aws

Guia do usuário

Amazon EventBridge

Amazon EventBridge: Guia do usuário

Table of Contents

O que é o Amazon EventBridge?

O EventBridge é um serviço com tecnologia sem servidor que usa eventos para conectar os componentes da aplicação, facilitando a criação de aplicações escaláveis orientadas por eventos. A arquitetura orientada por eventos é um estilo de criação de sistemas de software com acoplamento fraco que funcionam juntos emitindo e respondendo a eventos. A arquitetura orientada por eventos pode ajudar você a aumentar a agilidade e criar aplicações confiáveis e escaláveis.

Use o EventBridge para rotear eventos de origens como aplicações, serviços da AWS e software de terceiros desenvolvidos internamente para aplicações de consumo em toda a sua organização. O EventBridge fornece maneiras simples e consistentes de ingerir, filtrar, transformar e entregar eventos para que você possa criar aplicações rapidamente.

O seguinte vídeo fornece uma breve introdução aos atributos do Amazon EventBridge:

O EventBridge inclui duas formas de processar eventos: barramentos de eventos e pipes.

- Os barramentos de eventos são roteadores que recebem eventos e os entregam a zero ou mais destinos. Os barramentos de eventos são adequados para rotear eventos de várias origens para vários destinos, com a transformação opcional dos eventos antes da entrega a um destino.

  O seguinte vídeo fornece uma visão geral de alto nível dos barramentos de eventos:

- Pipes: o EventBridge Pipes é destinado a integrações ponto a ponto; cada pipe recebe eventos de uma única origem para processamento e entrega a um único destino. Os pipes também incluem suporte para transformações avançadas e enriquecimento de eventos antes da entrega a um destino.

Pipes e barramentos de eventos costumam ser usados juntos. Um caso de uso comum é criar um pipe com um barramento de eventos como destino; o pipe envia eventos para o barramento de eventos, que então envia esses eventos para vários destinos. Por exemplo, é possível criar um pipe com um fluxo do DynamoDB para uma origem e um barramento de eventos como destino. O pipe recebe eventos do fluxo do DynamoDB e os envia para o barramento de eventos, que os envia para vários destinos de acordo com as regras que você especificou no barramento de eventos.

O EventBridge é a evolução do Amazon CloudWatch Events.

Anteriormente, o EventBridge era chamado de Amazon CloudWatch Events. O barramento de eventos padrão e as regras criadas no CloudWatch Events também são exibidos no console do EventBridge. O EventBridge usa a mesma API do CloudWatch Events para que seu código que usa a API do CloudWatch Events permaneça o mesmo.

O EventBridge se baseia nos recursos do CloudWatch Events com atributos como eventos de parceiros, Schema Registry e EventBridge Pipes. Novos atributos adicionados ao EventBridge não são adicionados ao CloudWatch Events. Para obter mais informações, consulte ???.

Todos os atributos de costume no CloudWatch Events também estão presentes no EventBridge, incluindo:

- ???
- ???
- ???
- ???

Os atributos do EventBridge que se baseiam e expandem as capacidades dos eventos incluem:

- ???
- ???
- ???
- ???

EventBridge Configuração e pré-requisitos da Amazon

Para usar a Amazon EventBridge, você precisa de uma AWS conta. Sua conta permite que você use serviços como o Amazon EC2 para gerar eventos que você pode ver no EventBridge console. Você também pode instalar e configurar o AWS Command Line Interface (AWS CLI) para usar uma interface de linha de comando para ver eventos.

Tópicos

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um 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 inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. 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 inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, digite sua senha.

   Para obter ajuda ao fazer login usando o usuário raiz, consulte Signing in as the root user (Fazer login como usuário raiz) no Guia do usuário Início de Sessão da AWS .
2. Ative a autenticação multifator (MFA) para seu usuário raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (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 ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com seu usuário IAM Identity Center, use a URL de login enviada ao seu endereço de e-mail quando você criou o usuário IAM Identity Center user.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a 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 EventBridge console da Amazon

Para fazer login no EventBridge console da Amazon

- Faça login AWS Management Console e abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.

Credenciais da conta

Embora você possa usar suas credenciais de usuário raiz para acessar EventBridge, recomendamos que você use uma conta AWS Identity and Access Management (IAM) em vez disso. Se você estiver usando uma conta do IAM para acessar EventBridge, você deve ter as seguintes permissões.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "events:*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:events:*:*:*"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
```

