuperar informações sobre alarmes compostos, sua permissão `cloudwatch:DescribeAlarms` deve ter um escopo

Operações da API do CloudWatch	Permissões obrigatórias (ações de API)
	*. Não é possível retornar informações sobre alarmes compostos se a permissão `cloudwatch:DescribeAlarms` tiver um escopo mais limitado.
DescribeAlarmsForMetric	`cloudwatch:DescribeAlarmsForMetric` Necessária para visualizar os alarmes para uma métrica.
DisableAlarmActions	`cloudwatch:DisableAlarmActions` Necessária para desativar uma ação de alarme.
EnableAlarmActions	`cloudwatch:EnableAlarmActions` Necessária para ativar uma ação de alarme.
GetDashboard	`cloudwatch:GetDashboard` Necessária para exibir dados sobre os painéis existentes.
GetMetricData	`cloudwatch:GetMetricData` Necessária para representar em gráficos dados de métricas no console do CloudWatch para recuperar grandes lotes de dados de métricas e executar matemática métrica nesses dados.

Operações da API do CloudWatch	Permissões obrigatórias (ações de API)
GetMetricStatistics	`cloudwatch:GetMetricStatistics` Necessária para visualizar gráficos em outras partes do console do CloudWatch e nos widgets do painel.
GetMetricStream	`cloudwatch:GetMetricStream` Necessária para visualizar informações sobre um fluxo de métricas.
GetMetricWidgetImage	`cloudwatch:GetMetricWidgetImage` Necessário para recuperar um gráfico de snapshot de uma ou mais métricas do CloudWatch como uma imagem de bitmap.
ListDashboards	`cloudwatch:ListDashboards` Necessária para visualizar a lista de painéis do CloudWatch em sua conta.
ListMetrics	`cloudwatch:ListMetrics` Necessária para visualizar ou pesquisar nomes de métrica no console do CloudWatch e na CLI. Necessária para selecionar métricas nos widgets do painel.
ListMetricStreams	`cloudwatch:ListMetricStreams` Necessária para visualizar ou pesquisar a lista de fluxos de métricas na conta.

Operações da API do CloudWatch	Permissões obrigatórias (ações de API)
PutCompositeAlarm	`cloudwatch:PutCompositeAlarm` Necessária para criar um alarme composto. Para criar um alarme composto, sua permissão `cloudwatch:PutCompositeAlarm` deve ter um escopo `*`. Não é possível retornar informações sobre alarmes compostos se a permissão `cloudwatch:PutCompositeAlarm` tiver um escopo mais limitado.
PutDashboard	`cloudwatch:PutDashboard` Necessária para criar um painel ou atualizar um painel existente.
PutMetricAlarm	`cloudwatch:PutMetricAlarm` Necessária para criar ou atualizar um alarme.
PutMetricData	`cloudwatch:PutMetricData` Necessária para criar métricas.
PutMetricStream	`cloudwatch:PutMetricStream` Necessária para criar um fluxo de métricas.
SetAlarmState	`cloudwatch:SetAlarmState` Necessária para definir manualmente o estado de um alarme.

Operações da API do CloudWatch	Permissões obrigatórias (ações de API)
StartMetricStreams	`cloudwatch:StartMetricStreams` Necessária para iniciar o fluxo de métricas.
StopMetricStreams	`cloudwatch:StopMetricStreams` Necessária para interromper temporariamente o fluxo de métricas.
TagResource	`cloudwatch:TagResource` Necessária para adicionar ou atualizar etiquetas nos recursos do CloudWatch, como alarmes e regras do Contributor Insights.
UntagResource	`cloudwatch:UntagResource` Necessária para remover etiquetas de recurso no CloudWatch.

Operações da API do CloudWatch Contributor Insights e permissões necessárias para ações

Important

Quando você concede a permissão `cloudwatch:PutInsightRule` ao usuário, por padrão, ele pode criar uma regra que avalia qualquer grupo de logs no CloudWatch Logs. É possível adicionar condições de política do IAM que limitem essas permissões para que um usuário inclua e exclua grupos de logs específicos. Para ter mais informações, consulte Usar chaves de condição para limitar o acesso dos usuários do Contributor Insights aos grupos de log.

Operações da API do CloudWatch Contributor Insights	Permissões obrigatórias (ações de API)
DeleteInsightRules	`cloudwatch:DeleteInsightRules` Necessária para excluir regras do Contributor Insights.
DescribeInsightRules	`cloudwatch:DescribeInsightRules` Necessária para visualizar as regras do Contributor Insights em sua conta.
EnableInsightRules	`cloudwatch:EnableInsightRules` Necessária para habilitar regras do Contributor Insights.
GetInsightRuleReport	`cloudwatch:GetInsightRuleReport` Necessário para recuperar dados de séries temporais e outras estatísticas coletadas pelas regras do Contributor Insights.
PutInsightRule	`cloudwatch:PutInsightRule` Necessária para criar regras do Contributor Insights. Consulte a observação Importante no início desta tabela.

Operações da API do CloudWatch Events e permissões necessárias para ações

Operações de API do CloudWatch Events	Permissões obrigatórias (ações de API)

Operações de API do CloudWatch Events	Permissões obrigatórias (ações de API)
DeleteRule	`events:DeleteRule` Necessária para excluir uma regra.
DescribeRule	`events:DescribeRule` Necessária para listar os detalhes sobre uma regra.
DisableRule	`events:DisableRule` Necessária para desativar uma regra.
EnableRule	`events:EnableRule` Necessária para ativar uma regra.
ListRuleNamesByTarget	`events:ListRuleNamesByTarget` Necessária para listar as regras associadas a um destino.
ListRules	`events:ListRules` Necessária para listar todas as regras em sua conta.
ListTargetsByRule	`events:ListTargetsByRule` Necessária para listar todos os destinos associados a uma regra.

Operações de API do CloudWatch Events	Permissões obrigatórias (ações de API)
PutEvents	`events:PutEvents` Necessária para adicionar eventos personalizados que podem ser vinculados a regras.
PutRule	`events:PutRule` Necessária para criar ou atualizar uma regra.
PutTargets	`events:PutTargets` Necessária para adicionar destinos a uma regra.
RemoveTargets	`events:RemoveTargets` Necessária para remover um destino de uma regra.
TestEventPattern	`events:TestEventPattern` Necessária para testar um evento padrão em um determinado evento.

Operações de API do CloudWatch Logs e permissões necessárias para ações

Operações da API do CloudWatch Logs	Permissões obrigatórias (ações de API)
CancelExportTask	`logs:CancelExportTask` Necessária para cancelar uma tarefa de exportação pendente ou em execução.

Operações da API do CloudWatch Logs	Permissões obrigatórias (ações de API)
CreateExportTask	`logs:CreateExportTask` Necessária para exportar dados de um grupo de logs para um bucket do Amazon S3.
CreateLogGroup	`logs:CreateLogGroup` Necessária para criar um novo grupo de logs.
CreateLogStream	`logs:CreateLogStream` Necessária para criar um novo stream de logs em um grupo de logs.
DeleteDestination	`logs:DeleteDestination` Necessária para excluir um destino de log e desativar seus filtros de assinatura.
DeleteLogGroup	`logs:DeleteLogGroup` Necessária para excluir um grupo de logs e eventos de log arquivados associados.
DeleteLogStream	`logs:DeleteLogStream` Necessária para excluir um stream de logs e eventos de log arquivados associados.
DeleteMetricFilter	`logs:DeleteMetricFilter` Necessária para excluir um filtro de métrica associado a um grupo de logs.

Operações da API do CloudWatch Logs	Permissões obrigatórias (ações de API)
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Necessária para excluir uma definição de consulta salva no CloudWatch Logs Insights.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Necessária para excluir uma política de recursos do CloudWatch Logs.
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Necessária para excluir uma política de retenção do grupo de logs.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Necessária para excluir o filtro de assinatura associado a um grupo de logs.
DescribeDestinations	`logs:DescribeDestinations` Necessária para visualizar todos os destinos associado à conta.
DescribeExportTasks	`logs:DescribeExportTasks` Necessária para visualizar todas as tarefas de exportação associadas à conta.
DescribeLogGroups	`logs:DescribeLogGroups` Necessária para visualizar todos os grupos de logs associados à conta.

Operações da API do CloudWatch Logs	Permissões obrigatórias (ações de API)
DescribeLogStreams	`logs:DescribeLogStreams` Necessária para visualizar todos os streams de logs associados a um grupo de logs.
DescribeMetricFilters	`logs:DescribeMetricFilters` Necessária para visualizar todas as métricas associadas a um grupo de logs.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Necessária para visualizar a lista de definiçõe s de consulta salvas no CloudWatch Logs Insights.
DescribeQueries	`logs:DescribeQueries` Necessária para ver a lista de consultas do CloudWatch Logs Insights que estão agendadas, estão em execução ou foram executadas recentemente.
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Necessária para exibir uma lista de políticas de recursos do CloudWatch Logs.
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Necessária para visualizar todos os filtros de assinatura associados a um grupo de logs.

Operações da API do CloudWatch Logs	Permissões obrigatórias (ações de API)
FilterLogEvents	`logs:FilterLogEvents` Necessária para classificar eventos de log por padrão de filtros de grupos.
GetLogEvents	`logs:GetLogEvents` Necessária para recuperar eventos de log a partir de um stream de logs.
GetLogGroupFields	`logs:GetLogGroupFields` Necessária para recuperar a lista de campos incluídos nos eventos de log em um grupo de log.
GetLogRecord	`logs:GetLogRecord` Necessário para recuperar os detalhes de um único evento de log.
GetQueryResults	`logs:GetQueryResults` Necessária para recuperar os resultados das consultas do CloudWatch Logs Insights.
ListTagsLogGroup	`logs:ListTagsLogGroup` Necessária para listar as tags associadas a um grupo de log.

Operações da API do CloudWatch Logs	Permissões obrigatórias (ações de API)
PutDestination	`logs:PutDestination` Necessária para criar ou atualizar um fluxo de logs de destino (como um fluxo do Kinesis).
PutDestinationPolicy	`logs:PutDestinationPolicy` Necessária para criar ou atualizar uma política de acesso associada a um destino de log existente.
PutLogEvents	`logs:PutLogEvents` Necessária para carregar um lote de eventos de log para um stream de log.
PutMetricFilter	`logs:PutMetricFilter` Necessária para criar ou atualizar um filtro de métrica e associá-lo a um grupo de logs.
PutQueryDefinition	`logs:PutQueryDefinition` Necessária para salvar uma consulta no CloudWatch Logs Insights.
PutResourcePolicy	`logs:PutResourcePolicy` Necessário para criar uma política de recursos do CloudWatch Logs.

Operações da API do CloudWatch Logs	Permissões obrigatórias (ações de API)
PutRetentionPolicy	`logs:PutRetentionPolicy` Necessária para definir o número de dias nos quais manter os eventos de log (retenção) em um grupo de logs.
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Necessária para criar ou atualizar um filtro de assinatura e associá-lo a um grupo de logs.
StartQuery	`logs:StartQuery` Necessária para iniciar consultas do CloudWatch Logs Insights.
StopQuery	`logs:StopQuery` Necessária para interromper uma consulta do CloudWatch Logs Insights que está em andamento.
TagLogGroup	`logs:TagLogGroup` Obrigatório para adicionar ou atualizar as tags do grupo de logs.
TestMetricFilter	`logs:TestMetricFilter` Necessária para testar um padrão de filtro em relação a uma amostra de mensagens de eventos de log.

Operações de API do Amazon EC2 e permissões necessárias para ações

Operação da API do Amazon EC2	Permissões obrigatórias (ações de API)
DescribeInstanceStatus	`ec2:DescribeInstanceStatus` Necessária para visualizar os detalhes de status da instância do EC2.
DescribeInstances	`ec2:DescribeInstances` Necessária para visualizar detalhes da instância do EC2.
RebootInstances	`ec2:RebootInstances` Necessária para reinicializar uma instância do EC2.
StopInstances	`ec2:StopInstances` Necessária para interromper uma instância do EC2.
TerminateInstances	`ec2:TerminateInstances` Necessária para encerrar uma instância do EC2.

Operações de API do Amazon EC2 Auto Scaling e permissões necessárias para ações

Operações da API do Amazon EC2 Auto Scaling	Permissões obrigatórias (ações de API)

Operações da API do Amazon EC2 Auto Scaling	Permissões obrigatórias (ações de API)
Escalabilidade	`autoscaling:Scaling` Necessária para escalar um grupo do Auto Scaling.
Trigger	`autoscaling:Trigger` Necessária para acionar uma ação do Auto Scaling.

Validação de conformidade do Amazon CloudWatch

Auditores externos avaliam a segurança e a compatibilidade do Amazon CloudWatch como parte de vários programas de compatibilidade da AWS. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

Para obter uma lista de serviços da AWS no escopo de programas de conformidade específicos, consulte Serviços da AWS no escopo por programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

É possível fazer download de relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte Fazer download de relatórios no AWS Artifact.