```
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "events.amazonaws.com"
        }
      }
    }
  ]
}
```

Para ter mais informações, consulte Autenticação.

Configure o AWS Command Line Interface

Você pode usar o AWS CLI para realizar EventBridge operações.

Para obter informações sobre como instalar e configurar o AWS CLI, consulte Como configurar o AWS Command Line Interface no Guia do AWS Command Line Interface usuário.

Endpoints regionais

Você deve habilitar o uso EventBridge dos endpoints regionais padrão. Para obter mais informações, consulte Ativação e desativação AWS STS em uma AWS região no Guia do usuário do IAM.

Começando com a Amazon EventBridge

A base do EventBridge é criar regras que direcionem eventos para um alvo. Nesta seção, uma regra básica é criada. Para tutoriais sobre cenários e destinos específicos, consulte Tutoriais do Amazon EventBridge.

Crie uma regra na Amazon EventBridge

Para criar uma regra para eventos, você especifica uma ação a ser tomada ao EventBridge receber um evento que corresponda ao padrão do evento na regra. Quando um evento coincide, EventBridge envia o evento para o destino especificado e aciona a ação definida na regra.

Quando um AWS serviço em sua AWS conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta. Para escrever uma regra que corresponda AWS aos eventos dos serviços em sua conta, você deve associá-la ao barramento de eventos padrão.

Para criar uma regra para um AWS serviço

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra na mesma Região e barramento de eventos.
5. Em Barramento de eventos, selecione o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, escolha Barramento de eventos padrão da AWS . Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Tipo de Regra, escolha Regra com Padrão de Evento.
7. Selecione Next (Próximo).
8. Em Fonte do evento, selecione Serviços da AWS .
9. (Opcional) Em Eventos de amostra, escolha o tipo de evento.
10. Em Padrão de evento, siga um destes procedimentos:
    - Para usar um modelo para criar o padrão de evento, escolha Formulário de padrão de evento e escolha as opções Origem do evento e Tipo de evento. Se você escolher Todos

os eventos como o tipo de evento, todos os eventos emitidos por esse AWS serviço corresponderão à regra.

Para personalizar o modelo, escolha Padrão personalizado (editor JSON) e faça as alterações.

- Para utilizar um padrão de evento personalizado, escolha Padrão personalizado (editor JSON) e crie o padrão do evento.

11. Selecione Next (Próximo).
12. Em Tipos de destino, escolha Serviço da AWS .
13. Em Selecionar um destino, escolha o AWS serviço para o qual você deseja enviar informações ao EventBridge detectar um evento que corresponda ao padrão do evento.
14. Os campos exibidos variam de acordo com o serviço escolhido. Insira as informações específicas desse tipo de destino conforme necessário.
15. Para muitos tipos de alvo, EventBridge precisa de permissões para enviar eventos ao alvo. Nesses casos, EventBridge pode criar a função do IAM necessária para que sua regra seja executada. Execute um destes procedimentos:

    - Para criar um perfil do IAM automaticamente, escolha Criar um novo perfil para este recurso específico.
    - Para usar um perfil do IAM que você criou anteriormente, escolha Usar perfil existente e selecione o perfil existente na lista suspensa.

16. (Opcional) Para Configurações Adicionais, proceda da seguinte forma:

    a. Em Tempo Máximo do Evento, insira um valor entre um minuto (00:01) e 24 horas (24:00).
    b. Em Tentativas de Repetição, insira um número entre 0 e 185.
    c. Para fila de mensagens mortas, escolha se deseja usar uma fila padrão do Amazon SQS como fila de mensagens mortas. EventBridge envia eventos que correspondam a essa regra para a fila de mensagens mortas se não forem entregues com sucesso ao destino. Faça um dos procedimentos a seguir:

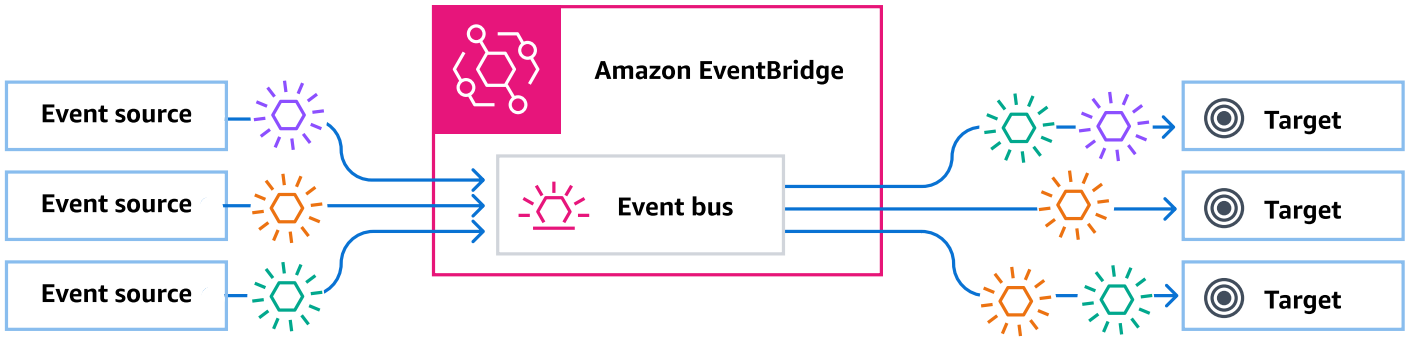
        - Escolha None (Nenhum) para não usar uma fila de mensagens não entregues.
        - Escolha Selecionar uma fila do Amazon SQS na conta atual da AWS para usar como a fila de mensagens não entregues e depois selecione na lista suspensa a fila a ser usada.
        - Escolha Selecionar uma fila do Amazon SQS em outra AWS conta como uma fila de mensagens mortas e, em seguida, insira o ARN da fila a ser usada. Você deve

anexar uma política baseada em recursos à fila que conceda EventBridge permissão para enviar mensagens para ela. Para ter mais informações, consulte Como conceder permissões para a fila de mensagens não entregues.

17. (Opcional) Selecione Adicionar outro destino para adicionar outro destino a essa regra.
18. Escolha Next (Próximo).
19. (Opcional) Insira uma ou mais tags para a regra. Para ter mais informações, consulte EventBridge Etiquetas da Amazon.
20. Escolha Próximo.
21. Analise os detalhes da regra e selecione Criar regra.

Ônibus de EventBridge eventos da Amazon

Um barramento de eventos é um roteador que recebe eventos e os entrega a zero ou mais destinos ou destinos. Os barramentos de eventos são adequados para rotear eventos de várias origens para vários destinos, com a transformação opcional dos eventos antes da entrega a um destino.



As regras associadas ao barramento de eventos avaliam os eventos à medida que eles chegam. Cada regra verifica se um evento corresponde ao padrão da regra. Se o evento coincidir, EventBridge envia o evento

Uma regra é associada a um barramento de eventos específico. Assim, a regra se aplica somente aos eventos recebidos por esse barramento de eventos.

> **Note**
>
> Você também pode processar eventos usando EventBridge Pipes. EventBridge O Pipes é destinado a point-to-point integrações; cada tubo recebe eventos de uma única fonte para processamento e entrega a um único destino. Os pipes também incluem suporte para transformações avançadas e enriquecimento de eventos antes da entrega a um destino. Para ter mais informações, consulte ???.

Tópicos

- Excluindo um ônibus de EventBridge eventos da Amazon
- Permissões de barramentos de evento do Amazon EventBridge Pipes
- Gere um modelo do AWS CloudFormation a partir de um barramento de eventos do Amazon EventBridge

Como funcionam os barramentos de eventos

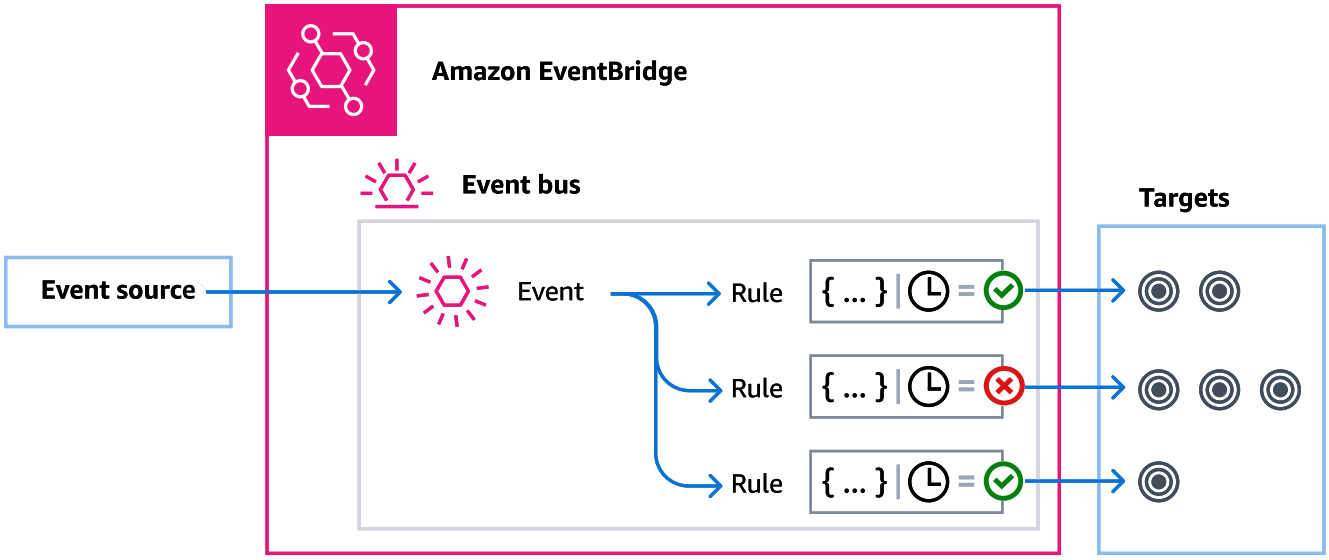
Os barramentos de eventos permitem o roteamento de eventos de várias origens para vários destinos.

Em um alto nível, veja como isso funciona:

1. Uma fonte de eventos, que pode ser um AWS serviço, seu próprio aplicativo personalizado ou um provedor de SaaS, envia um evento para um barramento de eventos.
2. EventBridge em seguida, avalia o evento em relação a cada regra definida para esse barramento de eventos.

   Para cada evento que corresponda a uma regra EventBridge , envie o evento para os destinos especificados para essa regra. Opcionalmente, como parte da regra, você também pode EventBridge especificar como transformar o evento antes de enviá-lo ao (s) destino (s).

   Um evento pode corresponder a várias regras, e cada regra pode especificar até cinco destinos. (Um evento pode não corresponder a nenhuma regra e, nesse caso, não EventBridge requer nenhuma ação.)



Considere um exemplo usando o barramento de eventos EventBridge padrão, que recebe automaticamente eventos dos AWS serviços:

1. É possível uma regra no barramento de eventos padrão para o evento `EC2 Instance State-change Notification`:
   - Especifique que a regra corresponda aos eventos em que uma instância do Amazon EC2 alterou seu estado `state` para `running`.

     Isto é feito ao especificar o JSON que define os atributos e valores que um evento deve corresponder para acionar a regra. Isto é chamado de padrão de evento.

     ```
     {
       "source": ["aws.ec2"],
       "detail-type": ["EC2 Instance State-change Notification"],
         "detail": {
           "state": ["running"]
       }
     }
     ```

   - Especifique o destino da regra para ser uma determinada função do Lambda.
2. Sempre que uma instância do Amazon EC2 muda de estado, o Amazon EC2 (a origem do evento) envia automaticamente esse evento para o barramento de eventos padrão.
3. EventBridge avalia todos os eventos enviados para o barramento de eventos padrão em relação à regra que você criou.

Se o evento corresponder à sua regra (ou seja, se o evento foi uma instância do Amazon EC2 mudando de estado para`running`), EventBridge envia o evento para o destino especificado. Neste caso, esta é a função do Lambda.

O seguinte vídeo descreve o que são barramentos de eventos e o que eles fazem: O que são barramentos de eventos

O seguinte vídeo aborda os diferentes barramentos de eventos e quando usá-los: As diferenças entre barramentos de eventos

Conceitos EventBridge do Amazon Event Bus

Aqui está uma análise mais detalhada dos principais componentes de uma arquitetura orientada a eventos construída em barramentos de eventos.

Barramentos de eventos

Um barramento de eventos é um roteador que recebe eventos e os entrega a zero ou mais destinos. Use um barramento de eventos quando precisar rotear eventos de várias origens para vários destinos, com a transformação opcional dos eventos antes da entrega em um destino.

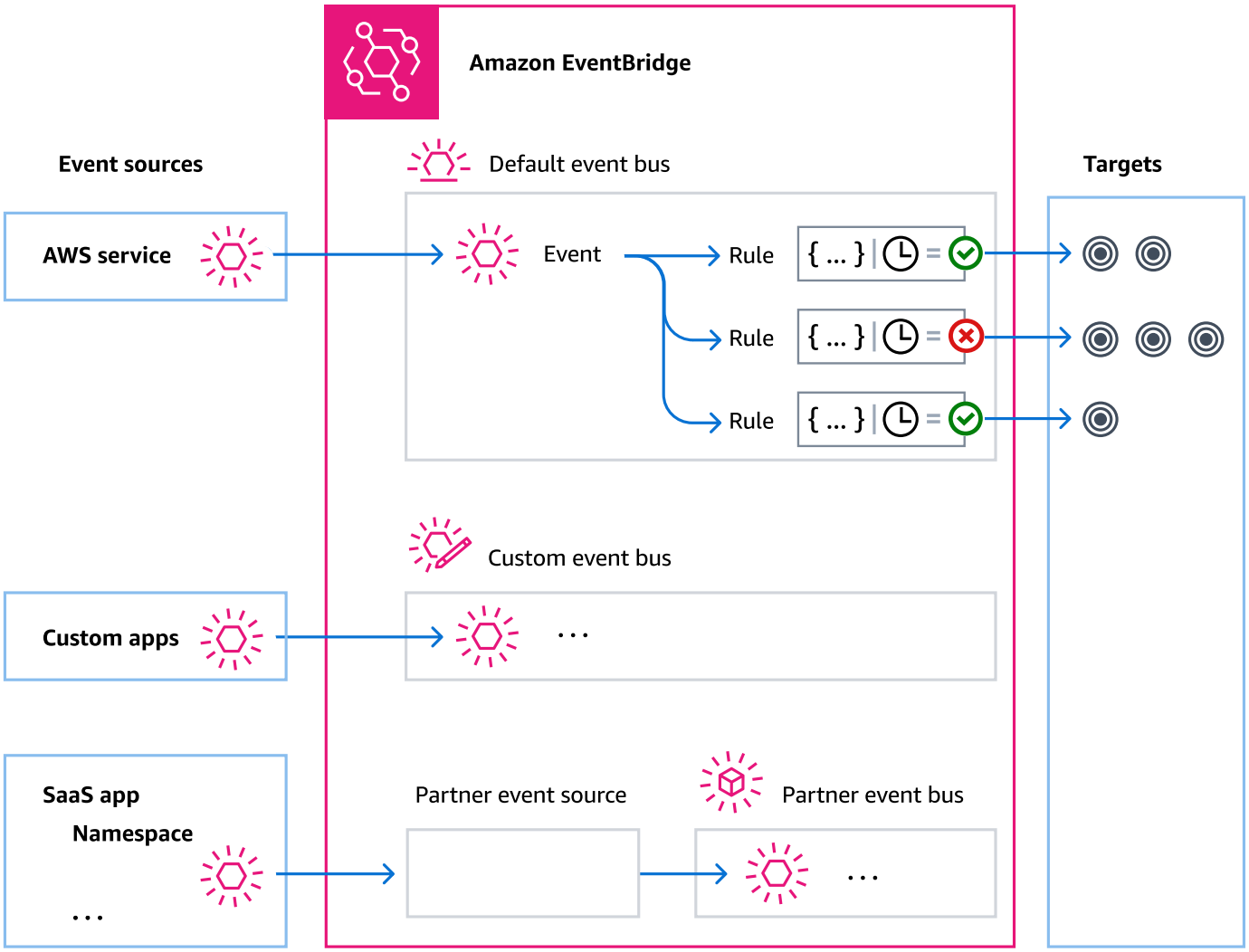
Sua conta inclui um barramento de eventos padrão que recebe automaticamente eventos dos AWS serviços. Você também pode:

- Crie barramentos de eventos adicionais, chamados de barramentos de eventos personalizados, e especifique quais eventos eles receberão.
- Crie barramentos de eventos parceiros, que recebem eventos de parceiros de SaaS.

Os casos de uso comuns para barramentos de eventos incluem:

- Usar um barramento de eventos como um agente entre diferentes workloads, serviços ou sistemas.
- Usar vários barramentos de eventos em suas aplicações para dividir o tráfego do evento. Por exemplo, criar um barramento para processar eventos contendo informações de identificação pessoal (PII) e outro barramento para eventos que não o façam.

- Agregar eventos enviando eventos de vários barramentos de eventos para um barramento de eventos centralizado. Este barramento centralizado pode estar na mesma conta que os outros barramentos, mas também pode estar em uma conta ou região diferente.



Eventos

Em sua forma mais simples, um EventBridge evento é um objeto JSON enviado para um barramento ou canal de eventos.

No contexto da arquitetura orientada a eventos (EDA), um evento geralmente representa um indicador de uma mudança em um recurso ou ambiente.

Para ter mais informações, consulte ???.

Origens de eventos

EventBridge pode receber eventos de fontes de eventos, incluindo:

- AWS serviços
- Aplicações personalizadas
- Parceiros de software como serviço (SaaS)

Regras

Uma regra recebe eventos de entrada e os encaminha como adequados para os destinos para processamento. É possível especificar como cada regra invoca os destinos com base em:

- Um padrão de evento, que contém um ou mais filtros para combinar eventos. Os padrões de eventos podem incluir filtros que correspondem a:
  - Metadados do evento: dados sobre o evento, como a origem do evento ou a conta ou região na qual o evento se originou.
  - Dados do evento: as propriedades do evento em si. Estas propriedades variam de acordo com o evento.
  - Conteúdo do evento: os valores reais das propriedades dos dados do evento.
- Um cronograma para invocar os destinos em intervalos regulares.

  Você pode especificar uma regra programada dentro EventBridge ou usando o EventBridge Scheduler.

> **Note**
>
> EventBridge oferece o Amazon EventBridge Scheduler, um programador sem servidor que permite criar, executar e gerenciar tarefas a partir de um serviço gerenciado central. EventBridge O Scheduler é altamente personalizável e oferece escalabilidade aprimorada em relação às regras EventBridge programadas, com um conjunto mais amplo de operações e serviços de API de destino. AWS
> Recomendamos que você use o EventBridge Scheduler para invocar alvos em uma agenda. Para ter mais informações, consulte ???.

Cada regra é definida para um barramento de eventos específico e se aplica somente aos eventos desse barramento de eventos.

Uma única regra pode enviar um evento para até cinco destinos.

Por padrão, é possível configurar até 300 regras por barramento de eventos. Esta cota pode ser aumentada para milhares de regras no console Service Quotas. Como o limite de regras se aplica a cada barramento, se precisar de ainda mais regras, poderá criar barramentos de eventos personalizados adicionais em sua conta.

Também é possível personalizar a forma como os eventos são recebidos em sua conta criando barramentos de eventos com permissões diferentes para diferentes serviços.

Para personalizar a estrutura ou a data de um evento antes de EventBridge passá-lo para um destino, use o transformador de entrada para editar as informações antes que elas cheguem ao destino.

Para ter mais informações, consulte ???.

Destinos

Um destino é um recurso ou endpoint para o qual EventBridge envia um evento quando o evento corresponde ao padrão de evento definido para uma regra.

Um destino pode receber vários eventos de vários barramentos de eventos.

Para ter mais informações, consulte ???.

Atributos avançados para barramentos de eventos

EventBridge inclui os seguintes recursos para ajudá-lo a desenvolver, gerenciar e usar barramentos de eventos.

Como usar destinos de API para habilitar chamadas de API REST entre serviços

EventBridge Os destinos da API são endpoints HTTP que você pode definir como destino de uma regra, da mesma forma que enviaria dados de eventos para um AWS serviço ou recurso. Ao usar destinos de API, é possível usar chamadas de API para rotear eventos entre serviços da AWS , aplicações de SaaS integradas e suas aplicações externas à AWS. Ao criar um destino da API,

é especificada uma conexão a ser usada para isso. Cada conexão inclui detalhes sobre o tipo de autorização e os parâmetros a serem usados para autorizar com o endpoint de destino da API.

Arquivamento e reprodução de eventos para ajudar no desenvolvimento e na recuperação de desastres

Também é possível arquivar ou salvar eventos e reproduzi-los posteriormente a partir do arquivo. O arquivamento é útil para:

- Testar uma aplicação porque você tem um repositório de eventos para usar em vez de ter que esperar por novos eventos.
- Hidratar um novo serviço quando ele é disponibilizado pela primeira vez on-line.
- Adicionar mais durabilidade às suas aplicações orientadas por eventos.

Como usar o Schema Registry para iniciar rapidamente a criação de padrões de eventos

Ao criar aplicativos sem servidor que usam EventBridge, pode ser útil conhecer a estrutura de eventos típicos sem precisar gerar o evento. A estrutura do evento é descrita em esquemas, que estão disponíveis para todos os eventos gerados pelos AWS serviços em EventBridge.

Para eventos que não vêm de AWS serviços, você pode:

- Criar ou fazer upload de esquemas personalizados.
- Use o Schema Discovery para criar EventBridge automaticamente esquemas para eventos enviados ao barramento de eventos.

Depois de ter encontrado ou criado um esquema para um evento, faça download das vinculações de código para linguagens de programação populares.

Como gerenciar o acesso a recursos com políticas

Para organizar AWS recursos ou monitorar custos EventBridge, você pode atribuir uma etiqueta ou tag personalizada aos AWS recursos. Usando políticas baseadas em tags, você pode controlar o que os recursos podem ou não fazer dentro EventBridge delas.

Além das políticas baseadas em tags, EventBridge oferece suporte a políticas baseadas em identidade e recursos para controlar o acesso a. EventBridge Use políticas baseadas em identidade para controlar as permissões de um grupo, perfil ou usuário. Use políticas baseadas em recursos

para dar permissões específicas a cada recurso, como uma função do Lambda ou um tópico do Amazon SNS.

Criando um ônibus de EventBridge eventos da Amazon

É possível criar um barramento de eventos personalizado para receber eventos de suas aplicações personalizadas. Suas aplicações também podem enviar eventos para seu barramento de eventos padrão. Ao criar um barramento de eventos, é possível anexar uma política baseada em recursos para conceder permissões a outras contas. Outras contas podem enviar eventos para o barramento de eventos na conta atual.

O seguinte vídeo mostra a criação de barramentos de eventos: Como criar um barramento de eventos

Como criar um barramento de eventos personalizado

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, selecione Event buses (Barramentos de eventos).
3. Selecione Create event bus (Criar barramento de eventos).
4. Insira um nome para o novo barramento de eventos.
5. Escolha o KMS key para usar EventBridge ao criptografar os dados do evento armazenados no barramento de eventos.

> **Note**
>
> Arquivos e descoberta de esquemas não são compatíveis com barramentos de eventos criptografados usando um chave gerenciada pelo cliente. Para habilitar arquivamentos ou descoberta de esquemas em um barramento de eventos, escolha usar um Chave pertencente à AWS. Para ter mais informações, consulte ???.

- Escolha Usar Chave pertencente à AWS EventBridge para criptografar os dados usando um Chave pertencente à AWS.

  Chave pertencente à AWS É um KMS key que EventBridge possui e gerencia para uso em várias AWS contas. Em geral, a menos que você precise auditar ou controlar a chave de criptografia que protege seus recursos, uma Chave pertencente à AWS é uma boa escolha.

Esse é o padrão.

- Escolha Usar chave gerenciada pelo cliente EventBridge para criptografar os dados usando o chave gerenciada pelo cliente que você especifica ou cria.

  Chaves gerenciadas pelo cliente estão KMS keys na sua AWS conta que você cria, possui e gerencia. Você tem controle total sobre eles KMS keys.

  a. Especifique um existente chave gerenciada pelo cliente ou escolha Criar um novo KMS key.

     EventBridge exibe o status da chave e quaisquer aliases de chave que tenham sido associados ao especificado chave gerenciada pelo cliente.

  b. Escolha a fila do Amazon SQS para usar como fila de mensagens mortas (DLQ) para esse barramento de eventos, se houver.

     EventBridge envia eventos que não foram criptografados com êxito para o DLQ, se configurados, para que você possa processá-los posteriormente.

6. Configure os recursos opcionais do barramento de eventos:

   - Especifique uma política baseada em recursos fazendo o seguinte:
     - Insira a política que inclui as permissões a serem concedidas para o barramento de eventos. É possível colar uma política de outra origem ou inserir o JSON da política. Você pode usar uma das políticas de exemplo e modificá-la para seu ambiente.
     - Para usar um modelo para a política, escolha Carregar modelo. Modifique a política conforme adequado para seu ambiente, incluindo a adição de ações adicionais que a entidade principal na política está autorizada a usar.

     Para obter mais informações sobre como conceder permissões a um barramento de eventos por meio de políticas baseadas em recursos, consulte. ???

   - Ativar um arquivamento (opcional)

     Você pode criar um arquivo de eventos para poder reproduzi-los facilmente mais tarde. Por exemplo, você pode querer repetir eventos para se recuperar de erros ou validar uma nova funcionalidade em sua aplicação. Para mais informações, consulte ???.

     a. Em Arquivos, escolha Ativado.

     b. Especifique um nome e uma descrição para o arquivo.

> **Note**
>
> Arquivos e descoberta de esquemas não são compatíveis com barramentos de eventos criptografados usando um chave gerenciada pelo cliente. Para habilitar arquivamentos ou descoberta de esquemas em um barramento de eventos, escolha usar um Chave pertencente à AWS. Para ter mais informações, consulte ???.

- Ativar a descoberta do esquema (opcional)

  Ative a descoberta de esquemas para inferir EventBridge automaticamente os esquemas diretamente dos eventos em execução nesse barramento de eventos. Para mais informações, consulte ???.

  a. Em Descoberta do esquema, escolha Ativado.

> **Note**
>
> Arquivos e descoberta de esquemas não são compatíveis com barramentos de eventos criptografados usando um chave gerenciada pelo cliente. Para habilitar arquivamentos ou descoberta de esquemas em um barramento de eventos, escolha usar um Chave pertencente à AWS. Para ter mais informações, consulte ???.

- Especifique as tags (opcional)

  Uma tag é um rótulo de atributo personalizado que você atribui a um AWS recurso. Use tags para identificar e organizar seus AWS recursos. Muitos AWS serviços oferecem suporte à marcação, então você pode atribuir a mesma tag a recursos de serviços diferentes para indicar que os recursos estão relacionados. Para mais informações, consulte ???.

  a. Em Etiquetas, escolha Adicionar nova etiqueta.

  b. Especifique uma chave e, opcionalmente, um valor para a nova tag.

7. Escolha Create (Criar).

Atualizando um ônibus de EventBridge eventos da Amazon

Você pode atualizar a configuração dos barramentos de eventos depois de criá-los. Isso inclui o barramento de eventos padrão, que é EventBridge criado automaticamente na sua conta.

Atualizando o KMS key usado para criptografia

> **Note**
>
> Arquivos e descoberta de esquemas não são compatíveis com barramentos de eventos criptografados usando um chave gerenciada pelo cliente. Para habilitar arquivamentos ou descoberta de esquemas em um barramento de eventos, escolha usar um Chave pertencente à AWS. Para ter mais informações, consulte ???.

Para alterar o KMS key usado para criptografia em repouso em um barramento de eventos usando o EventBridge console

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, selecione Event buses (Barramentos de eventos).
3. Escolha o ônibus do evento que você deseja atualizar.
4. Na página de detalhes do barramento de eventos, escolha a guia Criptografia.
5. Escolha o KMS key para usar EventBridge ao criptografar os dados do evento armazenados no barramento de eventos:
   - Escolha Usar Chave pertencente à AWS EventBridge para criptografar os dados usando um Chave pertencente à AWS.

     Chave pertencente à AWS É um KMS key que EventBridge possui e gerencia para uso em várias AWS contas. Em geral, a menos que você precise auditar ou controlar a chave de criptografia que protege seus recursos, uma Chave pertencente à AWS é uma boa escolha.

     Esse é o padrão.
   - Escolha Usar chave gerenciada pelo cliente EventBridge para criptografar os dados usando o chave gerenciada pelo cliente que você especifica ou cria.

     Chaves gerenciadas pelo cliente estão KMS keys na sua AWS conta que você cria, possui e gerencia. Você tem controle total sobre eles KMS keys.

     a. Especifique um existente chave gerenciada pelo cliente ou escolha Criar um novo KMS key.

EventBridge exibe o status da chave e quaisquer aliases de chave que tenham sido associados ao especificado chave gerenciada pelo cliente.

b. Escolha a fila do Amazon SQS para usar como fila de mensagens mortas (DLQ) para esse barramento de eventos, se houver.

   EventBridge envia eventos que não foram criptografados com êxito para o DLQ, se configurados, para que você possa processá-los posteriormente.

Atualização de permissões em um barramento de eventos

É possível conceder permissões adicionais a um barramento de eventos anexando uma política baseada em recursos a ele. Para obter instruções detalhadas sobre como atualizar as permissões concedidas a um barramento de eventos, consulte Gerenciamento de permissões de barramento de eventos.

Adicionar ou remover arquivos em barramentos de eventos

Um arquivamento permite capturar eventos para que você possa reproduzi-los facilmente mais tarde. Por exemplo, você pode querer repetir eventos para se recuperar de erros ou validar uma nova funcionalidade em sua aplicação. Para obter mais informações, consulte EventBridge arquivar e reproduzir.

> **Note**
>
> Arquivos e descoberta de esquemas não são compatíveis com barramentos de eventos criptografados usando um chave gerenciada pelo cliente. Para habilitar arquivamentos ou descoberta de esquemas em um barramento de eventos, escolha usar um Chave pertencente à AWS. Para ter mais informações, consulte ???.

Para adicionar ou remover um arquivo de um barramento de eventos usando o EventBridge console

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, selecione Event buses (Barramentos de eventos).
3. Escolha o ônibus do evento que você deseja atualizar.
4. Na página de detalhes do ônibus de eventos, escolha a guia Arquivos.

5. Execute um destes procedimentos:
   - Para adicionar um arquivo:
     a. Escolha Criar arquivo.
     b. Especifique atributos para o arquivo.
     c. Selecione Next (Próximo).
     d. Escolha o padrão de eventos a ser aplicado aos eventos do arquivo.
     e. Escolha Criar arquivo.
   - Para excluir um arquivo:
     a. Para a tag que você deseja remover, escolha Excluir.
     b. Insira o nome do arquivo e escolha Excluir.

        O arquivo é excluído permanentemente. Você não pode desfazer esta operação.

Para criar ou excluir um arquivo para um barramento de eventos usando o AWS CLI

- Para criar um arquivo, use create-archive.

  Para excluir permanentemente um arquivo, use delete-archive.

Iniciando ou interrompendo a descoberta de esquemas em ônibus de eventos

Para obter mais informações sobre a descoberta de esquemas, consulte EventBridge esquemas.

> **Note**
>
> Arquivos e descoberta de esquemas não são compatíveis com barramentos de eventos criptografados usando um chave gerenciada pelo cliente. Para habilitar arquivamentos ou descoberta de esquemas em um barramento de eventos, escolha usar um Chave pertencente à AWS. Para ter mais informações, consulte ???.

Para iniciar ou interromper a descoberta do esquema em um barramento de eventos usando o console EventBridge

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, selecione Event buses (Barramentos de eventos).
3. Escolha o ônibus do evento que você deseja atualizar.
4. Execute um destes procedimentos:
   - Para iniciar a descoberta do esquema, escolha Iniciar descoberta.
   - Para interromper a descoberta do esquema, escolha Excluir descoberta.

Para iniciar ou interromper a descoberta do esquema em um barramento de eventos usando o AWS CLI

- Para iniciar a descoberta do esquema, use create-discoverer.

  Para interromper a descoberta do esquema, use delete-discoverer.

Adicionar ou remover tags em ônibus de eventos

Uma tag é um rótulo de atributo personalizado que você atribui ou AWS atribui a um AWS recurso. Use tags para identificar e organizar seus AWS recursos. Para obter mais informações, consulte EventBridge tags.

Para adicionar ou remover tags de um barramento de eventos usando o EventBridge console

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, selecione Event buses (Barramentos de eventos).
3. Escolha o ônibus do evento que você deseja atualizar.
4. Na página de detalhes do ônibus de eventos, escolha a guia Tags e, em seguida, escolha Gerenciar tags.
5. Execute um destes procedimentos:
   - Para adicionar uma tag:
     a. Selecione Adicionar nova tag.
     b. Especifique a chave e o valor da tag
     c. Selecione Atualizar.

- Para remover uma tag:

  a. Para a tag que você deseja remover, escolha Remover.

  b. Selecione Atualizar.

Para adicionar ou remover tags de um barramento de eventos usando o AWS CLI

- Para adicionar tags, use tag-resource.

  Para remover tags, use untag-resource.

Atualizando o barramento de eventos padrão usando AWS CloudFormation

AWS CloudFormation permite que você configure e gerencie seus AWS recursos em contas e regiões de forma centralizada e repetível, tratando a infraestrutura como código. CloudFormation faz isso permitindo que você crie modelos, que definem os recursos que você deseja provisionar e gerenciar.

Como EventBridge provisiona o barramento de eventos padrão em sua conta automaticamente, você não pode criá-lo usando um CloudFormation modelo, como faria normalmente com qualquer recurso que desejasse incluir em uma CloudFormation pilha. Para incluir o barramento de eventos padrão em uma CloudFormation pilha, você deve primeiro importá-lo para uma pilha. Depois de importar o barramento de eventos padrão para uma pilha, você pode atualizar as propriedades do barramento de eventos conforme desejado.

Para importar um recurso existente em uma CloudFormation pilha nova ou existente, você precisa das seguintes informações:

- Um identificador exclusivo para o recurso a ser importado.

  Para barramentos de eventos padrão, o identificador é `Name` e, em seguida, o valor do identificador é`default`.

- Um modelo que descreve com precisão as propriedades atuais do recurso existente.

  O trecho do modelo abaixo contém um `AWS::Events::EventBus` recurso que descreve as propriedades atuais de um barramento de eventos padrão. Neste exemplo, o barramento de eventos foi configurado para usar a chave gerenciada pelo cliente e DLQ para criptografia em repouso.

Além disso, o `AWS::Events::EventBus` recurso que descreve o barramento de eventos padrão que você deseja importar deve incluir uma `DeletionPolicy` propriedade definida como`Retain`.

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Default event bus import example",
    "Resources": {
        "defaultEventBus": {
            "Type" : "AWS::Events::EventBus",
            "DeletionPolicy": "Retain",
            "Properties" : {
                "Name" : "default",
                "KmsKeyIdentifier" : "KmsKeyArn",
                "DeadLetterConfig" : {
                    "Arn" : "DLQ_ARN"
                }
            }
        }
    }
}
```

Para obter mais informações, consulte Como CloudFormation gerenciar os recursos existentes no Guia do CloudFormation usuário.

Excluindo um ônibus de EventBridge eventos da Amazon

Você pode excluir um ônibus de eventos personalizado ou de um parceiro. Você não pode excluir o barramento de eventos padrão. A exclusão de um barramento de eventos exclui as regras associadas a esse barramento de eventos.

Para excluir um barramento de eventos usando o EventBridge console

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, selecione Event buses (Barramentos de eventos).
3. Escolha o ônibus de eventos que você deseja excluir.
4. Execute um destes procedimentos:
   - Escolha Excluir.
   - Escolha o nome do ônibus do evento.

Na página de detalhes do ônibus do evento, escolha Excluir.

Permissões de barramentos de evento do Amazon EventBridge Pipes

O barramento de eventos padrão em sua conta da AWS só permite eventos de uma conta. É possível conceder permissões adicionais a um barramento de eventos anexando uma política baseada em recursos a ele. Com uma política baseada em recursos, é possível permitir `PutEventsPutRule`, e chamadas de API `PutTargets` de outra conta. Também é possível usar as condições do IAM na política para conceder permissões a uma organização, aplicar tags ou filtrar eventos somente para aqueles de uma regra ou conta específica. É possível definir uma política baseada em recursos para um barramento de eventos ao criá-lo ou posteriormente.

APIs do EventBridge que aceitam um parâmetro `Name` de barramento de eventos, como `PutRule`, `PutTargets`, `DeleteRule`, `RemoveTargets`, `DisableRule` e `EnableRule`, e também aceitam o ARN do barramento de eventos. Use esses parâmetros para referenciar barramentos de eventos entre contas ou regiões por meio das APIs. Por exemplo, você pode chamar `PutRule` para criar uma regra em um barramento de eventos em uma conta diferente sem precisar assumir um perfil.

É possível anexar as políticas de exemplo neste tópico a um perfil do IAM para conceder permissão para enviar eventos para uma conta ou região diferente. Use os perfis do IAM para definir políticas de controle organizacional e limites sobre quem pode enviar eventos da sua conta para outras contas. Recomendamos sempre usar perfis do IAM quando o destino de uma regra é um barramento de eventos. É possível anexar perfis do IAM usando chamadas `PutTarget`. Para obter informações sobre como criar uma regra para enviar eventos para uma conta ou região diferente, consulte Enviar e receber EventBridge eventos da Amazon entre AWS contas.

Tópicos

- Exemplo de política: envia eventos somente de uma região específica para uma região diferente
- Exemplo de política: nega o envio de eventos de regiões específicas

Gerenciamento de permissões de barramento de eventos

Use o procedimento a seguir para modificar as permissões de um barramento de eventos existente. Para obter informações sobre como usar o AWS CloudFormation para criar uma política de barramento de eventos, consulte AWS::Events::EventBusPolicy.

Para gerenciar permissões para um barramento de eventos existente

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação à esquerda, escolha Barramentos de eventos.
3. Em Nome, escolha o nome do barramento de eventos do para o qual gerenciar as permissões.

   Se uma política de recursos estiver conectada ao barramento de eventos, a política será exibida.
4. Escolha Gerenciar permissões e siga um destes procedimentos:
   - Insira a política que inclui as permissões a serem concedidas para o barramento de eventos. É possível colar uma política de outra origem ou inserir o JSON da política.
   - Para usar um modelo para a política, escolha Carregar modelo. Modifique a política conforme adequado para seu ambiente e adicione ações adicionais que a entidade principal na política está autorizada a usar.
5. Escolha Atualizar.

O modelo fornece exemplos de declarações de política que você pode personalizar para sua conta e seu ambiente. O modelo não é uma política válida. É possível modificar o modelo para seu caso de uso ou copiar uma das políticas de exemplo e personalizá-la.

O modelo carrega políticas que incluem um exemplo de como conceder permissões a uma conta para usar a ação `PutEvents`, como conceder permissões a uma organização e como conceder permissões à conta para gerenciar regras na conta. Também é possível personalizar o modelo para sua conta específica e excluir as outras seções do modelo. Mais exemplos de políticas estão incluídos adiante neste tópico.