Sua responsabilidade de compatibilidade ao usar o Amazon CloudWatch é determinada pela confidencialidade de seus dados, pelos objetivos de compatibilidade de sua empresa e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a compatibilidade:

- Guias de início rápido de segurança e compatibilidade: estes guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em compatibilidade e segurança na AWS.
- Whitepaper Architecting for HIPAA Security and Compliance: este whitepaper descreve como as empresas podem usar a AWS para criar aplicações em conformidade com a HIPAA.
- Recursos de conformidade da AWS: essa coleção de manuais e guias pode ser aplicada a seu setor e local.

- Avaliar recursos com regras no AWS Config Guia do desenvolvedor: AWS Config; avalia como suas configurações de recursos estão em conformidade com práticas internas, diretrizes do setor e regulamentos.
- AWS Security Hub: esse serviço da AWS fornece uma visão abrangente do estado de sua segurança na AWS que ajuda você a conferir sua conformidade com padrões e práticas recomendadas de segurança do setor.

Resiliência no Amazon CloudWatch

A infraestrutura global da AWS é criada com base em regiões e zonas de disponibilidade da AWS. As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, alta throughput e redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte Infraestrutura global da AWS.

Segurança de infraestrutura no Amazon CloudWatch

Como um serviço gerenciado, o Amazon CloudWatch é protegido pela segurança da rede global da AWS. Para obter informações sobre serviços de segurança da AWS e como a AWS protege a infraestrutura, consulte Segurança na Nuvem AWS. Para projetar seu ambiente da AWS usando as práticas recomendadas de segurança de infraestrutura, consulte Proteção de infraestrutura em Pilar segurança: AWS Well-Architected Framework.

Você usa as chamadas de API publicadas da AWS para acessar o CloudWatch por meio da rede. Os clientes devem oferecer suporte para:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Isolamento de rede

Uma Virtual Private Cloud (VPC) é uma rede virtual na área isolada logicamente na Nuvem Amazon Web Services. Uma sub-rede é um intervalo de endereços IP em uma VPC. Você pode implantar uma variedade de recursos da AWS nas sub-redes de suas VPCs. Por exemplo, é possível implantar instâncias do Amazon EC2, clusters do EMR e tabelas do DynamoDB em sub-redes. Para obter mais informações, consulte o Manual do usuário da Amazon VPC.

Para permitir que o CloudWatch se comunique com recursos em uma VPC sem passar pela internet pública, use o AWS PrivateLink. Para obter mais informações, consulte Usar o CloudWatch e o CloudWatch Synthetics com endpoints da VPC de interface.

Uma sub-rede privada é uma sub-rede sem rota padrão para a Internet pública. A implantação de um recurso da AWS em uma sub-rede privada não impede que o Amazon CloudWatch colete métricas internas do recurso.

Se você precisar publicar métricas personalizadas de um recurso da AWS em uma sub-rede privada, poderá fazê-lo usando um servidor de proxy. O servidor de proxy encaminha essas solicitações HTTPS aos endpoints de API públicos para o CloudWatch.

Security Hub da AWS

Monitore seu uso do CloudWatch em relação às práticas recomendadas de segurança com o AWS Security Hub. O Security Hub usa controles de segurança para avaliar configurações de recursos e padrões de segurança que ajudam você a cumprir vários frameworks de conformidade. Para obter mais informações sobre como usar o Security Hub para avaliar os recursos do CloudFront, consulte Controles do Amazon CloudWatch no Guia do usuário do AWS Security Hub.

Usar o CloudWatch e o CloudWatch Synthetics com endpoints da VPC de interface

Se você usar a Amazon Virtual Private Cloud (Amazon VPC) para hospedar os recursos da AWS, poderá estabelecer uma conexão privada entre a VPC, o CloudWatch e o CloudWatch Synthetics.

É possível usar essas conexões para habilitar o CloudWatch e o CloudWatch Synthetics para se comunicar com os recursos na VPC sem passar pela Internet pública.

A Amazon VPC é um produto da AWS que pode ser utilizado para iniciar os recursos da AWS em uma rede virtual definida por você. Com a VPC, você tem controle sobre as configurações de rede, como o intervalo de endereços IP, sub-redes, tabelas de rotas e gateways de rede. Para conectar a VPC ao CloudWatch ou ao CloudWatch Synthetics, defina um endpoint da VPC de interface para conectar a VPC aos produtos da AWS. O endpoint fornece conectividade confiável e escalável com o CloudWatch ou o CloudWatch Synthetics sem a necessidade de um gateway da Internet, da instância de conversão de endereço de rede (NAT) ou de uma conexão VPN. Para obter mais informações, consulte O que é a Amazon VPC? no Manual do usuário da Amazon VPC.

Os VPC endpoints de interface são desenvolvidos pelo AWS PrivateLink, uma tecnologia da AWS permite a comunicação privada entre os serviços da AWS usando uma interface de rede elástica com endereços IP privados. Para obter mais informações, consulte a publicação de blog New – AWS PrivateLink PrivateLink for AWS Services.

As etapas a seguir são para usuários da Amazon VPC. Para obter mais informações, consulte Conceitos básicos no Guia do usuário da Amazon VPC.

Endpoint da VPC do CloudWatch

No momento, o CloudWatch oferece suporte a endpoints da VPC nas seguintes regiões da AWS:

- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Hong Kong)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)

- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- Oriente Médio (Emirados Árabes Unidos)
- América do Sul (São Paulo)
- AWS GovCloud (Leste dos EUA)
- AWS GovCloud (Oeste dos EUA)

Criar um endpoint da VPC para o CloudWatch

Para começar a usar o CloudWatch na VPC, crie um endpoint da VPC de interface para o CloudWatch. O nome do serviço a ser escolhido é `com.amazonaws.`*`region`*`.monitoring`. Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

Você não precisa alterar as configurações do CloudWatch. O CloudWatch chama outros serviços da AWS usando endpoints públicos ou privados da VPC de interface, o que estiver em uso. Por exemplo, ao criar um endpoint da VPC de interface para o CloudWatch, se você já tiver uma métrica fluindo para o CloudWatch a partir de recursos localizados em sua VPC, essas métricas começarão a fluir por meio do endpoint da VPC de interface por padrão.

Controlar o acesso ao endpoint da VPC do CloudWatch

Uma política de VPC endpoint é uma política de recursos do IAM que você anexa a um endpoint quando cria ou modifica o endpoint. Se você não associar uma política ao criar um endpoint, a Amazon VPC associará uma política padrão que permita o acesso total ao serviço. Uma política de endpoint não substitui políticas de usuário do ou políticas de serviço específicas. É uma política separada para controlar o acesso do endpoint ao serviço especificado.

Políticas de endpoint devem ser gravadas em formato JSON.

Para obter mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Manual do usuário da Amazon VPC.

Veja a seguir um exemplo de uma política de endpoints do CloudWatch. Esta política permite que os usuários se conectem ao CloudWatch por meio da VPC para enviar dados de métrica ao CloudWatch e impede que outras ações do CloudWatch sejam executadas.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Como editar a política de endpoint da VPC para o CloudWatch

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel de navegação, escolha Endpoints.
3. Se você ainda não tiver criado o endpoint para o CloudWatch, escolha Create Endpoint (Criar endpoint). Selecione com.amazonaws.***region***.monitoring e escolha Create endpoint (Criar endpoint).
4. Selecione o endpoint com.amazonaws.***region***.monitoring e escolha a guia Policy (Política).
5. Escolha Edit Policy (Editar política) e faça as alterações.

Endpoint da VPC do CloudWatch Synthetics

No momento, o CloudWatch Synthetics oferece suporte a endpoints da VPC nas seguintes regiões da AWS:

- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Hong Kong)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Seul)

- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- América do Sul (São Paulo)

Criar um endpoint da VPC para o CloudWatch Synthetics

Para começar a usar o CloudWatch Synthetics com a VPC, crie um endpoint da VPC de interface para o CloudWatch Synthetics. O nome do serviço a ser escolhido é `com.amazonaws.region.synthetics`. Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

Você não precisa alterar as configurações do CloudWatch Synthetics. O CloudWatch Synthetics se comunica com outros produtos da AWS usando endpoints da VPC de interface públicos ou privados, o que estiver em uso. Por exemplo, se você criar um endpoint da VPC de interface para o CloudWatch Synthetics e já tiver um endpoint de interface para o Amazon S3, o CloudWatch Synthetics começará a se comunicar com o Amazon S3 por meio do endpoint da VPC de interface por padrão.

Controlar o acesso ao endpoint da VPC do CloudWatch Synthetics

Uma política de VPC endpoint é uma política de recursos do IAM que você anexa a um endpoint quando cria ou modifica o endpoint. Se você não anexar uma política quando criar um endpoint, anexaremos uma política padrão que permita o acesso total ao serviço. Uma política de endpoint não substitui políticas de usuário do ou políticas de serviço específicas. É uma política separada para controlar o acesso do endpoint ao serviço especificado.

As políticas de endpoint afetam os canaries que são gerenciados de forma privada pela VPC. Elas não são necessárias para canaries que são executados em sub-redes privadas.

Políticas de endpoint devem ser gravadas em formato JSON.

Para obter mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Manual do usuário da Amazon VPC.

Veja a seguir um exemplo de política de endpoint para o CloudWatch Synthetics. Essa política permite que os usuários se conectem ao CloudWatch Synthetics por meio da VPC para visualizar informações sobre canaries e suas execuções, mas não para criar, modificar nem excluir canaries.

```
{
    "Statement": [
        {
            "Action": [
                "synthetics:DescribeCanaries",
                "synthetics:GetCanaryRuns"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

Como editar a política de endpoint da VPC para o CloudWatch Synthetics

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel de navegação, escolha Endpoints.
3. Se você ainda não tiver criado o endpoint para o CloudWatch Synthetics, selecione Create Endpoint (Criar endpoint). Selecione com.amazonaws.***region***.synthetics e escolha Create endpoint (Criar endpoint).
4. Selecione o endpoint com.amazonaws.***region***.synthetics e escolha a guia Policy (Política).
5. Escolha Edit Policy (Editar política) e faça as alterações.

Considerações de segurança para canaries do Synthetics

As seções a seguir explicam os problemas de segurança que devem ser levados em conta ao criar e executar canaries no Synthetics.

Usar conexões seguras

Como o código do canário e os resultados das execuções de teste do canário podem conter informações confidenciais, não conecte o seu canário a endpoints por conexões não criptografadas. Sempre use conexões criptografadas, como as que começam com `https://`.

Considerações sobre a nomenclatura de canaries

O nome do recurso da Amazon (ARN) de um canário está incluído no cabeçalho de atendente-usuário como parte das chamadas de saída feitas de um navegador Chromium orientado para Puppeteer que está incluído como parte da biblioteca wrapper do CloudWatch Synthetics. Isso ajuda a identificar o tráfego do canário do CloudWatch Synthetics e relacioná-lo de volta aos canários que estão fazendo chamadas.

O ARN do canário inclui o nome do canário. Escolha nomes do canário que não revelem informações proprietárias.

Além disso, aponte os canaries somente a sites e a endpoints controlados por você.

Segredos e informações sigilosas no código canário

Se você passar seu código canário diretamente para o canário usando um arquivo zip, o conteúdo do script poderá ser visto em logs do AWS CloudTrail.

Se você tiver informações sigilosas ou segredos (como chaves de acesso ou credenciais de banco de dados) em um script do canário, é altamente recomendável armazenar o script como um objeto versionado no Amazon S3 e passar a localização do Amazon S3 para o canário, em vez de passar o código do canário por um arquivo zip.

Se você usar um arquivo zip para transmitir o script do canário, é altamente recomendável não incluir segredos ou informações sigilosas no código-fonte do canário. Para obter mais informações sobre como usar o AWS Secrets Manager para ajudar a manter os segredos seguros, consulte O que é o AWS Secrets Manager?.

Considerações sobre permissões

Recomendamos que você restrinja o acesso aos recursos criados ou usados pelo CloudWatch Synthetics. Use permissões restritas nos buckets do Amazon S3 onde os canaries armazenam os resultados de execução de teste e outros artefatos, como logs e capturas de tela.

Da mesma maneira, mantenha permissões restritas nos locais onde o código-fonte do canário está armazenado para que nenhum usuário acidentalmente ou maliciosamente exclua as camadas do Lambda ou as funções Lambda usadas para o canário.

Para ajudar a garantir que você execute o código do canário pretendido, é possível usar o versionamento de objeto no bucket do Amazon S3 onde o código do canário está armazenado. Então, ao especificar esse código para ser executado como canário, você poderá incluir o objeto `versionId` como parte do caminho, conforme os exemplos a seguir.

```
https://bucket.s3.amazonaws.com/path/object.zip?versionId=version-id
https://s3.amazonaws.com/bucket/path/object.zip?versionId=version-id
https://bucket.s3-region.amazonaws.com/path/object.zip?versionId=version-id
```

Rastreamentos de pilha e mensagens de exceção

Por padrão, os canários do CloudWatch Synthetics capturam qualquer exceção lançada pelo script do canário, independetemente se o script for personalizado ou de um esquema. O CloudWatch Synthetics registra tanto a mensagem de exceção como o rastreamento de pilha em três locais:

- Voltar ao serviço do CloudWatch Synthetics para acelerar a depuração ao descrever execuções de teste
- No CloudWatch Logs, conforme a configuração com as quais as funções do Lambda são criadas
- No arquivo de log do Synthetics, que é um arquivo de texto não criptografado cujo carregamento é feito no local do Amazon S3 especificado pelo valor definido para o `resultsLocation` do canário

Se quiser enviar e armazenar menos informações, será possível capturar exceções antes que elas retornem à biblioteca wrapper do CloudWatch Synthetics.

Também pode haver URLs de solicitação em seus erros. O CloudWatch Synthetics verifica se há URLs no erro gerado pelo script e edita parâmetros de URL restritos deles com base na configuração restrictedUrlParameters. Se estiver registrando mensagens de erro em seu script, poderá usar getSanitizedErrorMessage para redigir URLs antes de registrar.

Definir de forma estrita o escopo das funções do IAM

Recomendamos que você não configure o canário para acessar URLs ou endpoints possivelmente maliciosos. Apontar o canário para sites ou endpoints não confiáveis ou desconhecidos pode expor o código da função Lambda para scripts de usuários maliciosos. Supondo que um site malicioso possa

escapar do Chromium, ele poderia ter acesso ao código do Lambda de maneira semelhante se você estivesse conectado a ele usando um navegador de Internet.

Execute a função Lambda com uma função de execução do IAM com permissões de escopo restrito. Dessa forma, se a função Lambda for comprometida por um script malicioso, ela estará limitada às ações que pode realizar ao ser executada como a conta da AWS do canário.

Ao usar o console do CloudWatch para criar um canário, ele será criado com uma função de execução do IAM de escopo restrito.

Redação de dados sigilosos

O CloudWatch Synthetics captura URLs, código de status, motivo de falha (se houver), cabeçalhos e corpos de solicitações e respostas. Isso permite que um usuário do canário compreenda, monitore e depure canários.

As configurações descritas nas seções a seguir podem ser definidas em qualquer ponto da execução do canário. Também é possível optar por aplicar diferentes configurações a diferentes etapas do Synthetics.

URLs de solicitação

Por padrão, o CloudWatch Synthetics registra URLs de solicitação, códigos de status e o motivo do status de cada URL nos logs do canário. URLs de solicitação também podem aparecer em relatórios de execução do canário, arquivos HAR etc. Sua URL de solicitação pode conter parâmetros de consulta sigilosos, como tokens de acesso ou senhas. É possível redigir informações sigilosas que estejam sendo registradas pelo CloudWatch Synthetics.

Para redigir informações confidenciais, defina a propriedade de configuração restrictedUrlParameters. Para obter mais informações, consulte Classe SyntheticsConfiguration. Isso faz com que o CloudWatch Synthetics edite parâmetros de URL, inclusive valores de parâmetros de caminho e consulta, com base em restrictedUrlParameters antes de registrar. Se estiver registrando URLs em seu script, poderá usar getSanitizedUrl(url, stepConfig = null) para redigir URLs antes de registrar. Para obter mais informações, consulte SyntheticsLogHelper class.

Cabeçalhos

Por padrão, o CloudWatch Synthetics não registra cabeçalhos de solicitação/resposta. Para canaries de interface do usuário, este é o comportamento padrão para canaries que usam a versão de runtime `syn-nodejs-puppeteer-3.2` e posteriores.

Caso seus cabeçalhos não contenham informações sigilosas, será possível habilitar cabeçalhos no arquivo HAR e nos relatórios HTTP definindo as propriedades includeRequestHeaders e includeResponseHeaders como `true`. É possível habilitar todos os cabeçalhos, mas optar por restringir valores de chaves de cabeçalho sigilosos. Por exemplo, você pode optar por apenas redigir cabeçalhos `Authorization` de artefatos produzidos por canaries.

Corpo da solicitação e da resposta

Por padrão, o CloudWatch Synthetics não registra o corpo da solicitação/resposta em logs e relatórios do canário. Essa informação é útil principalmente para canaries da API. O Synthetics captura todas as solicitações HTTP e pode exibir corpos de cabeçalhos, de solicitações e de respostas. Para obter mais informações, consulte executeHttpStep(stepName, requestOptions, [callback], [stepConfig]). Você pode optar por habilitar o corpo da solicitação/resposta definindo as propriedades includeRequestBody e includeResponseBody como `true`.

Registrar chamadas de API do Amazon CloudWatch com o AWS CloudTrail

O Amazon CloudWatch e o CloudWatch Synthetics são integrados ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, uma função ou um produto da AWS. O CloudTrail captura chamadas de API realizadas por sua conta da AWS ou em nome dela. As chamadas capturadas incluem as chamadas do console e as chamadas de código para as operações de API.

Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail a um bucket do S3, incluindo eventos do CloudWatch. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Histórico de eventos. Ao usar as informações coletadas pelo CloudTrail, é possível determinar a solicitação que foi feita ao CloudWatch, o endereço IP do qual a solicitação foi feita, quem a fez e quando ela foi feita, além de outros detalhes.

Para saber mais sobre o CloudTrail, incluindo como configurá-lo e ativá-lo, consulte o AWS CloudTrail Guia do usuário do.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou usuário do IAM AWS Identity and Access Management
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte o Elemento userIdentity do CloudTrail.

Para obter um registro contínuo de eventos em sua conta da AWS, incluindo eventos do CloudWatch e do CloudWatch Synthetics, crie uma trilha. Uma trilha permite que o CloudTrail forneça arquivos de log para um bucket do S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra eventos de todas as regiões na partição da AWS e fornece os arquivos de log ao bucket do S3 que você especificar. Você pode configurar outros serviços da AWS para analisar e atuar mais profundamente sobre os dados de eventos coletados nos logs do CloudTrail. Para mais informações, consulte:

- Visão Geral para Criar uma Trilha
- Serviços e Integrações Compatíveis com CloudTrail
- Configurando Notificações Amazon SNS para CloudTrail
- Receber arquivos de log do CloudTrail de várias regiões e receber arquivos de log do CloudTrail de várias contas

> **Note**
>
> Para obter informações sobre as chamadas de API do CloudWatch Logs que estão registradas no CloudTrail, consulte CloudWatch Logs information in CloudTrail.

Tópicos

- Informações sobre o CloudWatch no CloudTrail
- Monitor de Internet do CloudWatch no CloudTrail
- Informações do CloudWatch Synthetics no CloudTrail

Informações sobre o CloudWatch no CloudTrail

O CloudWatch oferece suporte ao registro das seguintes ações como eventos nos arquivos de log do CloudTrail:

- DeleteAlarms
- DeleteAnomalyDetector
- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- DisableAlarmActions
- EnableAlarmActions
- GetDashboard
- ListDashboards

- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- SetAlarmState

Exemplo: entradas de arquivo de log do CloudWatch

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `PutMetricAlarm`.


```
{
    "Records": [{
        "eventVersion": "1.01",
        "userIdentity": {
            "type": "Root",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID"
        },
        "eventTime": "2014-03-23T21:50:34Z",
        "eventSource": "monitoring.amazonaws.com",
        "eventName": "PutMetricAlarm",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "threshold": 50.0,
            "period": 60,
            "metricName": "CloudTrail Test",
            "evaluationPeriods": 3,
            "comparisonOperator": "GreaterThanThreshold",
            "namespace": "AWS/CloudWatch",
            "alarmName": "CloudTrail Test Alarm",
            "statistic": "Sum"
        },
        "responseElements": null,
        "requestID": "29184022-b2d5-11e3-a63d-9b463e6d0ff0",
        "eventID": "b096d5b7-dcf2-4399-998b-5a53eca76a27"
    },
    ..additional entries
```

```
  ]
  }
```

A entrada do arquivo de log a seguir mostra que um usuário chamou a ação `PutRule` do CloudWatch Events.

```
{
        "eventVersion":"1.03",
        "userIdentity":{
           "type":"Root",
           "principalId":"123456789012",
           "arn":"arn:aws:iam::123456789012:root",
           "accountId":"123456789012",
           "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
           "sessionContext":{
              "attributes":{
                 "mfaAuthenticated":"false",
                 "creationDate":"2015-11-17T23:56:15Z"
              }
           }
        },
        "eventTime":"2015-11-18T00:11:28Z",
        "eventSource":"events.amazonaws.com",
        "eventName":"PutRule",
        "awsRegion":"us-east-1",
        "sourceIPAddress":"AWS Internal",
        "userAgent":"AWS CloudWatch Console",
        "requestParameters":{
           "description":"",
           "name":"cttest2",
           "state":"ENABLED",
           "eventPattern":"{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}",
           "scheduleExpression":""
        },
        "responseElements":{
           "ruleArn":"arn:aws:events:us-east-1:123456789012:rule/cttest2"
        },
        "requestID":"e9caf887-8d88-11e5-a331-3332aa445952",
        "eventID":"49d14f36-6450-44a5-a501-b0fdcdfaeb98",
        "eventType":"AwsApiCall",
        "apiVersion":"2015-10-07",
        "recipientAccountId":"123456789012"
```

```
}
```

A entrada do arquivo de log a seguir mostra que um usuário chamou a ação `CreateExportTask` do CloudWatch Logs.

```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

Monitor de Internet do CloudWatch no CloudTrail

O Monitor de Internet do CloudWatch é compatível com o registro em log das ações a seguir como eventos nos arquivos de log do CloudTrail.

- CreateMonitor
- DeleteMonitor
- GetHealthEvent
- GetMonitor
- GetQueryResults
- GetQueryStatus
- ListHealthEvents
- ListMonitors
- ListTagsForResource
- StartQuery
- StopQuery
- UpdateMonitor

Exemplo: entradas do arquivo de log do Monitor de Internet do CloudWatch

O exemplo a seguir mostra uma entrada de log do CloudTrail Internet Monitor que demonstra a ação `ListMonitors`.


```
{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::000000000000:assumed-role/role_name",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::000000000000:role/Admin",
                "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2022-10-11T17:25:41Z",
```

```
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2022-10-11T17:30:18Z",
        "eventSource": "internetmonitor.amazonaws.com",
        "eventName": "ListMonitors",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)",
        "requestParameters": null,
        "responseElements": null,
        "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
        "readOnly": true,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }
```

O exemplo a seguir mostra uma entrada de log do CloudTrail Internet Monitor que demonstra a ação `CreateMonitor`.

```
{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::000000000000:assumed-role/role_name",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::000000000000:role/Admin",
                "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2022-10-11T17:25:41Z",
```

```
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2022-10-11T17:30:08Z",
        "eventSource": "internetmonitor.amazonaws.com",
        "eventName": "CreateMonitor",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)",
        "requestParameters": {
            "MonitorName": "TestMonitor",
            "Resources": ["arn:aws:ec2:us-east-2:444455556666:vpc/vpc-febc0b95"],
            "ClientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
        },
        "responseElements": {
            "Arn": "arn:aws:internetmonitor:us-east-2:444455556666:monitor/ct-
onboarding-test",
            "Status": "PENDING"
        },
        "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }
```


Informações do CloudWatch Synthetics no CloudTrail

O CloudWatch Synthetics oferece suporte ao registro das seguintes ações como eventos nos arquivos de log do CloudTrail:

- CreateCanary
- DeleteCanary
- DescribeCanaries
- DescribeCanariesLastRun
- DescribeRuntimeVersions
- GetCanary

- GetCanaryRuns
- ListTagsForResource
- StartCanary
- StopCanary
- TagResource
- UntagResource
- UpdateCanary

Exemplo: entradas do arquivo de log do CloudWatch Synthetics

O exemplo a seguir mostra uma entrada de log do CloudTrail Synthetics que demonstra a ação `DescribeCanaries`.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
                "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:47Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "DescribeCanaries",
    "awsRegion": "us-east-1",
```

```
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "201ed5f3-15db-4f87-94a4-123456789",
    "eventID": "73ddbd81-3dd0-4ada-b246-123456789",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

O exemplo a seguir mostra uma entrada de log do CloudTrail Synthetics que demonstra a ação `UpdateCanary`.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
               "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
       "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:47Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "UpdateCanary",
    "awsRegion": "us-east-1",
```

```
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": {
        "Schedule": {
            "Expression": "rate(1 minute)"
        },
        "name": "sample_canary_name",
        "Code": {
            "Handler": "myOwnScript.handler",
            "ZipFile": "SAMPLE_ZIP_FILE"
        }
    },
    "responseElements": null,
    "requestID": "fe4759b0-0849-4e0e-be71-1234567890",
    "eventID": "9dc60c83-c3c8-4fa5-bd02-1234567890",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

O exemplo a seguir mostra uma entrada de log do CloudTrail Synthetics que demonstra a ação `GetCanaryRuns`.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
       "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
```

```
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "GetCanaryRuns",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": {
        "Filter": "TIME_RANGE",
        "name": "sample_canary_name",
        "FilterValues": [
            "2020-04-08T23:00:00.000Z",
            "2020-04-08T23:10:00.000Z"
        ]
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Etiquetar recursos do Amazon CloudWatch

Uma tag é um rótulo de atributo personalizado que você ou a AWS atribui a um recurso da AWS. Cada tag tem duas partes:

- Uma chave de tag (por exemplo `CostCenter`, `Environment` ou `Project`). Chaves de tag fazem distinção entre maiúsculas e minúsculas.
- Um campo opcional conhecido como um valor de tag (por exemplo, `111122223333` ou `Production`). Omitir o valor da tag é o mesmo que usar uma string vazia. Como chaves de tag, os valores das tags diferenciam maiúsculas de minúsculas.