Se tentar atualizar as permissões do barramento, mas a política contiver um erro, uma mensagem de erro indicará o problema específico na política.

```
 ### Choose which sections to include in the policy to match your use case. ###
 ### Be sure to remove all lines that start with ###, including the ### at the end of
the line. ###

 ### The policy must include the following: ###

 {
   "Version": "2012-10-17",
   "Statement": [

     ### To grant permissions for an account to use the PutEvents action, include the
following, otherwise delete this section: ###

     {

       "Sid": "AllowAccountToPutEvents",
       "Effect": "Allow",
       "Principal": {
         "AWS": "<ACCOUNT_ID>"
       },
       "Action": "events:PutEvents",
       "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/default"
     },

     ### Include the following section to grant permissions to all members of your AWS
Organizations to use the PutEvents action ###

     {
       "Sid": "AllowAllAccountsFromOrganizationToPutEvents",
       "Effect": "Allow",
       "Principal": "*",
       "Action": "events:PutEvents",
       "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/default",
       "Condition": {
         "StringEquals": {
           "aws:PrincipalOrgID": "o-yourOrgID"
         }
       }
     },

     ### Include the following section to grant permissions to the account to manage
the rules created in the account ###
```

```
      {
        "Sid": "AllowAccountToManageRulesTheyCreated",
        "Effect": "Allow",
        "Principal": {
          "AWS": "<ACCOUNT_ID>"
        },
        "Action": [
          "events:PutRule",
          "events:PutTargets",
          "events:DeleteRule",
          "events:RemoveTargets",
          "events:DisableRule",
          "events:EnableRule",
          "events:TagResource",
          "events:UntagResource",
          "events:DescribeRule",
          "events:ListTargetsByRule",
          "events:ListTagsForResource"],
        "Resource": "arn:aws:events:us-east-1:123456789012:rule/default",
        "Condition": {
          "StringEqualsIfExists": {
            "events:creatorAccount": "<ACCOUNT_ID>"
          }
        }
    }]
  }
```

Exemplo de política: envia eventos para um barramento padrão em uma conta diferente

O exemplo de política a seguir concede à conta 111122223333 permissão para publicar eventos no barramento de eventos padrão na conta 123456789012.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
        "Sid": "sid1",
        "Effect": "Allow",
        "Principal": {"AWS":"arn:aws:iam::111112222333:root"},
        "Action": "events:PutEvents",
```

```
        "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/default"
        }
    ]
  }
```

Exemplo de política: envia eventos para um barramento personalizado em uma conta diferente

O exemplo de política a seguir concede à conta 111122223333 permissão para publicar eventos no `central-event-bus` na conta 123456789012, mas somente para eventos com um valor de origem definido como `com.exampleCorp.webStore` e um `detail-type` definido como `newOrderCreated`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WebStoreCrossAccountPublish",
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/central-event-bus",
      "Condition": {
        "StringEquals": {
          "events:detail-type": "newOrderCreated",
          "events:source": "com.exampleCorp.webStore"
        }
      }
    }
  ]
}
```

Exemplo de política: envia eventos para um barramento de eventos na mesma conta

O exemplo de política a seguir anexado a um barramento de eventos chamado `CustomBus1` permite que o barramento de eventos receba eventos da mesma conta e região.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:123456789:event-bus/CustomBus1"
      ]
    }
  ]
}
```

Exemplo de política: envia eventos para a mesma conta e restringir atualizações

O exemplo de política a seguir concede à conta 123456789012 permissão para criar, excluir, atualizar, desativar e habilitar regras, além de adicionar ou remover destinos. Ele limita essas regras que correspondem a eventos com uma origem de `com.exampleCorp.webStore` e usa o `"events:creatorAccount": "${aws:PrincipalAccount}"` para garantir que somente a conta 123456789012 possa modificar essas regras e origens depois de criados.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "InvoiceProcessingRuleCreation",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": [
```

```
        "events:PutRule",
        "events:DeleteRule",
        "events:DescribeRule",
        "events:DisableRule",
        "events:EnableRule",
        "events:PutTargets",
        "events:RemoveTargets"
      ],
      "Resource": "arn:aws:events:us-east-1:123456789012:rule/central-event-bus/*",
      "Condition": {
        "StringEqualsIfExists": {
          "events:creatorAccount": "${aws:PrincipalAccount}",
          "events:source": "com.exampleCorp.webStore"
        }
      }
    }
  ]
}
```

Exemplo de política: envie eventos somente de uma regra específica para o barramento em uma região diferente

O exemplo de política a seguir concede à conta 111122223333 permissão para enviar eventos que correspondam a uma regra nomeada `SendToUSE1AnotherAccount` nas regiões do Oriente Médio (Bahrein) e Oeste dos EUA (Oregon) para um barramento de eventos chamado `CrossRegionBus`no Leste dos EUA (Norte da Virgínia) na conta 123456789012. O exemplo de política é adicionado ao barramento de eventos chamado `CrossRegionBus` na conta 123456789012. A política permite eventos somente se eles corresponderem a uma regra especificada para o barramento de eventos na conta 111122223333. A declaração `Condition` restringe os eventos somente aos eventos que correspondam às regras com o ARN da regra especificada.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSpecificRulesAsCrossRegionSource",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
```

```
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:events:us-west-2:111112222333:rule/CrossRegionBus/
SendToUSE1AnotherAccount",
            "arn:aws:events:me-south-1:111112222333:rule/CrossRegionBus/
SendToUSE1AnotherAccount"
          ]
        }
      }
    }
  ]
}
```

Exemplo de política: envia eventos somente de uma região específica para uma região diferente

O exemplo de política a seguir concede à conta 111122223333 permissão para enviar todos os eventos gerados nas regiões do Oriente Médio (Bahrein) e Oeste dos EUA (Oregon) para um barramento de eventos chamado `CrossRegionBus` no Leste dos EUA (Norte da Virgínia) na conta 123456789012. A conta 111122223333 não tem permissão para enviar eventos gerados em nenhuma outra região.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossRegionEventsFromUSWest2AndMESouth1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:events:us-west-2:*:*",
            "arn:aws:events:me-south-1:*:*"
          ]
```

```
        }
      }
    }
  ]
}
```

Exemplo de política: nega o envio de eventos de regiões específicas

O exemplo de política a seguir anexado a um barramento de eventos chamado `CrossRegionBus` na conta 123456789012 concede permissão para que o barramento de eventos receba eventos da conta 111122223333, mas não eventos gerados na região Oeste dos EUA (Oregon).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "1AllowAnyEventsFromAccount111112222333",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus"
    },
    {
      "Sid": "2DenyAllCrossRegionUSWest2Events",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:events:us-west-2:*:*"
          ]
        }
      }
    }
  ]
}
```

Gere um modelo do AWS CloudFormation a partir de um barramento de eventos do Amazon EventBridge

O AWS CloudFormation permite configurar e gerenciar recursos da AWS entre contas e regiões de forma centralizada e repetível tratando a infraestrutura como código. O CloudFormation faz isso permitindo que você crie modelos que definem os recursos que deseja provisionar e gerenciar.

O EventBridge permite que você gere modelos dos barramentos de eventos existentes em sua conta, para ajudar a começar a desenvolver modelos do CloudFormation. Além disso, o EventBridge oferece a opção de incluir as regras associadas a esse barramento de eventos em seu modelo. É possível utilizar esses modelos como base para criar pilhas de recursos sob o gerenciamento do CloudFormation.

Para obter mais informações sobre o CloudFormation, consulte o Guia do usuário do AWS CloudFormation.

> **Note**
>
> O EventBridge não inclui regras gerenciadas no modelo gerado.

Também é possível gerar um modelo a partir de uma ou mais regras contidas em um barramento de eventos selecionado.

Como gerar um modelo do CloudFormation de um barramento de eventos

1. Abra o console Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, selecione Event buses (Barramentos de eventos).
3. Escolha o barramento de eventos do qual deseja gerar um modelo do CloudFormation.
4. No menu Ações, escolha Modelo do CloudFormation e escolha em qual formato deseja que o EventBridge gere o modelo: JSON ou YAML.

   O EventBridge exibe o modelo, gerado no formato selecionado. Por padrão, todas as regras associadas ao barramento de eventos são incluídas no modelo.

   - Para gerar o modelo sem incluir regras, desmarque Incluir regras neste EventBus.
5. O EventBridge oferece a opção de baixar o arquivo de modelo ou copiar o modelo para a área de transferência.

- Para baixar o arquivo de modelo, escolha Baixar.
- Para copiar o modelo para a área de transferência, escolha Copiar.

6. Para sair do modelo, escolha Cancelar.

Depois de personalizar o modelo do AWS CloudFormation conforme necessário para o caso de uso, é possível utilizá-lo para criar pilhas no CloudFormation.

Considerações ao utilizar modelos do CloudFormation gerados no Amazon EventBridge

Considere os seguintes fatores ao utilizar um modelo do CloudFormation gerado de um barramento de eventos:

- O EventBridge não inclui nenhuma senha no modelo gerado.

  É possível editar o modelo para incluir parâmetros de modelo que permitam aos usuários especificar senhas ou outras informações sensitivas ao utilizar o modelo para criar ou atualizar uma pilha do CloudFormation.

  Além disso, os usuários podem utilizar o Secrets Manager para criar um segredo na região desejada e editá-lo para utilizar parâmetros dinâmicos.
- Os destinos no modelo gerado permanecem exatamente como foram especificados no barramento de eventos original. Isso poderá resultar em problemas entre regiões se você não editar adequadamente o modelo antes de utilizá-lo para criar pilhas em outras regiões.

  Além disso, o modelo gerado não criará os destinos downstream automaticamente.

EventBridge Eventos da Amazon

Um evento indica uma alteração em um ambiente, como um ambiente da AWS , um serviço ou uma aplicação de parceiro de SaaS ou uma das suas próprias aplicações ou serviços personalizados. A seguir estão os exemplos de eventos.

- O Amazon EC2 gera um evento quando o estado de uma instância muda de pendente para em execução.
- O Amazon EC2 Auto Scaling gera eventos ao executar ou encerrar instâncias.
- AWS CloudTrail publica eventos quando você faz chamadas de API.

Você também pode configurar eventos programados a serem gerados periodicamente.

Para obter uma lista de serviços que geram eventos e eventos de amostra de cada serviço, consulte Eventos de AWS serviços e siga os links na tabela.

Os eventos são representados como objetos JSON e todos têm uma estrutura semelhante e os mesmos campos de nível superior.

O conteúdo do campo de nível superior detail será diferente dependendo de qual serviço gerou o evento e do que ele trata. A combinação dos campos source e detail-type serve para identificar os campos e os valores encontrados no campo detail. Para obter exemplos de eventos gerados por AWS serviços, consulteEventos de AWS serviços.

Tópicos

- Referência de estrutura de eventos
- Adicionando EventBridge eventos da Amazon com PutEvents
- Eventos de AWS serviços
- Recebendo eventos de um parceiro SaaS com a Amazon EventBridge
- Como depurar os eventos de entrega

O seguinte vídeo explica os conceitos básicos dos eventos: O que é um evento

O vídeo a seguir aborda as formas como os eventos chegam EventBridge: De onde vêm os eventos

Referência de estrutura de eventos

Os campos a seguir aparecem em todos os eventos entregues em um barramento de eventos e incluem os metadados do evento:

```
{
  "???": "0",
  "???": "UUID",
  "???": "event name",
  "???": "event source",
  "???": "ARN",
  "???": "timestamp",
  "???": "region",
  "???": [
    "ARN"
  ],
  "???": {
    JSON object
  }
}
```

versão

Por padrão, isso é definido como 0 (zero) em todos os eventos.

id

Um UUID versão 4 gerado para cada evento. É possível usar um `id` para rastrear eventos à medida que eles percorrem as regras até os destinos.

detail-type

Identifica, em combinação com o campo source, os campos e os valores que serão exibidos no campo detail.

Eventos que são entregues por CloudTrail têm `AWS API Call via CloudTrail` como valor`detail-type`.

origem

Identifica o serviço que gerou o evento. Todos os eventos provenientes dos serviços da AWS começam com "aws". Os eventos gerados pelo cliente podem ter qualquer valor aqui, desde que ele não comece com "aws". Recomendamos o uso de strings de nome de domínio inversas no estilo pacote-nome Java.

Para encontrar o valor correto `source` para um AWS serviço, consulte a tabela de chaves de condição, selecione um serviço na lista e procure o prefixo do serviço. Por exemplo, o `source` valor para a Amazon CloudFront é `aws.cloudfront`.

conta

O número de 12 dígitos que identifica uma AWS conta.

horário

O time stamp do evento, que pode ser especificado pelo serviço que originou o evento. Se o evento abranger um intervalo de tempo, o serviço poderá informar o horário de início, portanto, esse valor poderá ser anterior à hora em que o evento é recebido.

região

Identifica a AWS região onde o evento se originou.

recursos

Esta matriz JSON contém ARNs que identificam os recursos que estão envolvidos no evento. O serviço que gera o evento determina se esses ARNs devem ser incluídos. Por exemplo, as alterações de estado da instância do Amazon EC2 incluem os ARNs da instância do Amazon EC2, e os eventos do Auto Scaling incluem os ARNs de instâncias e grupos do Auto Scaling, mas as chamadas de API com o AWS CloudTrail não incluem os ARNs de recursos.

detalhe

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo. Pode ser `"{}"`.

AWS Os eventos de chamada de API têm objetos detalhados com aproximadamente 50 campos aninhados em vários níveis de profundidade.

> **Note**
>
> PutEventsaceita dados no formato JSON. Para o tipo de dados do número JSON (número inteiro), as restrições são: um valor mínimo de -9.223.036.036.854.775.807.

Example Exemplo: notificação de alteração do estado da instância do Amazon EC2

O evento a seguir na Amazon EventBridge indica que uma instância do Amazon EC2 está sendo encerrada.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ec2:us-west-1:123456789012:instance/i-1234567890abcdef0"
  ],
  "detail": {
    "instance-id": " i-1234567890abcdef0",
    "state": "terminated"
  }
}
```

Informações mínimas necessárias para um evento personalizado válido

Ao criar eventos personalizados, eles devem incluir os seguintes campos:

- `detail`
- `detail-type`
- `source`

```
{
  "detail-type": "event name",
  "source": "event source",
  "detail": {
  }
}
```

Adicionando EventBridge eventos da Amazon com **`PutEvents`**

A `PutEvents` ação envia vários eventos EventBridge em uma única solicitação. Para obter mais informações, consulte PutEventsna Amazon EventBridge API Reference e put-events na AWS CLI Command Reference.

Cada solicitação `PutEvents` pode oferecer suporte a um número limitado de entradas. Para ter mais informações, consulte Cotas do Amazon EventBridge. A operação `PutEvents` tenta processar todas as entradas na ordem natural da solicitação. Depois de ligar`PutEvents`, EventBridge atribua a cada evento uma ID exclusiva.

Tópicos

- Como lidar com falhas com PutEvents
- Enviando eventos usando o AWS CLI
- Calculando o tamanho da entrada em EventBridge PutEvents eventos da Amazon

O exemplo de código Java a seguir envia dois eventos idênticos para EventBridge.

AWS SDK for Java Version 2.x

```
EventBridgeClient eventBridgeClient =
    EventBridgeClient.builder().build();

PutEventsRequestEntry requestEntry = PutEventsRequestEntry.builder()
    .resources("resource1", "resource2")
    .source("com.mycompany.myapp")
    .detailType("myDetailType")
    .detail("{ \"key1\": \"value1\", \"key2\": \"value2\" }")
    .build();

List <
PutEventsRequestEntry > requestEntries = new ArrayList <
PutEventsRequestEntry > ();
requestEntries.add(requestEntry);

PutEventsRequest eventsRequest = PutEventsRequest.builder()
    .entries(requestEntries)
    .build();

PutEventsResponse result = eventBridgeClient.putEvents(eventsRequest);

for (PutEventsResultEntry resultEntry: result.entries()) {
    if (resultEntry.eventId() != null) {
        System.out.println("Event Id: " + resultEntry.eventId());
    } else {
        System.out.println("PutEvents failed with Error Code: " +
 resultEntry.errorCode());
```

```
    }
}
```

AWS SDK for Java Version 1.0

```
EventBridgeClient eventBridgeClient =
    EventBridgeClient.builder().build();

PutEventsRequestEntry requestEntry = new PutEventsRequestEntry()
        .withTime(new Date())
        .withSource("com.mycompany.myapp")
        .withDetailType("myDetailType")
        .withResources("resource1", "resource2")
        .withDetail("{ \"key1\": \"value1\", \"key2\": \"value2\" }");

PutEventsRequest request = new PutEventsRequest()
        .withEntries(requestEntry, requestEntry);

PutEventsResult result = awsEventsClient.putEvents(request);

for (PutEventsResultEntry resultEntry : result.getEntries()) {
    if (resultEntry.getEventId() != null) {
        System.out.println("Event Id: " + resultEntry.getEventId());
    } else {
        System.out.println("Injection failed with Error Code: " +
 resultEntry.getErrorCode());
    }
}
```

Depois de executar esse código, o resultado `PutEvents` inclui uma matriz de entradas de resposta. Cada entrada na matriz de resposta corresponde diretamente com uma entrada na matriz em ordem, do início ao fim da solicitação e da resposta. A matriz `Entries` de resposta sempre inclui o mesmo número de entradas que a solicitação.

Como lidar com falhas com **`PutEvents`**

Por padrão, se uma entrada individual em uma solicitação falhar, EventBridge continuará processando o restante das entradas na solicitação. Uma matriz `Entries` de respostas pode incluir entradas com e sem êxito. É preciso detectar entradas processadas sem êxito e incluí-las em uma chamada subsequente.

As entradas de resultados com êxito incluem um valor `Id`, e as entradas de resultados sem êxito incluem valores `ErrorCode` e `ErrorMessage`. `ErrorCode`descreve o tipo de erro. `ErrorMessage` fornece mais informações sobre o erro. O exemplo a seguir tem três entradas de resultado de uma solicitação `PutEvents`. A segunda entrada não teve êxito.

```
{
    "FailedEntryCount": 1,
    "Entries": [
        {
            "EventId": "11710aed-b79e-4468-a20b-bb3c0c3b4860"
        },
        {   "ErrorCode": "InternalFailure",
            "ErrorMessage": "Internal Service Failure"
        },
        {
            "EventId": "d804d26a-88db-4b66-9eaf-9a11c708ae82"
        }
    ]
}
```

> **Note**
>
> Se você costuma `PutEvents` publicar um evento em um barramento de eventos que não existe, a correspondência de EventBridge eventos não encontrará uma regra correspondente e cancelará o evento. Embora EventBridge envie uma `200` resposta, ela não falhará na solicitação nem incluirá o evento no `FailedEntryCount` valor da resposta da solicitação.

É possível incluir as entradas que foram processadas sem êxito nas solicitações `PutEvents` subsequentes. Primeiro, para descobrir se há entradas com falha na solicitação, verifique o parâmetro `FailedRecordCount` em `PutEventsResult`. Se não for zero, cada `Entry` tenha um `ErrorCode` poderá ser adicionada, que não seja nula a uma solicitação subsequente. O exemplo a seguir mostra um manipulador com falha.

```
PutEventsRequestEntry requestEntry = new PutEventsRequestEntry()
        .withTime(new Date())
        .withSource("com.mycompany.myapp")
        .withDetailType("myDetailType")
        .withResources("resource1", "resource2")
        .withDetail("{ \"key1\": \"value1\", \"key2\": \"value2\" }");
```

```
List<PutEventsRequestEntry> putEventsRequestEntryList = new ArrayList<>();
for (int i = 0; i < 3; i++) {
    putEventsRequestEntryList.add(requestEntry);
}

PutEventsRequest putEventsRequest = new PutEventsRequest();
putEventsRequest.withEntries(putEventsRequestEntryList);
PutEventsResult putEventsResult = awsEventsClient.putEvents(putEventsRequest);

while (putEventsResult.getFailedEntryCount() > 0) {
    final List<PutEventsRequestEntry> failedEntriesList = new ArrayList<>();
    final List<PutEventsResultEntry> PutEventsResultEntryList =
 putEventsResult.getEntries();
    for (int i = 0; i < PutEventsResultEntryList.size(); i++) {
        final PutEventsRequestEntry putEventsRequestEntry =
 putEventsRequestEntryList.get(i);
        final PutEventsResultEntry putEventsResultEntry =
 PutEventsResultEntryList.get(i);
        if (putEventsResultEntry.getErrorCode() != null) {
            failedEntriesList.add(putEventsRequestEntry);
        }
    }
    putEventsRequestEntryList = failedEntriesList;
    putEventsRequest.setEntries(putEventsRequestEntryList);
    putEventsResult = awsEventsClient.putEvents(putEventsRequest);
    }
```

Enviando eventos usando o AWS CLI

Você pode usar o AWS CLI para enviar eventos personalizados para EventBridge que eles possam ser processados. O exemplo a seguir coloca um evento personalizado em EventBridge:

```
aws events put-events \
--entries '[{"Time": "2016-01-14T01:02:03Z", "Source": "com.mycompany.myapp",
 "Resources": ["resource1", "resource2"], "DetailType": "myDetailType", "Detail":
 "{ \"key1\": \"value1\", \"key2\": \"value2\" }"}]'
```

Também é possível criar um arquivo JSON que contenha eventos personalizados.

```
[
  {
```

```
    "Time": "2016-01-14T01:02:03Z",
    "Source": "com.mycompany.myapp",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }"
  }
]
```

Em seguida, para usar o AWS CLI para ler as entradas desse arquivo e enviar eventos, em um prompt de comando, digite:

```
aws events put-events --entries file://entries.json
```

Calculando o tamanho da entrada em EventBridge PutEvents eventos da Amazon

Você pode enviar eventos personalizados para EventBridge usando a `PutEvents` ação. Também é possível agrupar várias entradas de eventos em uma solicitação para maior eficiência. O tamanho total da entrada deve ser menor que 256 KB. Também é possível calcular o tamanho da entrada antes de enviar os eventos.

> **Note**
>
> O limite de tamanho é imposto na entrada. Mesmo que a entrada seja menor que o limite de tamanho, o evento em EventBridge é sempre maior que o tamanho da entrada devido aos caracteres e chaves necessários da representação JSON do evento. Para ter mais informações, consulte EventBridge Eventos da Amazon.

EventBridge calcula o `PutEventsRequestEntry` tamanho da seguinte forma:

- Se o parâmetro `Time` for especificado, ele será medido como 14 bytes.
- Os parâmetros `Source` e `DetailType` são o número de bytes para seus formulários codificados em UTF-8.
- Se o parâmetro `Detail` for especificado, ele será medido como o número de bytes para seu formato codificado UTF-8.
- Se o parâmetro `Resources` for especificado, cada entrada será medida como o número de bytes para seus formulários codificados UTF-8.

O código em Java de exemplo a seguir calcula o tamanho de um determinado objeto `PutEventsRequestEntry`:

```
int getSize(PutEventsRequestEntry entry) {
    int size = 0;
    if (entry.getTime() != null) {
        size += 14;
    }
    size += entry.getSource().getBytes(StandardCharsets.UTF_8).length;
    size += entry.getDetailType().getBytes(StandardCharsets.UTF_8).length;
```

```
    if (entry.getDetail() != null) {
        size += entry.getDetail().getBytes(StandardCharsets.UTF_8).length;
    }
    if (entry.getResources() != null) {
        for (String resource : entry.getResources()) {
            if (resource != null) {
                size += resource.getBytes(StandardCharsets.UTF_8).length;
            }
        }
    }
    return size;
}
```

> **Note**
>
> Se o tamanho da entrada for maior que 256 KB, recomendamos fazer o upload do evento em um bucket do Amazon S3 e incluindo o `Object URL` na entrada `PutEvents`.

Eventos de AWS serviços

Muitos AWS serviços geram eventos que EventBridge recebem. Quando um AWS serviço em sua conta emite um evento, ele vai para o barramento de eventos padrão da sua conta.

Entrega de eventos a partir de AWS serviços

Cada AWS serviço que gera eventos os envia EventBridge como melhor esforço ou como tentativa de entrega duradoura.

- O melhor esforço de entrega significa que o serviço tenta enviar todos os eventos para EventBridge, mas em alguns casos raros, um evento pode não ser entregue.
- Entrega durável significa que o serviço tentará entregar eventos com sucesso pelo EventBridge menos uma vez.

  EventBridge aceitará todos os eventos válidos em condições normais. Nos casos em que os eventos não puderem ser entregues devido a uma interrupção do EventBridge serviço, eles serão repetidos mais tarde pelo AWS serviço por até 24 horas.

Depois que um evento é entregue EventBridge, ele é comparado EventBridge às regras e, em seguida, segue a política de repetição e qualquer fila de mensagens sem saída especificada para o (s) destino (s) do evento.

Para obter uma lista de AWS serviços que geram eventos, consulte???.

Acessando eventos AWS de serviço via AWS CloudTrail

AWS CloudTrail é um serviço que registra automaticamente eventos, como chamadas de AWS API. Você pode criar EventBridge regras que usem as informações de CloudTrail. Para obter mais informações sobre CloudTrail, consulte O que é AWS CloudTrail? .

Todos os eventos que são entregues por CloudTrail têm `AWS API Call via CloudTrail` como valor`detail-type`.

Para registrar eventos com um `detail-type` valor de`AWS API Call via CloudTrail`, é necessária uma CloudTrail trilha com o registro ativado.

Ao usar CloudTrail com o Amazon S3, você precisa configurar CloudTrail para registrar eventos de dados. Para obter mais informações, consulte Habilitar o registro de CloudTrail eventos para buckets e objetos do S3.

Algumas ocorrências nos AWS serviços podem ser reportadas EventBridge tanto pelo próprio serviço quanto pelo CloudTrail. Por exemplo, uma chamada de API do Amazon EC2 que inicia ou interrompe uma instância gera EventBridge eventos, bem como eventos por meio dela. CloudTrail

CloudTrail permite que chamadores de API e proprietários de recursos recebam eventos em seus buckets do Amazon S3 criando trilhas e entrega eventos aos chamadores de API por meio deles. EventBridge Os proprietários de recursos, além dos chamadores de API, podem monitorar chamadas de API entre contas por meio de. EventBridge CloudTrailA integração com EventBridge fornece uma maneira conveniente de definir fluxos de trabalho automatizados baseados em regras em resposta a eventos.

Você não pode usar AWS eventos de chamada da API Put*Events maiores que 256 KB como padrões de eventos porque o tamanho máximo de qualquer solicitação Put*Events é 256 KB. Para obter mais informações sobre as chamadas de API que você pode usar, consulte serviços e integrações CloudTrail compatíveis.

Recebendo eventos de gerenciamento somente para leitura dos serviços AWS

Você pode configurar regras em seu barramento de eventos padrão ou personalizado para receber eventos de gerenciamento somente para leitura dos AWS serviços via. CloudTrail Os eventos de gerenciamento fornecem visibilidade das operações de gerenciamento que são realizadas nos recursos AWS da sua conta. Elas também são conhecidas como operações de plano de controle. Para obter mais informações, consulte Log de eventos de gerenciamento no Guia do usuário da CloudTrail .

Para cada regra nos barramentos de eventos padrão ou personalizados, é possível definir o estado da regra para controlar os tipos de eventos a serem recebidos:

- Desative a regra para que os eventos EventBridge não correspondam à regra.
- Ative a regra para que os eventos EventBridge correspondam à regra, exceto os eventos de AWS gerenciamento somente para leitura fornecidos por meio de. CloudTrail
- Ative a regra para que todos os eventos EventBridge correspondam à regra, incluindo eventos de gerenciamento somente para leitura entregues por meio de. CloudTrail

Ônibus de eventos parceiros não recebem AWS eventos.

Considere o seguinte ao decidir se deseja receber eventos de gerenciamento somente leitura:

- Certos eventos de gerenciamento somente para leitura, como AWS Key Management Service `GetKeyPolicy` and`DescribeKey`, ou IAM `GetPolicy` and `GetRole` events, ocorrem em um volume muito maior do que os eventos de mudança típicos.
- Talvez você já esteja recebendo eventos de gerenciamento somente leitura, se esses eventos não começarem com `Describe`, `Get` ou `List`. Por exemplo, eventos das seguintes AWS STS APIs são eventos de mudança, mesmo que comecem com o verbo: `Get`
  - `GetFederationToken`
  - `GetSessionToken`

  Para obter uma lista de eventos de gerenciamento somente para leitura que não aderem à convenção de `List` nomenclatura ou à `Describe` convenção de nomenclatura por AWS serviços, consulte. `Get` ???

Para criar uma regra que receba eventos de gerenciamento somente para leitura usando a CLI AWS

- Utilize o comando `put-rule` para criar ou atualizar a regra, com parâmetros para:
  - Especificar se a regra pertence ao barramento de eventos padrão ou a um barramento de eventos personalizado específico
  - Definir o estado da regra como `ENABLED_WITH_ALL_CLOUDTRAIL_MANAGEMENT_EVENTS`

```
aws events put-rule --name "ruleForManagementEvents" --event-bus-name
"default" --state "ENABLED_WITH_ALL_CLOUDTRAIL_MANAGEMENT_EVENTS"
```

> **Note**
>
> A habilitação de uma regra para eventos CloudWatch de gerenciamento é suportada somente por meio da AWS CLI e dos AWS CloudFormation modelos.

Example

O exemplo a seguir ilustra como corresponder a eventos específicos. A prática recomendada é definir uma regra dedicada para corresponder eventos específicos para clareza e facilidade de edição.