As tags ajudam você a fazer o seguinte:

- Identificar e organizar seus recursos da AWS. Muitos serviços da AWS oferecem suporte à marcação para que você possa atribuir a mesma tag a recursos de diferentes serviços para indicar que os recursos estão relacionados. Por exemplo, é possível atribuir a mesma etiqueta a uma regra do CloudWatch que você atribui a uma instância do EC2.

As seções a seguir apresentam mais informações sobre etiquetas para o CloudWatch.

Recursos compatíveis com o CloudWatch

Os recursos a seguir no CloudWatch permitem a marcação:

- Alarmes: você pode marcar alarmes usando o comando tag-resource da AWS CLI e a API TagResource. Também é possível visualizar e gerenciar suas tags de alarme usando a página de detalhes Alarmes no console do CloudWatch.
- Canaries: é possível etiquetar canaries usando o console do CloudWatch. Para obter mais informações, consulte Criar um canário.
- Regras do Contributor Insights: você pode marcar regras do Contributor Insights ao criá-las usando o comando put-insight-rule da AWS CLI e a API PutInsightRule. Você pode adicionar tags a regras existentes usando o comando tag-resource da AWS CLI e a API TagResource .
- Fluxos de métrica: é possível etiquetar fluxos de métrica ao criá-los usando o comando put-metric-stream da AWS CLI e o comandoPutMetricStream da API. Você pode adicionar etiquetas a fluxos de métricas existentes usando o comando tag-resource da AWS CLI e a API TagResource.

Para obter informações sobre como adicionar e gerenciar tags, consulte Gerenciar tags.

Gerenciar tags

As tags consistem nas propriedades `Value` e `Key` em um recurso. Você pode usar o console do CloudWatch, a AWS CLI ou a API do CloudWatch para adicionar, editar ou excluir os valores dessas propriedades. Para obter informações sobre como trabalhar com tags, consulte o seguinte:

- TagResource, UntagResource e ListTagsForResource na Referência da API do Amazon CloudWatch
- tag-resource, untag-resource e list-tags-for-resource na Referência da CLI do Amazon CloudWatch
- Trabalhar com o editor de tags no Manual do usuário do Resource Groups

Convenções de uso e nomenclatura de tags

As seguintes convenções básicas de uso e nomenclatura se aplicam ao uso de etiquetas com recursos do CloudWatch:

- Cada recurso pode ter um máximo de 50 tags.
- Em todos os recursos, cada chave de tag deve ser exclusiva e pode ter apenas um valor.
- O comprimento máximo da chave da tag é de 128 caracteres Unicode em UTF-8.
- O comprimento máximo do valor da tag é de 256 caracteres Unicode em UTF-8.
- Os caracteres permitidos são letras, números, espaços representáveis em UTF-8, além dos seguintes caracteres: . : + = @ _ / - (hífen).
- As chaves e os valores de tags diferenciam maiúsculas de minúsculas. Como melhor prática, decida-se sobre uma estratégia para letras maiúsculas em tags e implemente-a de forma consistente em todos os tipos de recursos. Por exemplo, decida se deseja usar `Costcenter`, `costcenter` ou `CostCenter` e use a mesma convenção para todas as tags. Evite usar tags semelhantes com tratamento do tamanho de letra inconsistente.
- O prefixo `aws:` é proibido em tags, pois ele é reservado para uso pela AWS. Você não pode editar nem excluir chaves nem valores de etiquetas com esse prefixo. As tags com esse prefixo não contam para as tags por limite de recurso.

Integração ao Grafana

É possível usar o Grafana versão 6.5.0 e posteriores para avançar contextualmente pelo console do CloudWatch e consultar uma lista dinâmica de métricas usando curingas. Isso pode ajudar você a monitorar métricas de recursos da AWS, como instâncias ou contêineres do Amazon Elastic Compute Cloud. Quando novas instâncias são criadas como parte de um evento Auto Scaling, elas aparecem no gráfico automaticamente. Você não precisa rastrear os novos IDs de instância. Os painéis pré-integrados ajudam a simplificar a experiência de começar a usar o monitoramento de recursos do Amazon EC2, do Amazon Elastic Block Store e do AWS Lambda.

Você pode usar o Grafana versão 7.0 e posteriores para executar consultas do CloudWatch Logs Insights em grupos de logs no CloudWatch Logs. É possível visualizar os resultados da consulta em gráficos de barras, linhas e barras empilhadas, bem como em um formato de tabela. Para obter mais informações sobre o CloudWatch Logs Insights, consulte Analisar dados de log com o CloudWatch Logs Insights.

Para obter mais informações sobre como começar a usar, consulte Usar o AWS CloudWatch no Grafana na documentação do Grafana Labs.

Console do CloudWatch entre contas e entre regiões

Para ter a mais rica experiência de observabilidade e descoberta entre contas para suas métricas, logs e rastreamentos, recomendamos que você use a observabilidade entre contas do CloudWatch. Para obter mais informações, consulte Observabilidade entre contas do CloudWatch.

O CloudWatch também oferece um painel do CloudWatch entre contas e regiões. Essa funcionalidade fornece visibilidade entre contas para painéis, alarmes, métricas e painéis automáticos. Ela não fornece visibilidade entre contas para logs ou para rastreamentos.

Se você também estiver usando a observabilidade entre contas do CloudWatch, um caso de uso desse painel do CloudWatch entre contas é permitir que uma de suas contas de origem da observabilidade entre contas do CloudWatch veja as métricas de outra conta de origem.

O restante desta seção descreve o painel entre contas e entre regiões. Você pode criar painéis que resumem os dados do CloudWatch de várias contas da AWS e várias regiões da AWS em um único painel. Também é possível criar um alarme em uma conta que observe uma métrica localizada em uma conta diferente.

Muitas organizações têm seus recursos da AWS implantados em várias contas, para fornecer limites de faturamento e segurança. Nesse caso, recomendamos que você designe uma ou mais de suas contas como suas contas de monitoramento e crie seus painéis entre essas contas.

A funcionalidade entre contas é integrado com o AWS Organizations, para ajudar a criar de forma eficiente seus painéis entre contas.

Funcionalidade entre regiões

A funcionalidade entre regiões agora é incorporada automaticamente. Você não precisa seguir nenhum passo adicional para poder exibir métricas de regiões diferentes em uma única conta no mesmo gráfico ou no mesmo painel. A funcionalidade entre regiões não é compatível com alarmes. Portanto, você não pode criar um alarme em uma região que observe uma métrica em uma região diferente.

Tópicos

- Desabilitar e limpar depois de usar contas cruzadas

Habilitar a funcionalidade entre contas no CloudWatch

Para configurar a funcionalidade entre contas no console do CloudWatch, use o console do CloudWatch para configurar suas contas de compartilhamento e contas de monitoramento.

Configurar uma conta de compartilhamento

Você deve habilitar o compartilhamento em cada conta que disponibilizará os dados para a conta de monitoramento.

Isso concederá as permissões somente para leitura escolhidas na etapa 5 a todos os usuários que visualizarem um painel entre contas na conta com a qual você compartilha, se o usuário tiver permissões correspondentes nessa conta.

Como permitir que sua conta compartilhe dados do CloudWatch com outras contas

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Configurações.
3. Em Share your CloudWatch data (Compartilhar dados do CloudWatch), escolha Configure (Configurar).
4. Em Sharing (Compartilhamento), escolha Specific accounts (Contas específicas) e insira os IDs das contas com as quais você deseja compartilhar dados.

   Todas as contas especificadas aqui poderão visualizar os dados do CloudWatch de sua conta. Especifique os IDs somente de contas que você conhece e nas quais confia.
5. Em Permissions (Permissões), especifique como compartilhar seus dados com uma das seguintes opções:

   - Provide read-only access to your CloudWatch metrics, dashboards, and alarms (Fornecer acesso somente leitura a métricas, painéis e alarmes do CloudWatch). Essa opção permite que as contas de monitoramento criem painéis entre contas que incluam widgets com dados do CloudWatch a partir de sua conta.
   - Incluir painéis automáticos do CloudWatch. Se você selecionar essa opção, os usuários na conta de monitoramento também poderão exibir as informações nos painéis automáticos dessa conta. Para obter mais informações, consulte Conceitos básicos do Amazon CloudWatch.

- Incluir acesso somente leitura do X-Ray para o mapa de rastreamento do X-Ray. Se você selecionar essa opção, os usuários na conta de monitoramento também poderão visualizar o mapa de rastreamento do X-Ray e as informações de rastreamento do X-Ray nessa conta. Para obter mais informações, consulte Using the X-Ray Trace Map.
- Full read-only access to everything in your account (Acesso total somente leitura a tudo o que estiver na sua conta). Essa opção permite que as contas usadas para compartilhamento criem painéis entre contas que incluam widgets com dados do CloudWatch de sua conta. Também permite que essas contas investiguem mais profundamente a sua conta e visualizem os dados da sua conta nos consoles de outros produtos da AWS.

6. Escolha Launch CloudFormation template (Iniciar modelo CloudFormation).

   Na tela de confirmação, digite **`Confirm`** e escolha Launch template (Iniciar modelo).

7. Marque a caixa de seleção I acknowledge... (Aceito...) e escolha Create stack (Criar pilha).

Compartilhar com uma organização inteira

Ao concluir o procedimento anterior, você cria uma função do IAM que permite que sua conta compartilhe dados com uma conta. Você pode criar ou editar uma função do IAM que compartilhe seus dados com todas as contas em uma organização. Faça isso somente se você conhece e confia em todas as contas da organização.

Isso concederá as permissões somente para leitura ouvidas pelas políticas exibidas na etapa 5 do procedimento anterior a todos os usuários que visualizarem um painel entre contas na conta com a qual você compartilha, se o usuário tiver permissões correspondentes nessa conta.

Como compartilhar os dados da conta do CloudWatch com todas as contas de uma organização

1. Caso ainda não tenha feito isso, conclua o procedimento anterior para compartilhar seus dados com uma conta da AWS.
2. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
3. No painel de navegação, escolha Perfis.
4. Na lista de funções, escolha CloudWatch-CrossAccountSharingRole.
5. Escolha Trust relationships (Relacionamentos de confiança), Edit trust relationship (Editar relacionamento de confiança).

   A política é semelhante a esta:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. Altere a política para o seguinte, substituindo *org-id* pelo ID da organização.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "org-id"
        }
      }

    }
  ]
}
```

7. Escolha Update Trust Policy.

Configurar uma conta de monitoramento

Habilite cada conta de monitoramento para exibir dados entre contas do CloudWatch.

Quando você concluir o procedimento a seguir, o CloudWatch criará uma função vinculada ao serviço que o CloudWatch usará na conta de monitoramento para acessar

dados compartilhados de outras contas. Essa função vinculada ao serviço é chamada AWSServiceRoleForCloudWatchCrossAccount. Para obter mais informações, consulte Usar funções vinculadas ao serviço para o CloudWatch.

Como permitir que sua conta visualize dados entre contas do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Settings (Configurações) e, na seção Cross-account cross-region (Contas cruzadas entre regiões), escolha Configure (Configurar).
3. Na seção View cross-account cross-region (Ver contas cruzadas entre regiões), escolha Enable (Habilitar) e marque a caixa de seleção Show selector in the console (Exibir o seletor no console) para permitir que um seletor de conta seja visualizado no console do CloudWatch quando você estiver representando grafos de uma métrica ou criando um alarme.
4. Em View cross-account cross-region (Exibir entre contas e entre regiões), escolha uma das seguintes opções:

   - Account Id Input (Entrada de ID da conta). Essa opção solicita que você insira manualmente um ID de conta sempre que deseja alternar contas ao exibir dados entre contas.
   - Seletor de contas do AWS Organization. Essa opção faz com que as contas especificadas ao concluir a integração entre contas com o Organizations sejam exibidas. Na próxima vez em que você usar o console, o CloudWatch exibirá uma lista suspensa dessas contas para seleção quando estiver exibindo dados entre contas.

     Para fazer isso, você deve primeiro ter usado sua conta de gerencimaneto da organização a fim de permitir que o CloudWatch visualize uma lista de contas em sua organização. Para obter mais informações, consulte (Opcional) Integrar com o AWS Organizations.
   - Custom account selector (Seletor de contas personalizado). Essa opção solicita que você insira uma lista de IDs de contas. Na próxima vez em que você usar o console, o CloudWatch exibirá uma lista suspensa dessas contas para seleção quando estiver exibindo dados entre contas.

     Você também pode inserir um rótulo para cada uma dessas contas a fim de ajudar a identificá-las ao escolher contas para exibição.

     As configurações do seletor de conta que um usuário faz aqui são mantidas somente para esse usuário, não para todos os outros usuários na conta de monitoramento.
5. Escolha Habilitar.

Depois de concluir essa configuração, você poderá criar painéis entre contas. Para obter mais informações, consulte Painéis entre contas e entre regiões.

(Opcional) Integrar com o AWS Organizations

Se você quiser integrar a funcionalidade entre contas com o AWS Organizations, deverá disponibilizar uma lista de todas as contas da organização para as contas de monitoramento.

Como habilitar a funcionalidade do CloudWatch entre contas a fim de acessar uma lista de todas as contas de sua organização

1. Faça login na conta de gerenciamento de sua organização.
2. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação, escolha Settings (Configurações) e, depois, Configure (Configurar).
4. Em Grant permission to view the list of accounts in the organization (Conceder permissão para exibir a lista de contas da organização), escolha Specific accounts (Contas específicas) que deverão inserir uma lista de IDs de conta. A lista de contas da organização será compartilhada somente com as contas especificadas aqui.
5. Escolha Share organization account list (Compartilhar lista de contas da organização).
6. Escolha Launch CloudFormation template (Iniciar modelo CloudFormation).

   Na tela de confirmação, digite **`Confirm`** e escolha Launch template (Iniciar modelo).

Solucionar problemas de configuração entre contas do CloudWatch

Esta seção contém dicas de solução de problemas para implantação do console entre contas no CloudWatch.

Estou recebendo erros de acesso negado exibindo dados entre contas

Verifique o seguinte:

- Sua conta de monitoramento deve ter uma função chamada AWSServiceRoleForCloudWatchCrossAccount. Caso contrário, você precisa criar essa função. Para obter mais informações, consulte Set Up a Monitoring Account.
- Cada conta de compartilhamento deve ter uma função chamada CloudWatch-CrossAccountSharingRole. Caso contrário, você precisa criar essa função. Para obter mais informações, consulte Set Up A Sharing Account.

- A função de compartilhamento deve confiar na conta de monitoramento.

Para confirmar se suas funções estão configuradas corretamente para o console entre contas do CloudWatch

1. Faça login no AWS Management Console e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Perfis.
3. Na lista de funções, verifique se a função necessária existe. Em uma conta de compartilhamento, procure CloudWatch-CrossAccountSharingRole. Em uma conta de monitoramento, procure AWSServiceRoleForCloudWatchCrossAccount.
4. Se você estiver em uma conta de compartilhamento e a CloudWatch-CrossAccountSharingRole já existir, escolha CloudWatch-CrossAccountSharingRole.
5. Escolha Trust relationships (Relacionamentos de confiança), Edit trust relationship (Editar relacionamento de confiança).
6. Confirme se a política lista o ID da conta de monitoramento ou o ID de uma organização que contenha a conta de monitoramento.

Não vejo uma lista suspensa de contas no console

Primeiro, verifique se você criou as funções do IAM corretas, conforme discutido na seção de solução de problemas anterior. Se elas estiverem configuradas corretamente, verifique se ativou essa conta para exibir dados entre contas, conforme descrito em Enable Your Account to View Cross-Account Data.

Desabilitar e limpar depois de usar contas cruzadas

Para desabilitar a funcionalidade de entre contas do CloudWatch, siga estas etapas.

Etapa 1: remover as pilhas ou os perfis entre contas

O melhor método é remover as pilhas AWS CloudFormation que foram usadas para habilitar a funcionalidade entre contas.

- Em cada uma das contas de compartilhamento, remova a pilha CloudWatch-CrossAccountSharingRole.

- Se você usou o AWS Organizations para habilitar a funcionalidade entre contas em todas as contas de uma organização, remova a pilha CloudWatch-CrossAccountListAccountsRole da conta de gerenciamento da organização.

Se você não usou as pilhas do AWS CloudFormation para habilitar a funcionalidade entre contas, faça o seguinte:

- Em cada uma das contas de compartilhamento, exclua a função do IAM CloudWatch-CrossAccountSharingRole.
- Se você usou o AWS Organizations para habilitar a funcionalidade entre contas em todas as contas de uma organização, remova a função do IAM CloudWatch-CrossAccountSharing-ListAccountsRole da conta de gerenciamento da organização.

Etapa 2: remover o perfil vinculado ao serviço

Na conta de monitoramento, exclua a função do IAM vinculada ao serviço AWSServiceRoleForCloudWatchCrossAccount.

Cotas de serviço do CloudWatch

O CloudWatch tem as cotas a seguir de métricas, alarmes, solicitações de API e notificações de e-mail de alarme.

> **Note**
>
> Em alguns serviços da AWS, incluindo o CloudWatch, é possível usar as métricas de uso do CloudWatch para visualizar o uso atual do serviço nos gráficos e painéis do CloudWatch. É possível usar uma função matemática métrica do CloudWatch para exibir as cotas de serviço desses recursos nos gráficos. Também é possível configurar alarmes que alertam você quando o uso se aproxima de uma cota de serviço. Para ter mais informações, consulte Visualizar as Service Quotas e definir alarmes.

Recurso	Cota padrão
Ações de alarme	5/alarme. Essa cota não pode ser alterada.
Período de avaliação do alarme	O valor máximo, calculado multiplicando-se o período de alarme pelo número de períodos de avaliação utilizados, é de um dia (86.400 segundos). Essa cota não pode ser alterada.
Alarmes	10/mês/cliente gratuitamente. Alarmes adicionais incorrerão em cobranças. Não há limite para o total de alarmes por conta. Os alarmes baseados em expressões matemáticas de métricas podem ter até 10 métricas. 200 alarmes do Metrics Insights por Região. É possível solicitar um aumento da cota.
Modelos de detecção de anomalias	500 por região, por conta.
Solicitações de API	1.000.000/mês/cliente gratuitamente.

Recurso	Cota padrão
Canaries	200 por região, por conta. É possível solicitar um aumento da cota.
Solicitações de API do Contributor Insights	As APIs a seguir têm uma cota de 20 transações por segundo (TPS) por região. • DescribeInsightRules Essa cota não pode ser alterada. • GetInsightRuleReport É possível solicitar um aumento da cota. As APIs a seguir têm uma cota de 5 TPS por região. Essa cota não pode ser alterada. • DeleteInsightRules • PutInsightRule As APIs a seguir têm uma cota de 1 TPS por região. Essa cota não pode ser alterada. • DisableInsightRules • EnableInsightRules
Regras do Contributor Insights	100 regras por região, por conta. É possível solicitar um aumento da cota.
Métricas personalizadas	Sem cota.

Recurso	Cota padrão
Painéis	Até 500 widgets por painel. Até 500 métricas por widget de painel. Até 2500 métricas por painel, entre todos os widgets. Essas cotas incluem todas as métricas recuperadas para uso em funções matemáticas, mesmo que essas métricas não sejam exibidas no gráfico. Essas cotas não podem ser alteradas.
DescribeAlarms	9 transações por segundo (TPS), por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. É possível solicitar um aumento da cota.
Solicitação DeleteAlarms Solicitação DescribeAlarmHistory Solicitação DisableAlarmActions Solicitação EnableAlarmActions Solicitação SetAlarmState	3 TPS por região para cada uma destas operações. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. Essas cotas não podem ser alteradas.
Solicitação DescribeAlarmsForMetric	9 TPS por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. Essas cotas não podem ser alteradas.
Solicitação DeleteDashboards SolicitaçãoGetDashboard SolicitaçãoListDashboards SolicitaçãoPutDashboard	10 TPS por região para cada uma destas operações. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. Essas cotas não podem ser alteradas.

Recurso	Cota padrão
PutAnomalyDetector DescribeAnomalyDetectors	10 TPS por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado.
DeleteAnomalyDetector	5 TPS por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado.
Dimensões	30/métrico. Essa cota não pode ser alterada.

Recurso	Cota padrão
GetMetricData	10 TPS por região para operações que incluem consultas do Metrics Insights. Para operações que não incluem consultas do Metrics Insights, a cota é de 50 TPS por região. Esse é o número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. É possível solicitar um aumento da cota. Para operações `GetMetricData` que incluem uma consulta do Metrics Insights, a cota é de 4.300.000 pontos de dados por segundo (DPS) para as 3 horas mais recentes. Calcula-se isso relativamente ao número total de pontos de dados verificados pela consulta (que não pode incluir mais de 10.000 métricas). 180.000 Datapoints por segundo (DPS) se o `StartTime` usado na solicitação da API for menor que ou igual a três horas a partir da hora atual. 396.000 DPS se o `StartTime` for mais que três horas a partir da hora atual. Esse é o número máximo de datapoints que você pode solicitar por segundo usando uma ou mais chamadas à API sem ser limitado. Essa cota não pode ser alterada. O DPS é calculado com base em pontos de dados estimados, não em pontos de dados reais. A estimativa do ponto de dados é calculada usando o intervalo de tempo, o período e o período de retenção solicitados. Isso significa que, se os pontos de dados reais nas métricas solicitadas forem esparsos ou vazios, a limitação ainda ocorrerá se os pontos de dados estimados excederem a cota. A cota DPS é por região.

Recurso	Cota padrão
GetMetricData	Uma única chamada `GetMetricData` pode incluir o seguinte: • No máximo 500 estruturas `MetricDataQuery` . • No máximo 100 funções `SERVICE_QUOTA()` . • No máximo 100 funções `SEARCH()`. • No máximo 5 funções `LAMBDA()`. Essas cotas não podem ser alteradas.
GetMetricStatistics	400 TPS por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. É possível solicitar um aumento da cota.
GetMetricWidgetImage	Até 500 métricas por imagem. Essa cota não pode ser alterada. 20 TPS por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. É possível solicitar um aumento da cota.
ListMetrics	25 TPS por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. É possível solicitar um aumento da cota.
Valores de dados de métricas	O valor de um ponto de dados de métricas deve estar dentro do intervalo de -2^360 a 2^360. Valores especiais (por exemplo, NaN, +Infinity, -Infinity) não são compatíve is. Essa cota não pode ser alterada.

Recurso	Cota padrão
Itens MetricDatum	1.000/solicitações PutMetricData. Um objeto MetricDatum pode conter um único valor ou um objeto StatisticSet que represente muitos valores. Essa cota não pode ser alterada.
Metrics	10/mês/cliente gratuitamente.
Consultas do Metrics Insights	Uma consulta única não pode processar mais de 10.000 métricas. Isto significa que, se as cláusulas SELECT, FROM e WHERE corresponderem a mais de 10.000 métricas, apenas as primeiras 10.000 dessas métricas encontradas serão processadas pela consulta. Uma consulta única não pode retornar mais de 500 séries temporais. Você só pode consultar os dados das três últimas horas.