Nesse caso, a regra dedicada corresponde ao evento `AssumeRole` de gerenciamento de AWS Security Token Service.

```
{
    "source" : [ "aws.sts" ],
    "detail-type": ["AWS API Call via CloudTrail"],
    "detail" : {
        "eventName" : ["AssumeRole"]
    }
}
```

AWS serviços que geram eventos

A tabela a seguir mostra AWS os serviços que geram eventos. Escolha o nome do serviço para ver mais informações sobre como esse serviço e o EventBridge trabalho em conjunto.

Cada AWS serviço que gera eventos os envia EventBridge como melhor esforço ou como tentativa de entrega duradoura. Para ter mais informações, consulte ???.

Essa tabela inclui uma representação dos AWS serviços para os quais enviam eventos EventBridge, mas não inclui todos os serviços. Para serviços não listados que enviam eventos para EventBridge, assuma o melhor esforço de entrega.

Serviço	Tipo de tentativa
Alexa for Business	Melhor esforço
AWS Account Management	Melhor esforço
Amazon API Gateway	Melhor esforço
AWS AppConfig	Melhor esforço
Amazon AppFlow	Melhor esforço
Application Auto Scaling	Melhor esforço
AWS Perfil de custos de aplicativos	Melhor esforço
AWS Application Migration Service	Melhor esforço
Amazon Athena	Melhor esforço
AWS Backup	Melhor esforço
AWS Batch	Durável
Amazon Braket	Durável
AWS Certificate Manager	Melhor esforço
Amazon Chime	Melhor esforço
Amazon Cloud Directory	Melhor esforço
AWS CloudFormation	Durável
Amazon CloudFront	Melhor esforço

Serviço	Tipo de tentativa
AWS CloudHSM	Melhor esforço
Amazon CloudSearch	Melhor esforço
AWS CloudShell	Melhor esforço
Eventos de AWS CloudTrail	Melhor esforço
Amazon CloudWatch	Durável
Amazon CloudWatch Application Insights	Melhor esforço
Monitor de CloudWatch Internet da Amazon	Melhor esforço
CloudWatch Registros da Amazon	Melhor esforço
Amazon CloudWatch Synthetics	Melhor esforço
AWS CodeArtifact	Durável
AWS CodeBuild	Melhor esforço
AWS CodeCommit	Melhor esforço
AWS CodeDeploy	Melhor esforço
Amazon CodeGuru Profiler	Melhor esforço
AWS CodePipeline	Melhor esforço
AWS CodeStar	Melhor esforço
CodeConnections	Melhor esforço
Identidade do Amazon Cognito	Melhor esforço
Grupos de usuários do Amazon Cognito	Melhor esforço
Amazon Cognito Sync	Melhor esforço

Serviço	Tipo de tentativa
AWS Config	Melhor esforço
Amazon Connect	Melhor esforço
Amazon Connect Voice ID	Melhor esforço
AWS Control Tower	Melhor esforço
AWS Database Migration Service	Melhor esforço
AWS Data Exchange	Melhor esforço
Amazon Data Lifecycle Manager	Melhor esforço
AWS Data Pipeline	Melhor esforço
AWS DataSync	Melhor esforço
AWS Device Farm	Melhor esforço
DevOpsGuru da Amazon	Melhor esforço
AWS Direct Connect	Melhor esforço
AWS Directory Service	Melhor esforço
Amazon DynamoDB	Melhor esforço
AWS Elastic Beanstalk	Melhor esforço
Amazon Elastic Block Store	Melhor esforço
Modificações ao volume do Amazon Elastic Block Store	Melhor esforço
Amazon ElastiCache	Melhor esforço
Amazon Elastic Compute Cloud (Amazon EC2)	Melhor esforço
Amazon EC2 Auto Scaling	Melhor esforço

Serviço	Tipo de tentativa
Amazon EC2 Fleets	Melhor esforço
Interrupção da instância spot do Amazon EC2	Melhor esforço
Amazon Elastic Container Registry	Melhor esforço
Amazon Elastic Container Service	Durável
AWS Elastic Disaster Recovery	Melhor esforço
Amazon Elastic File System	Melhor esforço
Amazon Elastic Kubernetes Service	Melhor esforço
Elastic Load Balancing	Melhor esforço
Amazon Elastic MapReduce	Melhor esforço
Amazon Elastic Transcoder	Melhor esforço
AWS Elemental MediaConnect	Melhor esforço
AWS Elemental MediaConvert	Durável
AWS Elemental MediaLive	Melhor esforço
AWS Elemental MediaPackage	Melhor esforço
AWS Elemental MediaStore	Durável
Amazon EMR	Melhor esforço
Amazon EMR no EKS	Melhor esforço
Amazon EMR Serverless	Melhor esforço
Regras EventBridge programadas da Amazon	Durável
EventBridge Esquemas da Amazon	Melhor esforço

Serviço	Tipo de tentativa
AWS Fault Injection Service	Melhor esforço
Previsão	Melhor esforço
Amazon GameLift	Melhor esforço
AWS Glue	Melhor esforço
AWS Glue DataBrew	Melhor esforço
AWS Ground Station	Melhor esforço
Amazon GuardDuty	Melhor esforço
AWS Health	Melhor esforço
AWS HealthLake	Durável
AWS Identity and Access Management (IAM)	Melhor esforço
IAM Access Analyzer	Melhor esforço
Amazon Inspector Classic	Melhor esforço
Amazon Inspector	Melhor esforço
AWS IoT	Melhor esforço
AWS IoT Analytics	Durável
AWS IoT Greengrass V1	Melhor esforço
AWS IoT Greengrass V2	Melhor esforço
Amazon Interactive Video Service	Melhor esforço
Amazon Kinesis	Melhor esforço
Amazon Data Firehose	Melhor esforço

Serviço	Tipo de tentativa
AWS Key Management Service Exclusão da CMK	Durável
AWS Key Management Service Rotação CMK	Melhor esforço
AWS Key Management Service expiração do material chave importado	Melhor esforço
AWS Lambda	Melhor esforço
Amazon Location Service	Durável
Amazon Machine Learning	Melhor esforço
Amazon Macie	Melhor esforço
Amazon Managed Blockchain	Melhor esforço
AWS Managed Services	Melhor esforço
AWS Management Console Entrar	Melhor esforço
AWS Marketplace de medição	Melhor esforço
AWS Migration Hub	Melhor esforço
AWS Migration Hub Refactor Spaces	Melhor esforço
AWS Monitoramento	Melhor esforço
AWS Network Manager	Melhor esforço
OpenSearch Serviço Amazon	Melhor esforço
AWS OpsWorks	Durável
AWS OpsWorks CM	Melhor esforço
AWS Organizations	Melhor esforço

Serviço	Tipo de tentativa
Amazon Polly	Melhor esforço
AWS Private Certificate Authority	Melhor esforço
AWS Proton	Melhor esforço
Amazon QLDB	Durável
Amazon QuickSight	Melhor esforço
Amazon RDS	Melhor esforço
AWS Lixeira	Melhor esforço
Amazon Redshift	Durável
API de dados do Amazon Redshift	Melhor esforço
Amazon Redshift sem servidor	Melhor esforço
AWS Resource Access Manager	Melhor esforço
AWS Resource Groups	Melhor esforço
AWS Resource Groups Tagging API	Melhor esforço
Amazon Route 53	Melhor esforço
Amazon Route 53 Recovery Readiness	Melhor esforço
Amazon SageMaker	Melhor esforço
Savings Plans	Melhor esforço
AWS Secrets Manager	Melhor esforço
AWS Security Hub	Durável
AWS Security Token Service	Melhor esforço

Serviço	Tipo de tentativa
AWS Server Migration Service	Melhor esforço
AWS Service Catalog	Melhor esforço
AWS Signer	Durável
Amazon Simple Email Service	Melhor esforço
Amazon Simple Storage Service (Amazon S3)	Durável
Amazon S3 Glacier	Melhor esforço
Amazon S3 on Outposts	Melhor esforço
Amazon Simple Queue Service	Melhor esforço
Amazon Simple Notification Service	Melhor esforço
Amazon Simple Workflow Service	Melhor esforço
AWS Step Functions	Melhor esforço
AWS Storage Gateway	Durável
AWS Support	Melhor esforço
AWS Systems Manager	Melhor esforço
Amazon Transcribe	Melhor esforço
AWS Transfer Family	Melhor esforço
AWS Transit Gateway	Melhor esforço
Amazon Translate	Durável
AWS Trusted Advisor	Melhor esforço
AWS WAF	Melhor esforço

Serviço	Tipo de tentativa
AWS WAF Regional	Melhor esforço
AWS Well-Architected Tool	Melhor esforço
Amazon WorkDocs	Melhor esforço
Amazon WorkSpaces	Melhor esforço
AWS X-Ray	Melhor esforço

Eventos de gerenciamento gerados por AWS serviços

Em geral, as APIs que geram eventos de gerenciamento (ou somente leitura) começam com os verbos `Describe`, `Get` ou `List`. A tabela abaixo lista AWS os serviços e os eventos de gerenciamento que eles geram e que não seguem essa convenção de nomenclatura. Para obter mais informações sobre eventos de gerenciamento, consulte ???.

Eventos de gerenciamento que não começam com **`Describe`**, **`Get`** ou **`List`**

A tabela a seguir lista AWS os serviços e os eventos de gerenciamento que eles geram e que não seguem as convenções de nomenclatura típicas de começar com `DescribeGet`, ou. `List`

Serviço	Nome do evento	Tipo de evento
Alexa for Business	ResolveRoom	Chamada de API
Alexa for Business	SearchAddressBooks	Chamada de API
Alexa for Business	SearchContacts	Chamada de API
Alexa for Business	SearchDevices	Chamada de API
Alexa for Business	SearchProfiles	Chamada de API
Alexa for Business	SearchRooms	Chamada de API
Alexa for Business	SearchSkillGroups	Chamada de API

Serviço	Nome do evento	Tipo de evento
Alexa for Business	SearchUsers	Chamada de API
IAM Access Analyzer	ValidatePolicy	Chamada de API
AWS AdSpace Salas limpas	BatchGetSchema	Chamada de API
AWS Amplify Construtor de interface	ExportComponents	Chamada de API
AWS Amplify Construtor de interface	ExportForms	Chamada de API
AWS Amplify Construtor de interface	ExportThemes	Chamada de API
OpenSearch Serviço Amazon	BatchGetCollection	Chamada de API
Amazon API Gateway	ExportApi	Chamada de API
AWS AppConfig	ValidateConfiguration	Chamada de API
Amazon AppFlow	RetrieveConnectorData	Chamada de API
Amazon CloudWatch Application Insights	UpdateApplicationDashboardConfiguration	Chamada de API
Amazon Athena	BatchGetNamedQuery	Chamada de API
Amazon Athena	BatchGetPreparedStatement	Chamada de API
Amazon Athena	BatchGetQueryExecution	Chamada de API
Amazon Athena	CheckQueryCompatibility	Chamada de API
Amazon Athena	ExportNotebook	Chamada de API
AWS Auto Scaling	AreScalableTargetsRegistered	Chamada de API
AWS Auto Scaling	Teste	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS Marketplace	SearchAgreements	Chamada de API
AWS Backup	CreateLegalHold	Chamada de API
AWS Backup	ExportBackupPlanTemplate	Chamada de API
AWS Backup gateway	TestHypervisorConfiguration	Chamada de API
AWS Billing and Cost Management	AWSPaymentInstrumentGateway.Obter	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.DescribeMakePaymentPage	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.DescribePaymentsDashboard	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetAccountPreferences	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetAdvancePaySummary	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetAsoBulkDownload	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetBillingContactAddress	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetDocuments	Ação do console

Serviço	Nome do evento	Tipo de evento
AWS Billing and Cost Management	AWSPaymentPortalService.GetEligiblePaymentInstruments	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetEntitiesByIds	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetFundingDocuments	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetKybcValidationStatus	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetOneTimePasswordStatus	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentHistory	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileByArn	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileCurrencies	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfiles	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileServiceProviders	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentsDue	Ação do console

Serviço	Nome do evento	Tipo de evento
AWS Billing and Cost Management	AWSPaymentPortalService.GetRemittanceInformation	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetTaxInvoiceMetadata	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetTermsAndConditionsForProgramGroup	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetTransactionsHistory	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetUnappliedFunds	Ação do console
AWS Billing and Cost Management	AWSPaymentPortalService.GetUnpaidInvoices	Ação do console
AWS Billing and Cost Management	AWSPaymentPreferenceGateway.Obter	Ação do console
AWS Billing and Cost Management	CancelBulkDownload	Ação do console
AWS Billing and Cost Management	DownloadCommercialInvoice	Ação do console
AWS Billing and Cost Management	DownloadCsv	Ação do console
AWS Billing and Cost Management	DownloadDoc	Ação do console
AWS Billing and Cost Management	Baixar ECSV ForBillingPeriod	Ação do console

Serviço	Nome do evento	Tipo de evento
AWS Billing and Cost Management	DownloadPaymentHistory	Ação do console
AWS Billing and Cost Management	DownloadRegistrationDocument	Ação do console
AWS Billing and Cost Management	DownloadTaxInvoice	Ação do console
AWS Billing and Cost Management	FindBankRedirectPaymentInstruments	Ação do console
AWS Billing and Cost Management	Encontre o ECSV ForBillingPeriod	Ação do console
AWS Billing and Cost Management	ValidateReportDestination	Ação do console
AWS Billing and Cost Management	VerifyChinaPaymentEligibility	Ação do console
Amazon Braket	SearchCompilations	Chamada de API
Amazon Braket	SearchDevices	Chamada de API
Amazon Braket	SearchQuantumTasks	Chamada de API
Amazon Connect Cases	BatchGetField	Chamada de API
Amazon Connect Cases	SearchCases	Chamada de API
Amazon Connect Cases	SearchRelatedItems	Chamada de API
Amazon Chime	RetrieveDataExports	Chamada de API
Amazon Chime	SearchChannels	Chamada de API
Identidade do SDK do Amazon Chime	DeleteProfile	Evento de serviço