Taxas de solicitação da API Observability Access Manager (OAM).	1 TPS por região para PutSinkPolicy. 10 TPS por região para cada outra API OAM do CloudWatch. O número máximo de solicitações de operação que podem ser feitas por segundo sem sofrer controle de utilização. Essas cotas não podem ser alteradas.
Links da conta de origem de OAM	Cada conta de origem pode ser vinculada a até 5 contas de monitoramento. Não é possível alterar esta cota.
Coletores de OAM	1 coletor por região por conta Não é possível alterar esta cota.

Recurso	Cota padrão
Solicitação PutCompositeAlarm	3 TPS por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. É possível solicitar um aumento da cota.
Solicitação PutMetricAlarm	3 TPS por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. É possível solicitar um aumento da cota.
Solicitação PutMetricData	1 MB para solicitações HTTP POST. O PutMetricData pode lidar com 500 transações por segundo (TPS), que é o número máximo de solicitações de operação que você pode fazer por segundo sem que haja controle de utilização. O `PutMetricData` pode lidar com mil métricas por solicitação. É possível solicitar um aumento da cota.
Notificações por e-mail do Amazon SNS	1.000/mês/cliente gratuitamente.
Grupos sintéticos	20 por conta. Não é possível alterar esta cota.
TagResource	20 TPS por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. Não é possível alterar esta cota.

Recurso	Cota padrão
UntagResource	20 TPS por região. O número máximo de solicitações de operação que você pode fazer por segundo sem ser limitado. Não é possível alterar esta cota.

Histórico do documento

A tabela a seguir descreve as alterações importantes em cada versão do Manual do usuário do Amazon CloudWatch a partir de junho de 2018. Para receber notificações sobre atualizações dessa documentação, você poderá se inscrever em um feed RSS.

Alteração	Descrição	Data
O mapa de serviços do CloudWatch Application Signals é compatível com canário, clientes do RUM e agrupamentos de dependências de serviços da AWS.	A versão prévia do Application Signals adicionou agrupamentos padrão no mapa de serviços para canários, clientes do RUM e dependências de serviços da AWS que são de tipos semelhantes. Essa alteração reduz o número de ícones na visualização padrão do mapa de serviços para facilitar a visualização e a navegação.	21 de maio de 2024
Atualização da política do IAM CloudWatchReadOnlyAccess	O CloudWatch alterou o escopo de uma permissão em CloudWatchReadOnlyAccess. O escopo da política adicionou as ações `application-signals:BatchGet*` , `application-signals:Get*` e `application-signals:List*` com a finalidade de que os usuários possam usar o CloudWatch Application Signals para visualizar, investigar e diagnosticar problemas relacionados	17 de maio de 2024

	com a integridade de seus serviços. O CloudWatch também adicionou uma ação `iam:GetRole` para que os usuários possam verificar se o Application Signals está configurado.	
Atualização da política do IAM CloudWatchFullAccessV2	O CloudWatch alterou o escopo de uma permissão em CloudWatchFullAccessV2. O escopo da política adicionou `application-signals:*` com a finalidade de que os usuários possam usar o CloudWatch Application Signals para visualizar, investigar e diagnosticar problemas relacionados com a integridade de seus serviços.	17 de maio de 2024
Lambda Insights tem suporte para AWS GovCloud (Leste dos EUA) e para AWS GovCloud (Oeste dos EUA)	O CloudWatch Lambda Insights adicionou suporte para as regiões AWS GovCloud (Leste dos EUA) e AWS GovCloud (Oeste dos EUA).	29 de abril de 2024
Observabilidade entre contas do CloudWatch compatível com filtros de recursos	Agora é possível criar filtros para especificar quais namespaces de métricas e grupos de logs são compartilhados da conta de origem para a conta de monitoramento ao criar o vínculo entre as contas.	26 de abril de 2024

Atualizações do CloudWatch Application Signals	A versão prévia do Application Signals adicionou três recursos. Agora, o Application Signals é compatível com aplicações em Python. Ele oferece um processo de habilitação mais simples para aplicações em arquiteturas do Amazon EKS. Além disso, ele inclui novas configurações que podem ser usadas para gerenciar a cardinalidade das métricas coletadas.	26 de abril de 2024
CloudWatch Container Insights com observabilidade aprimorada para o Amazon EKS pode coletar métricas do AWS Elastic Fabric Adapter (EFA)	Agora, é possível usar o CloudWatch Container Insights com observabilidade aprimorada para o Amazon EKS para coletar métricas do AWS Elastic Fabric Adapter (EFA) de clusters do Amazon EKS.	23 de abril de 2024

Atualizada a política do IAM	O CloudWatch atualizou a política CloudWatchApplicationSignalsServiceRolePolicy. O escopo das permissões `logs:StartQuery` e `logs:GetQueryResults`, nesta política, foi alterado para adicionar `arn:aws:logs:*:*:log-group:/aws/appsignals/*:*` e `"arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"` com a finalidade de habilitar o Application Signals em mais arquiteturas. Esta política está anexada ao perfil vinculado ao serviço AWSServiceRoleForCloudWatchApplicationSignals.	18 de abril de 2024
Monitor de Internet fornece um mapa de condições globais da Internet para clientes autenticados da AWS	Agora, o Monitor de Internet do Amazon CloudWatch exibe um mapa de condições globais da Internet que está disponível no console para todos os clientes autenticados da AWS. Para visualizar o mapa, no console do Amazon CloudWatch, navegue até o Monitor de Internet.	16 de abril de 2024

CloudWatch Container Insights com observabilidade aprimorada para o Amazon EKS pode coletar métricas do AWS Neuron	Agora, é possível usar o CloudWatch Container Insights com observabilidade aprimorada para o Amazon EKS para coletar métricas do AWS Neuron de clusters do Amazon EKS.	16 de abril de 2024
CloudWatch Application Signals adiciona uma guia de Visão geral do serviço e mais métricas para auxiliar no diagnóstico	Uma nova guia Visão geral do serviço exibe uma visão geral do seu serviço, incluindo o número de operações, as dependências, os sintético s e as páginas de clientes. A guia mostra as principais métricas de todo o seu serviço e as principais operações e dependências. Além disso, agora é possível visualiza r rastreamentos do X-Ray correlacionados a problemas , incluindo falhas, erros e problemas de latência.	16 de abril de 2024
CloudWatch Container Insights com observabilidade aprimorada para o Amazon EKS adiciona suporte para o sistema Windows	Agora é possível usar o CloudWatch Container Insights com observabilidade aprimorada para o Amazon EKS para coletar métricas de nós de processamento do Windows em clusters do Amazon EKS.	10 de abril de 2024

Atualização da política do IAM CloudWatchApplicationSignalsServiceRolePolicy	O CloudWatch alterou o escopo de uma permissão em CloudWatchApplicationSignalsServiceRolePolicy. O escopo da permissão `cloudwatch:GetMetricData` foi alterado para `*` com a finalidade de que o Application Signals possa recuperar métricas de origens em contas vinculadas.	8 de abril de 2024
O Monitor de Internet do Amazon CloudWatch agora oferece suporte à observabilidade entre contas	Agora você pode usar a observabilidade entre contas do Monitor de Internet para monitorar suas aplicações que abrangem várias Contas da AWS em uma única Região da AWS.	29 de março de 2024

As políticas CloudWatchAgentServerPolicy e CloudWatchAgentAdminPolicy foram atualizadas	O CloudWatch adicionou permissões às políticas `CloudWatchAgentServerPolicy` e `CloudWatchAgentAdminPolicy` para permitir que o agente do CloudWatch publique rastreamentos do X-Ray e modifique os períodos de retenção do grupo de logs. Nas duas políticas, as permissões `xray:PutTraceSegments`, `xray:PutTelemetryRecords`, `xray:GetSamplingRules`, `xray:GetSamplingTargets`, `xray:GetSamplingStatisticSummaries` e `logs:PutRetentionPolicy` foram adicionadas	12 de fevereiro de 2024

Novo perfil vinculado ao serviço e à política do IAM para o CloudWatch Network Monitor	O CloudWatch adicionou um novo perfil vinculado ao serviço, denominado AWSServiceRoleForNetworkMonitor. O CloudWatch adicionou esse novo perfil vinculado ao serviço para permitir que você crie monitores para buscar métricas de rede entre sub-redes de origem e endereços IP de destino. A nova política do IAM CloudWatchNetworkMonitorServiceRolePolicy está anexada a esse perfil e ela concede permissão ao CloudWatch para buscar métricas de rede em seu nome.	22 de dezembro de 2023
CloudWatch lança o Amazon CloudWatch Network Monitor	O CloudWatch lançou um novo recurso, o Amazon CloudWatch Network Monitor. Esse é um novo serviço ativo de monitoramento de rede que identifica se existe um problema de rede na rede da AWS ou na rede da sua própria empresa.	22 de dezembro de 2023

Atualização da política CloudWatchReadOnlyAccess	O CloudWatch adicionou permissões somente leitura atuais para o CloudWatch Synthetics, X-Ray e CloudWatch RUM e novas permissões somente leitura para o CloudWatch Application Signals a CloudWatchReadOnlyAccess para que os usuários com essa política possam fazer a triagem e diagnosticar problemas da integridade do serviço relatados pelo CloudWatch Application Signals. A permissão `cloudwatch:GenerateQuery` foi adicionada para que os usuários com essa política possam gerar uma string de consulta do CloudWatch Metrics Insights de uma solicitação em linguagem natural.	5 de dezembro de 2023

Política CloudWatchFullAccessV2 atualizada	O CloudWatch adicionou permissões atuais a CloudWatchFullAccessV2 para o CloudWatch Synthetics, X-Ray e CloudWatch RUM e adicionou novas permissões para o CloudWatch Application Signals para que os usuários com essa política possam gerenciar por completo o Application Signals para fazer a triagem e diagnosticar problemas com a integridade do serviço.	5 de dezembro de 2023

Novo perfil vinculado ao serviço e nova política do IAM	O CloudWatch adicionou um novo perfil vinculado ao serviço, denominado AWSServiceRoleForCloudWatchApplicationSignals. O CloudWatch adicionou esse novo perfil vinculado ao serviço para permitir que o CloudWatch Application Signals colete dados do CloudWatch Logs, dados de rastreamento do X-Ray, dados de métricas do CloudWatch e dados de marcação de aplicações que você habilitou para o CloudWatch Application Signals. A nova política do IAM CloudWatchApplicationSignalsServiceRolePolicy é anexada a esse perfil e ela concede permissão ao CloudWatch Application Signals para coletar dados de monitoramento e marcação de outros serviços relevantes da AWS.	30 de novembro de 2023

CloudWatch lança a versão de pré-visualização do Application Signals	O CloudWatch Application Signals está na versão de pré-visualização. Use o Application Signals para instrumentar aplicações na AWS para que você possa monitorar a integridade atual das aplicações, criar objetivos de nível de serviço (SLOs) e rastrear a performance de longo prazo das aplicações em relação aos seus objetivos de negócios. Para obter mais informações, consulte Application Signals.	30 de novembro de 2023
CloudWatch adiciona suporte para consultar outras fontes de dados	Você pode usar o CloudWatch para consultar, visualizar e criar alarmes para métricas de outras fontes de dados. Para obter mais informações, consulte Querying metrics from other data sources.	26 de novembro de 2023
CloudWatch Metrics Insights compatível com a geração de consultas em linguagem natural	O CloudWatch Metrics Insights é compatível com consultas em linguagem natural para gerar e atualizar consultas. Para obter mais informações, consulte Use natural language to generate and update CloudWatch Metric Insights queries.	26 de novembro de 2023

CloudWatch lança o Container Insights com observabilidade aprimorada para o Amazon EKS	O CloudWatch lançou uma nova versão do Container Insights. Essa versão é compatível com a observabilidade aprimorada dos clusters do Amazon EKS e pode coletar métricas mais detalhadas dos clusters que executam o Amazon EKS. Após a instalação, ela coleta automaticamente a telemetria detalhada da infraestrutura e os registros de contêineres dos clusters do Amazon EKS. Em seguida, é possível usar painéis selecionados e imediatamente utilizáveis para detalhar a telemetria de aplicações e infraestrutura.	6 de novembro de 2023
Fluxos de métricas do CloudWatch adicionam uma configuração rápida de parceiros	Os fluxos de métricas do CloudWatch agora oferecem uma opção de configuração rápida de parceiros, a qual você pode usar para configurar rapidamente um fluxo de métricas para alguns provedores externos.	17 de outubro de 2023

CloudWatch lança recomendações de alarmes	O CloudWatch Synthetics agora fornece recomendações de alarme para métricas de outros serviços da AWS. Essas recomendações podem ajudar você a identificar as métricas para as quais você deve definir alarmes para seguir as práticas recomendadas de monitoramento desses serviços.	16 de outubro de 2023
O CloudWatch Synthetics lança o runtime syn-nodejs-puppeteer-6.0	O CloudWatch Synthetics lançou o runtime `syn-nodejs-puppeteer-6.0` .	26 de setembro de 2023
Foi adicionado suporte ao Amazon CloudWatch Application Insights para aplicações entre contas	Agora é possível compartilhar as aplicações do CloudWatch Application Insights além dos limites das contas.	26 de setembro de 2023

Novo perfil vinculado ao serviço e nova política do IAM	O CloudWatch adicionou um novo perfil vinculado ao serviço, chamado AWSServiceRoleForCloudWatchMetrics_DbPerfInsights. O CloudWatch adicionou esse novo perfil vinculado ao serviço para permitir que o CloudWatch substitua métricas do Insights de Performance para alarmes, detecção de anomalias e captura de instantâneos. A nova política do IAM AWSServiceRoleForCloudWatchMetrics_DBPerfInsightsServiceRolePolicy está anexada a esse perfil, e a política concede permissão ao CloudWatch para buscar métricas do Insights de Performance em seu nome.	20 de setembro de 2023
Foi adicionada uma nova função matemática de métrica	O CloudWatch adicionou uma nova função matemática de métrica, `DB_PERF_INSIGHTS` , que pode ser usada para buscar métricas do Insights de Performance de serviços de banco de dados da AWS para alarmes, detecção de anomalias e captura de instantâneos.	20 de setembro de 2023

Atualização da política CloudWatchReadOnlyAccess	O CloudWatch adicionou a permissão `application-autoscaling:DescribeScalingPolicies` a CloudWatchReadOnlyAccess para que os usuários com essa política possam acessar informações sobre as políticas do Application Auto Scaling.	14 de setembro de 2023
O agente do CloudWatch adicionou suporte para AL2023	O agente do CloudWatch oferece suporte a AL2023.	8 de agosto de 2023
Nova política do IAM gerenciada, CloudWatchFullAccessV2	O CloudWatch adicionou uma nova política: CloudWatchFullAccessV2. Essa política concede acesso total às ações e recursos do CloudWatch, ao mesmo tempo em que define melhor o escopo das permissões concedidas a outros serviços, como o Amazon SNS e o Amazon EC2 Auto Scaling.	1º de agosto de 2023

Perfil vinculado ao serviço atualizado para o Monitor de Internet do Amazon CloudWatch: atualização de uma política existente	Adiciona novas permissões, `elasticloadbalancing:DescribeLoadBalancers` e `ec2:DescribeNetworkInterfaces`, ao perfil vinculado ao serviço do Monitor de Internet, para oferecer suporte ao monitoramento de tráfego para recursos específicos do Network Load Balancer.	25 de julho de 2023
Adição de suporte a recursos do Network Load Balancer no Monitor de Internet do Amazon CloudWatch	Adiciona suporte para criar um monitor no Monitor de Internet com recursos específicos do Network Load Balancer, para fornecer níveis mais granulares de observabilidade para sua aplicação.	25 de julho de 2023

Recurso de variáveis de painel	O CloudWatch lançou variáveis de painel que podem ser usadas para criar painéis flexíveis que exibam rapidamente conteúdos diferentes, dependendo de como você definir um campo de entrada no painel. Por exemplo, é possível criar um painel que pode alternar rapidamente entre diferente s funções do Lambda ou IDs de instância do Amazon EC2, ou um que pode alternar para diferentes regiões da AWS. Para obter mais informações, consulte Crie painéis flexíveis com variáveis do painel.	28 de junho de 2023
O Monitor de Internet agora oferece suporte à personali zação do limite para eventos de integridade	O Monitor de Internet adicionou a capacidade de personalizar o limite para quando uma pontuação global de performance ou pontuação de disponibilidade acionar um evento de integridade. Para obter mais informaçõ es, consulte Monitoramento de performance e disponibi lidade em tempo real no Monitor de Internet do Amazon CloudWatch.	26 de junho de 2023

O Monitor de Internet agora oferece suporte a todas as regiões comerciais	O Monitor de Internet adicionou sete novas Regiões da AWS e agora oferece suporte a todas as regiões comerciais.	19 de junho de 2023
Novas versões da extensão Lambda Insights	O CloudWatch adicionou a versão 1.0.229.0 da extensão Lambda Insights para plataformas x86-64 e plataformas ARM64. Para obter mais informações, consulte Versões disponíveis da extensão do Lambda Insights.	12 de junho de 2023
Atualização da política CloudWatchReadOnlyAccess	O CloudWatch adicionou permissões ao CloudWatchReadOnlyAccess. As permissões `logs:StartLiveTail` e `logs:StopLiveTail` foram adicionadas para que os usuários com essa política possam usar o console para iniciar e interromper as sessões de teste ao vivo do CloudWatch Logs. Para obter mais informações, consulte Usar o Live Tail para visualizar registros quase em tempo real.	6 de junho de 2023

O CloudWatch RUM adicionou suporte para métricas personalizadas	É possível usar os monitores de aplicações do CloudWatch RUM para criar métricas personalizadas e enviá-las para o CloudWatch e o CloudWatch Evidently. Esse recurso inclui uma atualização da política do IAM gerenciada AmazonCloudWatchRUMServiceRolePolicy. Nessa política, uma chave de condição foi alterada para que o CloudWatch RUM possa enviar métricas personalizadas para namespaces de métricas personalizadas.	9 de fevereiro de 2023
Políticas gerenciadas novas e atualizadas para o CloudWatch	Para dar suporte à observabilidade entre contas do CloudWatch, as políticas `CloudWatchFullAccess` e `CloudWatchReadOnlyAccess` foram atualizadas e as seguintes novas políticas gerenciadas foram adicionadas: `CloudWatchCrossAccountSharingConfiguration` , `OAMFullAccess` e `OAMReadOnlyAccess` . Para mais informações, consulte as CloudWatch atualiza para AWS políticas gerenciadas.	7 de fevereiro de 2023

Atualizações da política de perfil vinculado a serviço do CloudWatch Application Insights: atualização de uma política existente.	O CloudWatch Application Insights atualizou uma política de função vinculada a serviço da AWS existente.	19 de dezembro de 2022
Suporte do Amazon CloudWatch Application Insights para aplicações e microsserviços em contêiner a partir do console do Container Insights.	É possível exibir problemas detectados pelo CloudWatch Application Insights para o Amazon ECS e o Amazon EKS no painel do Container Insights.	17 de novembro de 2021
Monitoramento do Amazon CloudWatch Application Insights para bancos de dados SAP HANA.	Você pode monitorar bancos de dados SAP HANA com o Application Insights.	15 de novembro de 2021
Suporte do Amazon CloudWatch Application Insights para o monitoramento de todos os recursos em uma conta.	Você pode integrar e monitorar todos os recursos em uma conta.	15 de setembro de 2021
Compatibilidade do Amazon CloudWatch Application Insights com o Amazon FSx.	É possível monitorar métricas recuperadas do Amazon FSx.	31 de agosto de 2021
O SDK Metrics não é mais compatível.	O CloudWatch SDK Metrics não é mais compatível.	25 de agosto de 2021
O Amazon CloudWatch Application Insights é compatível com configurações do monitoramento de contêineres.	É possível monitorar contêineres usando as práticas recomendadas com o Amazon CloudWatch Application Insights.	18 de maio de 2021

Os fluxos de métricas estão disponíveis ao público	Você pode usar fluxos de métricas para transmitir continuamente as métricas do CloudWatch para um destino de sua preferência. Para obter mais informações, consulte Fluxos de métricas no Manual do usuário do Amazon CloudWatch.	31 de março de 2021
Monitoramento do Amazon CloudWatch Application Insights para bancos de dados Oracle no Amazon RDS e no Amazon EC2.	É possível monitorar métricas e logs recuperados do Oracle com o Amazon CloudWatch Application Insights.	16 de janeiro de 2021
O Lambda Insights está disponível ao público	O Lambda Insights do CloudWatch Lambda é uma solução de monitoramento e solução de problemas para aplicações sem servidor em execução no AWS Lambda. Para obter mais informaçõ es, consulte Usar o Lambda Insights no Manual do usuário do Amazon CloudWatch.	3 de dezembro de 2020
Monitoramento do Amazon CloudWatch Application Insights para métricas do Prometheus JMX Exporter.	É possível monitorar métricas recuperadas do Prometheus JMX Exporter com o Amazon CloudWatch Application Insights.	20 de novembro de 2020

CloudWatch Synthetics lança nova versão de runtime	O CloudWatch Synthetics lançou nova versão de runtime. Para obter mais informações, consulte Versões do runtime do canário no Manual do usuário do Amazon CloudWatch.	11 de setembro de 2020
Monitoramento do Amazon CloudWatch Application Insights para Postgre SQL no Amazon RDS e no Amazon EC2.	É possível monitorar aplicações criadas com o PostgreSQL em execução no Amazon RDS ou no Amazon EC2.	11 de setembro de 2020
CloudWatch compatível com compartilhamento de painel	Agora é possível compartilhar painéis do CloudWatch com pessoas de fora da sua organização e conta da AWS. Para obter mais informações, consulte Compartilhar painéis do CloudWatch no Manual do usuário do Amazon CloudWatch.	10 de setembro de 2020
Configurar monitores para aplicações .NET usando o SQL Server no backend com o CloudWatch Application Insights	Você pode usar o tutorial da documentação para ajudar a configurar monitores para aplicações .NET usando o SQL Server no backend com o CloudWatch Application Insights.	19 de agosto de 2020

Compatibilidade do AWS CloudFormation com aplicações do Amazon CloudWatch Application Insights.	É possível adicionar o monitoramento do CloudWatch Application Insights, incluindo as principais métricas e telemetria, à aplicação, ao banco de dados e ao servidor Web, diretamente a partir de modelos do AWS CloudFormation.	30 de julho de 2020
Monitoramento do Amazon CloudWatch Application Insights para clusters de banco de dados do Aurora for MySQL.	É possível monitorar clusters de banco de dados do Aurora for MySQL (RDS Aurora) com o Amazon CloudWatch Application Insights.	2 de julho de 2020
Disponibilidade geral do CloudWatch Contributor Insights	O CloudWatch Contributor Insights já está disponível para o público. Ele permite que você analise os dados de log e crie séries temporais que exibem dados de colaboradores. É possível ver métricas sobre os principais colaboradores, o número total de colaboradores exclusivos e o uso deles. Para obter mais informações, consulte Usar o Contributor Insights para analisar dados de alta cardinalidade no Guia do usuário do Amazon CloudWatch.	2 de abril de 2020

Pré-visualização pública do CloudWatch Synthetics	O CloudWatch Synthetics agora está em pré-visualização pública. Ele permite que você crie canaries para monitorar seus endpoints e APIs. Para obter mais informações, consulte Usar canaries no Manual do usuário do Amazon CloudWatch.	25 de novembro de 2019
Pré-visualização pública do CloudWatch Contributor Insights	O CloudWatch Contributor Insights agora está em pré-visualização pública. Ele permite que você analise os dados de log e crie séries temporais que exibem dados de colaboradores. É possível ver métricas sobre os principais colaboradores, o número total de colaboradores exclusivos e o uso deles. Para obter mais informações, consulte Usar o Contributor Insights para analisar dados de alta cardinalidade no Guia do usuário do Amazon CloudWatch.	25 de novembro de 2019

CloudWatch lança o recurso ServiceLens	O ServiceLens melhora a capacidade de observação de seus serviços e aplicativos permitindo a integração de rastreamentos, métricas, logs e alarmes em um só lugar. O ServiceLens integra o CloudWatch ao AWS X-Ray para oferecer uma visão completa de sua aplicação.	21 de novembro de 2019
Usar o CloudWatch para gerenciar proativamente suas cotas de serviço da AWS	Você pode usar o CloudWatch para gerenciar proativamente suas cotas de serviço da AWS. As métricas de uso do CloudWatch fornecem visibilidade sobre o uso de recursos e operações de API de sua conta. Para obter mais informações, consulte Integração do Service Quotas e métricas de uso no Manual do usuário do Amazon CloudWatch.	19 de novembro de 2019
CloudWatch envia eventos quando os alarmes mudam de estado	O CloudWatch agora envia um evento ao Amazon EventBridge quando algum estado de alarme do CloudWatch é alterado. Para obter mais informações, consulte Eventos de alarme e EventBridge no Manual do usuário do Amazon CloudWatch.	8 de outubro de 2019

Container Insights	O CloudWatch Container Insights já está disponível para o público. Ele permite coletar, agregar e resumir métricas e logs de seus aplicativos e microsserviços em contêineres. Para obter mais informações, consulte Usar o Container Insights no Manual do usuário do Amazon CloudWatch.	30 de agosto de 2019
Atualizações das métricas de pré-visualização do Container Insights no Amazon EKS e no Kubernetes	A pré-visualização pública do Container Insights no Amazon EKS e no Kubernetes foi atualizada. InstanceId agora está incluído como uma dimensão nas instâncias do EC2 do cluster. Isso permite que os alarmes que foram criados nessas métricas acionem as seguintes ações do EC2: parar, encerrar, reinicializar ou recuperar. Além disso, métricas de pod e de serviço agora são relatadas pelo namespace do Kubernetes para simplificar o monitoramento e o alarme de métricas por namespace.	19 de agosto de 2019
Atualizações para integração com o OpsCenter do AWS Systems Manager	Atualizações sobre como o CloudWatch Application Insights se integra ao Systems Manager OpsCenter.	7 de agosto de 2019

Métricas de uso do CloudWatch	As métricas de uso do CloudWatch ajudam você a controlar o uso de seus recursos do CloudWatch e permanecer dentro dos limites do serviço. Para ter mais informações, consulte https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Usage-Metrics.html.	6 de agosto de 2019
Pré-visualização pública do CloudWatch Container Insights	O CloudWatch Container Insights agora está em pré-visualização pública. Ele permite coletar, agregar e resumir métricas e logs de seus aplicativos e microsserviços em contêineres. Para obter mais informações, consulte Usar o Container Insights no Manual do usuário do Amazon CloudWatch.	9 de julho de 2019

Pré-visualização pública da detecção de anomalias do CloudWatch	A detecção de anomalias do CloudWatch agora está em pré-visualização pública. O CloudWatch aplica algoritmos de machine learning aos dados passados de uma métrica para criar um modelo dos valores esperados da métrica. Você pode usar esse modelo para visualização e configuração de alarmes. Para obter mais informações, consulte Usar a deteção de anomalias do CloudWatch no Manual do usuário do Amazon CloudWatch.	9 de julho de 2019