Serviço	Nome do evento	Tipo de evento
Identidade do SDK do Amazon Chime	DeleteWorkTalkAccount	Evento de serviço
AWS Salas limpas	BatchGetSchema	Chamada de API
Amazon Cloud Directory	BatchRead	Chamada de API
Amazon Cloud Directory	LookupPolicy	Chamada de API
AWS CloudFormation	DetectStackDrift	Chamada de API
AWS CloudFormation	DetectStackResourceDrift	Chamada de API
AWS CloudFormation	DetectStackSetDrift	Chamada de API
AWS CloudFormation	EstimateTemplateCost	Chamada de API
AWS CloudFormation	ValidateTemplate	Chamada de API
AWS CloudShell	RedeemCode	Chamada de API
AWS CloudTrail	LookupEvents	Chamada de API
AWS CodeArtifact	ReadFromRepository	Chamada de API
AWS CodeArtifact	SearchPackages	Chamada de API
AWS CodeArtifact	VerifyResourcesExistForTagris	Chamada de API
AWS CodeBuild	BatchGetBuildBatches	Chamada de API
AWS CodeBuild	BatchGetBuilds	Chamada de API
AWS CodeBuild	BatchGetProjects	Chamada de API
AWS CodeBuild	BatchGetReportGroups	Chamada de API
AWS CodeBuild	BatchGetReports	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS CodeBuild	BatchPutCodeCoverages	Chamada de API
AWS CodeBuild	BatchPutTestCases	Chamada de API
AWS CodeBuild	RequestBadge	Evento de serviço
AWS CodeCommit	BatchDescribeMergeConflicts	Chamada de API
AWS CodeCommit	BatchGetCommits	Chamada de API
AWS CodeCommit	BatchGetPullRequests	Chamada de API
AWS CodeCommit	BatchGetRepositories	Chamada de API
AWS CodeCommit	EvaluatePullRequestApprovalRules	Chamada de API
AWS CodeCommit	GitPull	Chamada de API
AWS CodeDeploy	BatchGetApplicationRevisions	Chamada de API
AWS CodeDeploy	BatchGetApplications	Chamada de API
AWS CodeDeploy	BatchGetDeploymentGroups	Chamada de API
AWS CodeDeploy	BatchGetDeploymentInstances	Chamada de API
AWS CodeDeploy	BatchGetDeployments	Chamada de API
AWS CodeDeploy	BatchGetDeploymentTargets	Chamada de API
AWS CodeDeploy	BatchGetOnPremisesInstances	Chamada de API
Amazon CodeGuru Profiler	BatchGetFrameMetricData	Chamada de API
Amazon CodeGuru Profiler	SubmitFeedback	Chamada de API
AWS CodePipeline	PollForJobs	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS CodePipeline	PollForThirdPartyJobs	Chamada de API
CodeConnections	StartAppRegistrationHandshake	Chamada de API
CodeConnections	Iniciar para AuthHandshake	Chamada de API
CodeConnections	ValidateHostWebhook	Chamada de API
Amazon CodeWhisperer	CreateCodeScan	Chamada de API
Amazon CodeWhisperer	CreateProfile	Chamada de API
Amazon CodeWhisperer	CreateUploadUrl	Chamada de API
Amazon CodeWhisperer	GenerateRecommendations	Chamada de API
Amazon CodeWhisperer	UpdateProfile	Chamada de API
Identidade do Amazon Cognito	LookupDeveloperIdentity	Chamada de API
Grupos de usuários do Amazon Cognito	AdminGetDevice	Chamada de API
Grupos de usuários do Amazon Cognito	AdminGetUser	Chamada de API
Grupos de usuários do Amazon Cognito	AdminListDevices	Chamada de API
Grupos de usuários do Amazon Cognito	AdminListGroupsForUser	Chamada de API
Grupos de usuários do Amazon Cognito	AdminListUserAuthEvents	Chamada de API
Grupos de usuários do Amazon Cognito	Beta_Authorize_GET	Evento de serviço

Serviço	Nome do evento	Tipo de evento
Grupos de usuários do Amazon Cognito	Confirm_GET	Evento de serviço
Grupos de usuários do Amazon Cognito	ConfirmForgotPassword_OBTER	Evento de serviço
Grupos de usuários do Amazon Cognito	Error_GET	Evento de serviço
Grupos de usuários do Amazon Cognito	ForgotPassword_OBTER	Evento de serviço
Grupos de usuários do Amazon Cognito	IntrospectToken	Chamada de API
Grupos de usuários do Amazon Cognito	Login_Error_POST	Evento de serviço
Grupos de usuários do Amazon Cognito	Login_GET	Evento de serviço
Grupos de usuários do Amazon Cognito	Mfa_GET	Evento de serviço
Grupos de usuários do Amazon Cognito	MfaOption_OBTER	Evento de serviço
Grupos de usuários do Amazon Cognito	ResetPassword_OBTER	Evento de serviço
Grupos de usuários do Amazon Cognito	Signup_GET	Evento de serviço
Grupos de usuários do Amazon Cognito	UserInfo_OBTER	Evento de serviço
Grupos de usuários do Amazon Cognito	UserInfo_PUBLICAR	Evento de serviço

Serviço	Nome do evento	Tipo de evento
Amazon Cognito Sync	BulkPublish	Chamada de API
Amazon Comprehend	BatchContainsPiiEntities	Chamada de API
Amazon Comprehend	BatchDetectDominantLanguage	Chamada de API
Amazon Comprehend	BatchDetectEntities	Chamada de API
Amazon Comprehend	BatchDetectKeyPhrases	Chamada de API
Amazon Comprehend	BatchDetectPiiEntities	Chamada de API
Amazon Comprehend	BatchDetectSentiment	Chamada de API
Amazon Comprehend	BatchDetectSyntax	Chamada de API
Amazon Comprehend	BatchDetectTargetedSentiment	Chamada de API
Amazon Comprehend	ClassifyDocument	Chamada de API
Amazon Comprehend	ContainsPiiEntities	Chamada de API
Amazon Comprehend	DetectDominantLanguage	Chamada de API
Amazon Comprehend	DetectEntities	Chamada de API
Amazon Comprehend	DetectKeyPhrases	Chamada de API
Amazon Comprehend	DetectPiiEntities	Chamada de API
Amazon Comprehend	DetectSentiment	Chamada de API
Amazon Comprehend	DetectSyntax	Chamada de API
Amazon Comprehend	DetectTargetedSentiment	Chamada de API
Amazon Comprehend	DetectToxicContent	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS Compute Optimizer	ExportAutoScalingGroupRecommendations	Chamada de API
AWS Compute Optimizer	Exportar EBS VolumeRecommendations	Chamada de API
AWS Compute Optimizer	Exportar EC InstanceRecommendations	Chamada de API
AWS Compute Optimizer	Exportar ECS ServiceRecommendations	Chamada de API
AWS Compute Optimizer	ExportLambdaFunctionRecommendations	Chamada de API
AWS Compute Optimizer	Exportar RDS InstanceRecommendations	Chamada de API
AWS Config	BatchGetAggregateResourceConfig	Chamada de API
AWS Config	BatchGetResourceConfig	Chamada de API
AWS Config	SelectAggregateResourceConfig	Chamada de API
AWS Config	SelectResourceConfig	Chamada de API
Amazon Connect	AdminGetEmergencyAccessToken	Chamada de API
Amazon Connect	SearchQueues	Chamada de API
Amazon Connect	SearchRoutingProfiles	Chamada de API
Amazon Connect	SearchSecurityProfiles	Chamada de API
Amazon Connect	SearchUsers	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS Glue DataBrew	SendProjectSessionAction	Chamada de API
AWS Data Pipeline	EvaluateExpression	Chamada de API
AWS Data Pipeline	QueryObjects	Chamada de API
AWS Data Pipeline	ValidatePipelineDefinition	Chamada de API
AWS DataSync	VerifyResourcesExistForTagris	Chamada de API
AWS DeepLens	BatchGetDevice	Chamada de API
AWS DeepLens	BatchGetModel	Chamada de API
AWS DeepLens	BatchGetProject	Chamada de API
AWS DeepLens	CreateDeviceCertificates	Chamada de API
AWS DeepRacer	AdminGetAccountConfig	Chamada de API
AWS DeepRacer	AdminListAssociatedUsers	Chamada de API
AWS DeepRacer	TestRewardFunction	Chamada de API
AWS DeepRacer	VerifyResourcesExistForTagris	Chamada de API
Amazon Detective	BatchGetGraphMemberDatasources	Chamada de API
Amazon Detective	BatchGetMembershipDatasources	Chamada de API
Amazon Detective	SearchGraph	Chamada de API
DevOpsGuru da Amazon	SearchInsights	Chamada de API
DevOpsGuru da Amazon	SearchOrganizationInsights	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS Database Migration Service	BatchStartRecommendations	Chamada de API
AWS Database Migration Service	ModifyRecommendation	Chamada de API
AWS Database Migration Service	StartRecommendations	Chamada de API
AWS Database Migration Service	VerifyResourcesExistForTagris	Chamada de API
AWS Directory Service	VerifyTrust	Chamada de API
Amazon Elastic Compute Cloud	ConfirmProductInstance	Chamada de API
Amazon Elastic Compute Cloud	ReportInstanceStatus	Chamada de API
Amazon Elastic Container Registry	BatchCheckLayerAvailability	Chamada de API
Amazon Elastic Container Registry	BatchGetImage	Chamada de API
Amazon Elastic Container Registry	BatchGetImageReferrer	Chamada de API
Amazon Elastic Container Registry	BatchGetRepository ScanningConfiguration	Chamada de API
Amazon Elastic Container Registry	DryRunEvent	Evento de serviço
Amazon Elastic Container Registry	PolicyExecutionEvent	Evento de serviço

Serviço	Nome do evento	Tipo de evento
Amazon Elastic Container Registry Public	BatchCheckLayerAvailability	Chamada de API
Amazon Elastic Container Service	DiscoverPollEndpoint	Chamada de API
Amazon Elastic Container Service	FindSubfleetRoute	Chamada de API
Amazon Elastic Container Service	ValidateResources	Chamada de API
Amazon Elastic Container Service	VerifyTaskSetsExist	Chamada de API
Amazon Elastic Kubernetes Service	AccessKubernetesApi	Chamada de API
AWS Elastic Beanstalk	CheckDNSAvailability	Chamada de API
AWS Elastic Beanstalk	RequestEnvironmentInfo	Chamada de API
AWS Elastic Beanstalk	RetrieveEnvironmentInfo	Chamada de API
AWS Elastic Beanstalk	ValidateConfigurationSettings	Chamada de API
Amazon Elastic File System	NewClientConnection	Evento de serviço
Amazon Elastic File System	UpdateClientConnection	Evento de serviço
Amazon Elastic Transcoder	ReadJob	Chamada de API
Amazon Elastic Transcoder	ReadPipeline	Chamada de API
Amazon Elastic Transcoder	ReadPreset	Chamada de API
Amazon EventBridge	TestEventPattern	Chamada de API
Amazon EventBridge	TestScheduleExpression	Chamada de API

Serviço	Nome do evento	Tipo de evento
Amazon FinSpace API	BatchListCatalogNodesByDataset	Chamada de API
Amazon FinSpace API	BatchListNodesByDataset	Chamada de API
Amazon FinSpace API	BatchValidateAccess	Chamada de API
Amazon FinSpace API	CreateAuditRecordsQuery	Chamada de API
Amazon FinSpace API	SearchDatasets	Chamada de API
Amazon FinSpace API	SearchDatasetsV	Chamada de API
Amazon FinSpace API	ValidateIdToken	Chamada de API
AWS Firewall Manager	DisassociateAdminAccount	Chamada de API
Amazon Forecast	InvokeForecastEndpoint	Chamada de API
Amazon Forecast	QueryFeature	Chamada de API
Amazon Forecast	QueryForecast	Chamada de API
Amazon Forecast	QueryWhatIfForecast	Chamada de API
Amazon Forecast	VerifyResourcesExistForTagris	Chamada de API
Amazon Fraud Detector	BatchGetVariable	Chamada de API
Amazon Fraud Detector	VerifyResourcesExistForTagris	Chamada de API
FreeRTOS	VerifyEmailAddress	Chamada de API
Amazon GameLift	RequestUploadCredentials	Chamada de API
Amazon GameLift	ResolveAlias	Chamada de API
Amazon GameLift	SearchGameSessions	Chamada de API

Serviço	Nome do evento	Tipo de evento
Amazon GameLift	ValidateMatchmakingRuleSet	Chamada de API
Amazon GameSparks	ExportSnapshot	Chamada de API
Amazon Location Service	BatchGetDevicePosition	Chamada de API
Amazon Location Service	CalculateRoute	Chamada de API
Amazon Location Service	CalculateRouteMatrix	Chamada de API
Amazon Location Service	SearchPlaceIndexForPosition	Chamada de API
Amazon Location Service	SearchPlaceIndexForSuggestions	Chamada de API
Amazon Location Service	SearchPlaceIndexForText	Chamada de API
Amazon S3 Glacier	InitiateJob	Chamada de API
AWS Glue	BatchGetBlueprints	Chamada de API
AWS Glue	BatchGetColumnStatisticsForTable	Chamada de API
AWS Glue	BatchGetCrawlers	Chamada de API
AWS Glue	BatchGetCustomEntityTypes	Chamada de API
AWS Glue	BatchGetDataQualityResult	Chamada de API
AWS Glue	BatchGetDevEndpoints	Chamada de API
AWS Glue	BatchGetJobs	Chamada de API
AWS Glue	BatchGetTransformação ML	Chamada de API
AWS Glue	BatchGetPartition	Chamada de API
AWS Glue	BatchGetTriggers	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS Glue	BatchGetWorkflows	Chamada de API
AWS Glue	QueryJobRuns	Chamada de API
AWS Glue	QueryJobRunsAggregated	Chamada de API
AWS Glue	QueryJobs	Chamada de API
AWS Glue	QuerySchemaVersion Metadata	Chamada de API
AWS Glue	SearchTables	Chamada de API
AWS HealthLake	ReadResource	Chamada de API
AWS HealthLake	SearchWithGet	Chamada de API
AWS HealthLake	SearchWithPost	Chamada de API
AWS Identity and Access Management	GenerateCredentialReport	Chamada de API
AWS Identity and Access Management	GenerateOrganizationsAccess Report	Chamada de API
AWS Identity and Access Management	GenerateServiceLas tAccessedDetails	Chamada de API
AWS Identity and Access Management	SimulateCustomPolicy	Chamada de API
AWS Identity and Access Management	SimulatePrincipalPolicy	Chamada de API
AWS Loja de identidades	IsMemberInGroups	Chamada de API
AWS Autenticação do Identity Store	BatchGetSession	Chamada de API