CloudWatch Application Insights para .NET e SQL Server	O CloudWatch Application Insights para .NET e SQL Server facilita a capacidade de observação de aplicações.NET e SQL Server. Ele pode ajudar você a configurar os melhores monitores para os recursos do aplicativo, a analisar dados continuamente para procurar sinais de problemas com seus aplicativos.	21 de junho de 2019

A seção do atendente do CloudWatch foi reorganizada	A documentação do atendente do CloudWatch foi reescrita para aumentar a clareza, sobretudo para os clientes que usam a linha de comando para instalar e configurar o atendente. Para obter mais informações, consulte Coletar métricas e logs de instâncias do Amazon EC2 e de servidores on-premises com o atendente do CloudWatch no Manual do usuário do Amazon CloudWatch.	28 de março de 2019
Função SEARCH adicionada às expressões matemáticas de métricas	Agora, você pode usar uma função SEARCH nas expressões matemáticas de métricas. Isso permite que você crie painéis que são atualizados automaticamente à medida que são criados novos recursos que correspondem à consulta de pesquisa. Para obter mais informações, consulte Usar expressões de pesquisa em gráficos no Manual do usuário do Amazon CloudWatch.	21 de março de 2019

AWS SDK Metrics para Enterprise Support	O SDK Metrics ajuda a avaliar a integridade de seus produtos da AWS e a diagnosticar a latência causada por atingir seus limites de uso da conta ou por uma interrupção de serviço. Para obter mais informações, consulte Monitorar as aplicações da AWS usando SDK Metrics no Manual do usuário do Amazon CloudWatch.	11 de dezembro de 2018
Alarmes em expressões matemáticas	O CloudWatch oferece suporte à criação de alarmes com base em expressões matemáticas de métrica. Para obter mais informações, consulte Alarmes de expressões matemáticas no Manual do usuário do Amazon CloudWatch.	20 de novembro de 2018
Nova página inicial do console do CloudWatch	A Amazon criou uma nova página inicial no console do CloudWatch, que exibe automaticamente as métricas e os alarmes principais para todos os produtos da AWS que você está usando. Para obter mais informações, consulte Conceitos básicos do Amazon CloudWatch no Manual do usuário do Amazon CloudWatch.	19 de novembro de 2018

modelos do AWS CloudFormation para o atendente do CloudWatch	A Amazon carregou modelos do AWS CloudFormation que é possível usar para instalar e atualizar o atendente do CloudWatch. Para obter mais informações, consulte Instalar o atendente do CloudWatch em novas instâncias usando o AWS CloudFormation no Manual do usuário do Amazon CloudWatch.	9 de novembro de 2018
Melhorias do atendente do CloudWatch	O atendente do CloudWatch foi atualizado para funcionar com os protocolos StatsD e collectd. Ele também melhorou o suporte entre contas. Para obter mais informações, consulte Recuperar métricas personalizadas com o StatsD, Recuperar métricas personalizadas com o collectd e Enviar métricas e logs para uma conta da AWS diferente no Manual do usuário do Amazon CloudWatch.	28 de setembro de 2018
Suporte para endpoints da Amazon VPC	Agora você pode estabelecer uma conexão privada entre seu VPC e CloudWatch. Para obter mais informações, consulte Usar o CloudWatch com endpoints da VPC de interface no Manual do usuário do Amazon CloudWatch.	28 de junho de 2018

A tabela a seguir descreve alterações importantes no Manual do usuário do Amazon CloudWatch antes de junho de 2018.

Alteração	Descrição	Data de lançamento
Matemática de métricas	Agora, você pode executar expressões matemáticas em métricas do CloudWatch, produzindo novas séries temporais que você pode adicionar a gráficos em seu painel. Para ter mais informações, consulte Usar matemática de métricas.	4 de abril de 2018
Alarmes "M out of N"	Agora você pode configurar um alarme para ser acionado com base em pontos de dados "M out of N" em qualquer intervalo de avaliação de alarme. Para ter mais informações, consulte Avaliar um alarme.	8 de dezembro de 2017
Agente do CloudWatch	Foi lançado um novo atendente unificado do CloudWatch. Você pode usar o atendente unificado de várias plataformas para coletar métricas do sistema e arquivos de log personalizados das instâncias do Amazon EC2 e de servidores on-premises. O novo atendente oferece suporte ao Windows e ao Linux e permite a personalização das métricas a serem coletadas, incluindo métricas de sub-recurso, como núcleo por CPU. Para ter mais informações, consulte Coletar métricas, logs e rastreamentos com o agente do CloudWatch.	7 de setembro de 2017
Métricas do gateway NAT	Foram adicionadas métricas para o gateway NAT da Amazon VPC.	7 de setembro de 2017
Métricas de alta resolução	Agora, se quiser, você pode configurar métricas personalizadas como métricas de alta resolução, com uma granularidade tão baixa quanto um segundo. Para ter mais informações, consulte Métricas de alta resolução.	26 de julho de 2017

Alteração	Descrição	Data de lançamento
APIs de painel	Agora você pode criar, modificar e excluir painéis usando APIs e a AWS CLI. Para ter mais informações, consulte Criar um painel do CloudWatch.	6 de julho de 2017
AWS Direct Connect métricas	Inclusão de métricas para o AWS Direct Connect.	29 de junho de 2017
Métricas de VPN da Amazon VPC	Adição de métricas para a VPN da Amazon VPC.	15 de maio de 2017
Métricas do AppStream 2.0	Foram adicionadas métricas para o AppStream 2.0.	8 de março de 2017
Seletor de cor do console do CloudWatch	Agora você pode escolher a cor para cada métrica nos widgets do seu painel. Para ter mais informaçõ es, consulte Editar um gráfico em um painel do CloudWatch .	27 de fevereiro de 2017
Alarmes em painéis	Agora é possível adicionar alarmes aos painéis. Para ter mais informações, consulte Adicionar ou remover um widget de alarme em um painel do CloudWatch .	15 de fevereiro de 2017
Adição de métricas para o Amazon Polly	Foram adicionadas métricas para o Amazon Polly.	1° de dezembro de 2016
Foram adicionad as métricas para o Amazon Managed Service for Apache Flink	Foram adicionadas métricas para o Amazon Managed Service for Apache Flink.	1° de dezembro de 2016

Alteração	Descrição	Data de lançamento
Inclusão de suporte para estatísticas de percentil	É possível especificar qualquer percentil usando até duas casas decimais (por exemplo, p95.45). Para ter mais informações, consulte Percentis.	17 de novembro de 2016
Adição de métricas para o Amazon Simple Email Service	Foram adicionadas métricas para o Amazon Simple Email Service.	2 de novembro de 2016
Atualizada a retenção de métricas	O Amazon CloudWatch agora retém os dados de métricas por 15 meses, em vez de 14 dias.	1 de novembro de 2016
Atualizada a interface do console de métricas	O console do CloudWatch foi atualizado com melhorias nas funcionalidades existentes e novas funcionalidades.	1 de novembro de 2016
Adição de métricas para o Amazon Elastic Transcoder	Foram adicionadas métricas para o Amazon Elastic Transcoder.	20 de setembro de 2016
Adição de métricas para Amazon API Gateway	Foram adicionadas métricas para Amazon API Gateway.	9 de setembro de 2016
Inclusão de métricas para o AWS Key Management Service	Inclusão de métricas para o AWS Key Managemen t Service.	9 de setembro de 2016

Alteração	Descrição	Data de lançamento
Adição de métricas para os novos balanceadores de carga da aplicação compatíveis com o Elastic Load Balancing	Foram adicionadas métricas dos balanceadores de carga da aplicação.	11 de agosto de 2016
Inclusão de novas métricas NetworkPacketsIn e NetworkPacketsOut para o Amazon EC2	Inclusão de novas métricas NetworkPacketsIn e NetworkPacketsOut para o Amazon EC2.	23 de março de 2016
Adição de novas métricas para a frota Spot do Amazon EC2	Foram adicionadas novas métricas para a frota Spot do Amazon EC2.	21 de março de 2016
Adição de novas métricas do CloudWatch Logs	Foram adicionadas novas métricas do CloudWatch Logs.	10 de março de 2016
Adição de métricas e dimensões do Amazon OpenSearch Service e do AWS WAF	Foram adicionadas métricas e dimensões do Amazon OpenSearch Service e do AWS WAF.	14 de outubro de 2015

Alteração	Descrição	Data de lançamento
Adição de suporte para painéis do CloudWatch	Os painéis são páginas de início personalizáveis no console do CloudWatch que você pode usar para monitorar seus recursos em uma única visualização, mesmo aqueles distribuídos em regiões diferentes. Para ter mais informações, consulte Usar painéis do Amazon CloudWatch.	8 de outubro de 2015
Inclusão de métricas e dimensões do AWS Lambda	Inclusão de métricas e dimensões do AWS Lambda.	4 de setembro de 2015
Adição de métricas e dimensões do Amazon Elastic Container Service	Foram adicionadas métricas e dimensões do Amazon Elastic Container Service.	17 de agosto de 2015
Adição de métricas e dimensões do Amazon Simple Storage	Foram adicionadas métricas e dimensões do Amazon Simple Storage.	26 de julho de 2015
Novo recurso: Reinicializar ação de alarme	Adicionada a ação de alarme de reinicialização e a nova função do IAM para uso com ações de alarme. Para ter mais informações, consulte Criar alarmes para interromper, terminar, reinicializar ou recuperar uma instância do EC2.	23 de julho de 2015
Adição de métricas e dimensões do Amazon WorkSpaces	Foram adicionadas métricas e dimensões do Amazon WorkSpaces.	30 de abril de 2015

Alteração	Descrição	Data de lançamento
Adição de métricas e dimensões do Amazon Machine Learning	Foram adicionadas métricas e dimensões do Amazon Machine Learning.	9 de abril de 2015
Novo recurso: ações de alarme de recuperação de instância do Amazon EC2	Atualizadas ações de alarme para incluir novas ações de recuperação de instância do EC2. Para ter mais informações, consulte Criar alarmes para interromper, terminar, reinicializar ou recuperar uma instância do EC2.	12 de março de 2015
Adição de métricas e dimensões do Amazon CloudFront e do Amazon CloudSearch	Foram adicionadas métricas e dimensões do Amazon CloudFront e do Amazon CloudSearch.	6 de março de 2015
Adição de métricas e dimensões do Amazon Simple Workflow Service	Foram adicionadas métricas e dimensões do Amazon Simple Workflow Service.	9 de maio de 2014
Guia atualizado para adicionar suporte ao AWS CloudTrail	Adição de um novo tópico para explicar como você pode usar o AWS CloudTrail para registrar uma atividade no Amazon CloudWatch. Para ter mais informações, consulte Registrar chamadas de API do Amazon CloudWatch com o AWS CloudTrail.	30 de abril de 2014

Alteração	Descrição	Data de lançamento
Atualizado o guia para usar a nova AWS Command Line Interface (AWS CLI)	A AWS CLI é uma CLI entre serviços com uma instalação simplificada, configuração unificada e sintaxe de linha de comando consistente. A AWS CLI é compatível com Linux/Unix, Windows e Mac. Os exemplos da CLI deste guia foram atualizados para usar a nova AWS CLI. Para obter informações sobre como instalar e configurar a nova AWS CLI, consulte Configurar a AWS CLI no Manual do usuário da AWS Command Line Interface.	21 de fevereiro de 2014
Adição de métricas e dimensões do Amazon Redshift e do AWS OpsWorks	Foram adicionadas métricas e dimensões do Amazon Redshift e do AWS OpsWorks.	16 de julho de 2013
Adição de métricas e dimensões do Amazon Route 53	Foram adicionadas métricas e dimensões do Amazon Route 53.	26 de junho de 2013
Novo recurso: ações de alarme do Amazon CloudWatch	Inclusão de uma nova seção para documentar ações de alarme do Amazon CloudWatch, que podem ser usadas para interromper ou terminar uma instância do Amazon Elastic Compute Cloud. Para ter mais informações, consulte Criar alarmes para interromper, terminar, reinicializar ou recuperar uma instância do EC2.	8 de janeiro de 2013
Atualizadas as métricas do EBS	Atualizadas as métricas do EBS para incluir duas novas métricas para volumes de Provisioned IOPS.	20 de novembro de 2012

Alteração	Descrição	Data de lançamento
Novos alertas de pagamento	Agora, você pode monitorar suas despesas da AWS usando métricas do Amazon CloudWatch e criar alarmes para notificar você quando o limite especificado for excedido. Para ter mais informaçõ es, consulte Criar um alarme de faturamento para monitorar suas cobranças estimadas da AWS.	10 de maio de 2012
Novas métricas	Agora, você pode acessar seis novas métricas do Elastic Load Balancing que fornecem contagens de vários códigos de resposta HTTP.	19 de outubro de 2011
Novo atributo	Agora você pode acessar métricas do Amazon EMR.	30 de junho de 2011
Novo atributo	Agora é possível acessar métricas do Amazon Simple Notification Service e do Amazon Simple Queue Service.	14 de julho de 2011
Novo recurso	Inclusão de informações sobre como usar a API `PutMetricData` para publicar métricas personalizadas. Para ter mais informações, consulte Publicar métricas personalizadas do .	10 de maio de 2011
Atualizada a retenção de métricas	O Amazon CloudWatch agora retém o histórico de um alarme por duas semanas, em vez de seis semanas. Com essa alteração, o período de retenção para os alarmes corresponde ao período de retenção dos dados de métricas.	7 de abril de 2011
Novo atributo	Inclusão de capacidade de enviar notificações do Amazon Simple Notification Service ou do Auto Scaling quando uma métrica ultrapassa um limite. Para ter mais informações, consulte Alarmes.	2 de dezembro de 2010

Alteração	Descrição	Data de lançamento
Novo atributo	Uma série de ações do CloudWatch agora incluem os parâmetros MaxRecords e NextToken, o que permite controlar as páginas de resultados a serem exibidas.	2 de dezembro de 2010
Novo atributo	Esse serviço agora é integrado ao AWS Identity and Access Management (IAM).	2 de dezembro de 2010