Serviço	Nome do evento	Tipo de evento
Amazon Inspector Classic	PreviewAgents	Chamada de API
Amazon Inspector Classic	BatchGetAccountStatus	Chamada de API
Amazon Inspector Classic	BatchGetFreeTrialInfo	Chamada de API
Amazon Inspector Classic	BatchGetMember	Chamada de API
Faturamento da AWS	ValidateDocumentDeliveryS3LocationInfo	Chamada de API
AWS IoT	SearchIndex	Chamada de API
AWS IoT	TestAuthorization	Chamada de API
AWS IoT	TestInvokeAuthorizer	Chamada de API
AWS IoT	ValidateSecurityProfileBehaviors	Chamada de API
AWS IoT Analytics	SampleChannelData	Chamada de API
AWS IoT SiteWise	GatewaysVerifyResourcesExistForTagrisInternal	Chamada de API
AWS IoT Things Graph	SearchEntities	Chamada de API
AWS IoT Things Graph	SearchFlowExecutions	Chamada de API
AWS IoT Things Graph	SearchFlowTemplates	Chamada de API
AWS IoT Things Graph	SearchSystemInstances	Chamada de API
AWS IoT Things Graph	SearchSystemTemplates	Chamada de API
AWS IoT Things Graph	SearchThings	Chamada de API
AWS IoT TwinMaker	ExecuteQuery	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS IoT Wireless	CreateNetworkAnalyzerConfiguration	Chamada de API
AWS IoT Wireless	DeleteNetworkAnalyzerConfiguration	Chamada de API
AWS IoT Wireless	DeregisterWirelessDevice	Chamada de API
Amazon Interactive Video Service	BatchGetChannel	Chamada de API
Amazon Interactive Video Service	BatchGetStreamKey	Chamada de API
Amazon Kendra	BatchGetDocumentStatus	Chamada de API
Amazon Kendra	Consulta	Chamada de API
Amazon Managed Service for Apache Flink	DiscoverInputSchema	Chamada de API
AWS Key Management Service	Decrypt	Chamada de API
AWS Key Management Service	Encrypt	Chamada de API
AWS Key Management Service	GenerateDataKey	Chamada de API
AWS Key Management Service	GenerateDataKeyPair	Chamada de API
AWS Key Management Service	GenerateDataKeyPairWithoutPlaintext	Chamada de API
AWS Key Management Service	GenerateDataKeyWithoutPlaintext	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS Key Management Service	GenerateMac	Chamada de API
AWS Key Management Service	GenerateRandom	Chamada de API
AWS Key Management Service	ReEncrypt	Chamada de API
AWS Key Management Service	Sign	Chamada de API
AWS Key Management Service	Verificar	Chamada de API
AWS Key Management Service	VerifyMac	Chamada de API
AWS Lake Formation	SearchDatabasesByEtiquetas LF	Chamada de API
AWS Lake Formation	SearchTablesByEtiquetas LF	Chamada de API
AWS Lake Formation	StartQueryPlanning	Chamada de API
Amazon Lex	BatchCreateCustomVocabularyItem	Chamada de API
Amazon Lex	BatchDeleteCustomVocabularyItem	Chamada de API
Amazon Lex	BatchUpdateCustomVocabularyItem	Chamada de API
Amazon Lex	DeleteCustomVocabulary	Chamada de API
Amazon Lex	SearchAssociatedTranscripts	Chamada de API

Serviço	Nome do evento	Tipo de evento
Amazon Lightsail	Criar GUI SessionAccessDetails	Chamada de API
Amazon Lightsail	DownloadDefaultKeyPair	Chamada de API
Amazon Lightsail	IsVpcPeered	Chamada de API
CloudWatch Registros da Amazon	FilterLogEvents	Chamada de API
Amazon Macie	BatchGetCustomDataIdentifiers	Chamada de API
Amazon Macie	UpdateFindingsFilter	Chamada de API
AWS Elemental MediaConnect	ManagedDescribeFlow	Chamada de API
AWS Elemental MediaConnect	PrivateDescribeFlowMeta	Chamada de API
AWS Application Migration Service	OperationalDescribeJobLogItems	Chamada de API
AWS Application Migration Service	OperationalDescribeJobs	Chamada de API
AWS Application Migration Service	OperationalDescribeReplicationConfigurationTemplates	Chamada de API
AWS Application Migration Service	OperationalDescribeSourceServer	Chamada de API
AWS Application Migration Service	OperationalGetLaunchConfiguration	Chamada de API
AWS Application Migration Service	OperationalListSourceServers	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS Application Migration Service	VerifyClientRoleForMgn	Chamada de API
AWS HealthOmics	VerifyResourceExists	Chamada de API
AWS HealthOmics	VerifyResourcesExistForTagris	Chamada de API
Amazon Polly	SynthesizeLongSpeech	Chamada de API
Amazon Polly	SynthesizeSpeech	Chamada de API
Amazon Polly	SynthesizeSpeechGet	Chamada de API
AWS serviço que fornece redes privadas gerenciadas	Ping	Chamada de API
AWS Proton	DeleteEnvironmentTemplateVersion	Chamada de API
AWS Proton	DeleteServiceTemplateVersion	Chamada de API
Amazon QLDB	ShowCatalog	Chamada de API
Amazon QuickSight	GenerateEmbedUrlForAnonymousUser	Chamada de API
Amazon QuickSight	GenerateEmbedUrlForRegisteredUser	Chamada de API
Amazon QuickSight	QueryDatabase	Evento de serviço
Amazon QuickSight	SearchAnalyses	Chamada de API
Amazon QuickSight	SearchDashboards	Chamada de API
Amazon QuickSight	SearchDataSets	Chamada de API

Serviço	Nome do evento	Tipo de evento
Amazon QuickSight	SearchDataSources	Chamada de API
Amazon QuickSight	SearchFolders	Chamada de API
Amazon QuickSight	SearchGroups	Chamada de API
Amazon QuickSight	SearchUsers	Chamada de API
Amazon Relational Database Service	DownloadCompleteDB LogFile	Chamada de API
Amazon Relational Database Service	Baixar DB LogFilePortion	Chamada de API
Amazon Rekognition	CompareFaces	Chamada de API
Amazon Rekognition	DetectCustomLabels	Chamada de API
Amazon Rekognition	DetectFaces	Chamada de API
Amazon Rekognition	DetectLabels	Chamada de API
Amazon Rekognition	DetectModerationLabels	Chamada de API
Amazon Rekognition	DetectProtectiveEquipment	Chamada de API
Amazon Rekognition	DetectText	Chamada de API
Amazon Rekognition	RecognizeCelebrities	Chamada de API
Amazon Rekognition	SearchFaces	Chamada de API
Amazon Rekognition	SearchFacesByImage	Chamada de API
Amazon Rekognition	SearchUsers	Chamada de API
Amazon Rekognition	SearchUsersByImage	Chamada de API
Explorador de recursos da AWS	BatchGetView	Chamada de API

Serviço	Nome do evento	Tipo de evento
Explorador de recursos da AWS	Pesquisar	Chamada de API
AWS Resource Groups	SearchResources	Chamada de API
AWS Resource Groups	ValidateResourceSharing	Chamada de API
AWS RoboMaker	BatchDescribeSimulationJob	Chamada de API
Amazon Route 53	TestDNSAnswer	Chamada de API
Domínios do Amazon Route 53	checkAvailabilities	Chamada de API
Domínios do Amazon Route 53	CheckDomainAvailability	Chamada de API
Domínios do Amazon Route 53	checkDomainTransferability	Chamada de API
Domínios do Amazon Route 53	CheckDomainTransferability	Chamada de API
Domínios do Amazon Route 53	isEmailReachable	Chamada de API
Domínios do Amazon Route 53	searchDomains	Chamada de API
Domínios do Amazon Route 53	sendVerificationMessage	Chamada de API
Domínios do Amazon Route 53	ViewBilling	Chamada de API
Domínios do Amazon Route 53	viewBilling	Chamada de API

Serviço	Nome do evento	Tipo de evento
Amazon CloudWatch RUM	BatchGetRumMetricDefinitions	Chamada de API
Amazon Simple Storage Service	echo	Chamada de API
Amazon Simple Storage Service	GenerateInventory	Evento de serviço
Amazon SageMaker	BatchDescribeModelPackage	Chamada de API
Amazon SageMaker	DeleteModelCard	Chamada de API
Amazon SageMaker	QueryLineage	Chamada de API
Amazon SageMaker	RenderUiTemplate	Chamada de API
Amazon SageMaker	Pesquisar	Chamada de API
EventBridge Esquemas da Amazon	ExportSchema	Chamada de API
EventBridge Esquemas da Amazon	SearchSchemas	Chamada de API
Amazon SimpleDB	DomainMetadata	Chamada de API
AWS Secrets Manager	ValidateResourcePolicy	Chamada de API
AWS Service Catalog	ScanProvisionedProducts	Chamada de API
AWS Service Catalog	SearchProducts	Chamada de API
AWS Service Catalog	SearchProductsAsAdmin	Chamada de API
AWS Service Catalog	SearchProvisionedProducts	Chamada de API
Amazon SES	BatchGetMetricData	Chamada de API
Amazon SES	TestRenderEmailTemplate	Chamada de API

Serviço	Nome do evento	Tipo de evento
Amazon SES	TestRenderTemplate	Chamada de API
Amazon Simple Notification Service	CheckIfPhoneNumber IsOptedOut	Chamada de API
AWS SQL Workbench	BatchGetNotebookCell	Chamada de API
AWS SQL Workbench	ExportNotebook	Chamada de API
Amazon EC2 Systems Manager	ExecuteApi	Chamada de API
AWS Systems Manager Incident Manager	DeleteContactChannel	Chamada de API
AWS IAM Identity Center	IsMemberInGroup	Chamada de API
AWS IAM Identity Center	SearchGroups	Chamada de API
AWS IAM Identity Center	SearchUsers	Chamada de API
AWS STS	AssumeRole	Chamada de API
AWS STS	AssumeRoleWithSAML	Chamada de API
AWS STS	AssumeRoleWithWebIdentity	Chamada de API
AWS STS	DecodeAuthorizationMessage	Chamada de API
AWS Configurações fiscais	BatchGetTaxExemptions	Chamada de API
AWS WAFV2	CheckCapacity	Chamada de API
AWS WAFV2	GenerateMobileSdkR eleaseUrl	Chamada de API
AWS Well-Architected Tool	ExportLens	Chamada de API
AWS Well-Architected Tool	TagResource	Chamada de API

Serviço	Nome do evento	Tipo de evento
AWS Well-Architected Tool	UntagResource	Chamada de API
AWS Well-Architected Tool	UpdateGlobalSettings	Chamada de API
Amazon Connect Wisdom	QueryAssistant	Chamada de API
Amazon Connect Wisdom	SearchContent	Chamada de API
Amazon Connect Wisdom	SearchSessions	Chamada de API
Amazon WorkDocs	AbortDocumentVersionUpload	Chamada de API
Amazon WorkDocs	AddUsersToGroup	Chamada de API
Amazon WorkDocs	BatchGetUsers	Chamada de API
Amazon WorkDocs	CheckAlias	Chamada de API
Amazon WorkDocs	CompleteDocumentVersionUpload	Chamada de API
Amazon WorkDocs	CreateAnnotation	Chamada de API
Amazon WorkDocs	CreateComment	Chamada de API
Amazon WorkDocs	CreateFeedbackRequest	Chamada de API
Amazon WorkDocs	CreateFolder	Chamada de API
Amazon WorkDocs	CreateGroup	Chamada de API
Amazon WorkDocs	CreateShare	Chamada de API
Amazon WorkDocs	CreateUser	Chamada de API
Amazon WorkDocs	DeleteAnnotation	Chamada de API
Amazon WorkDocs	DeleteComment	Chamada de API
Amazon WorkDocs	DeleteDocument	Chamada de API

Serviço	Nome do evento	Tipo de evento
Amazon WorkDocs	DeleteFeedbackRequest	Chamada de API
Amazon WorkDocs	DeleteFolder	Chamada de API
Amazon WorkDocs	DeleteFolderContents	Chamada de API
Amazon WorkDocs	DeleteGroup	Chamada de API
Amazon WorkDocs	DeleteOrganizationShare	Chamada de API
Amazon WorkDocs	DeleteUser	Chamada de API
Amazon WorkDocs	DownloadDocumentVersion	Chamada de API
Amazon WorkDocs	DownloadDocumentVersionUnderlays	Chamada de API
Amazon WorkDocs	InitiateDocumentVersionUpload	Chamada de API
Amazon WorkDocs	LogoutUser	Chamada de API
Amazon WorkDocs	PaginatedOrganizationActivity	Chamada de API
Amazon WorkDocs	PublishAnnotations	Chamada de API
Amazon WorkDocs	PublishComments	Chamada de API
Amazon WorkDocs	RestoreDocument	Chamada de API
Amazon WorkDocs	RestoreFolder	Chamada de API
Amazon WorkDocs	SearchGroups	Chamada de API
Amazon WorkDocs	SearchOrganizationUsers	Chamada de API
Amazon WorkDocs	TransferUserResources	Chamada de API
Amazon WorkDocs	UpdateAnnotation	Chamada de API

Serviço	Nome do evento	Tipo de evento
Amazon WorkDocs	UpdateComment	Chamada de API
Amazon WorkDocs	UpdateDocument	Chamada de API
Amazon WorkDocs	UpdateDocumentVersion	Chamada de API
Amazon WorkDocs	UpdateFolder	Chamada de API
Amazon WorkDocs	UpdateGroup	Chamada de API
Amazon WorkDocs	UpdateOrganization	Chamada de API
Amazon WorkDocs	UpdateUser	Chamada de API
Amazon WorkMail	AssumeImpersonationRole	Chamada de API
Amazon WorkMail	QueryDnsRecords	Chamada de API
Amazon WorkMail	SearchMembers	Chamada de API
Amazon WorkMail	TestAvailabilityConfiguration	Chamada de API
Amazon WorkMail	TestInboundMailFlowRules	Chamada de API
Amazon WorkMail	TestOutboundMailFlowRules	Chamada de API

EventBridge referência de detalhes de eventos

EventBridge ela própria emite os seguintes eventos. Esses eventos são enviados automaticamente para o barramento de eventos padrão, como acontece com qualquer outro AWS serviço.

Para obter as definições dos campos de metadados incluídos em todos os eventos, consultethe section called “Referência de estrutura de eventos”.

Tópicos

- Evento agendado
- Esquema criado
- Versão do esquema criada

Evento agendado

Abaixo estão os campos de detalhes do `Scheduled Event` evento.

Os `detail-type` campos `source` e estão incluídos porque contêm valores específicos para EventBridge eventos. Para obter as definições dos outros campos de metadados incluídos em todos os eventos, consultethe section called “Referência de estrutura de eventos”.

```
{
  . . .,
  "detail-type": "Scheduled Event",
  "source": "aws.events",
  . . .,
  "detail": {}
}
```

`detail-type`

Identifica o tipo de evento.

Para esse evento, esse valor é`Scheduled Event`.

Obrigatório: Sim

`source`

Identifica o serviço que gerou o evento. Para EventBridge eventos, esse valor é`aws.events`.

Obrigatório: Sim

`detail`

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Obrigatório: Sim

Não há campos obrigatórios neste objeto para `Scheduled Event` eventos.

Example Exemplo de evento programado

```
{
  "version": "0",
```

```
  "id": "89d1a02d-5ec7-412e-82f5-13505f849b41",
  "detail-type": "Scheduled Event",
  "source": "aws.events",
  "account": "123456789012",
  "time": "2016-12-30T18:44:49Z",
  "region": "us-east-1",
  "resources": ["arn:aws:events:us-east-1:123456789012:rule/SampleRule"],
  "detail": {}
}
```

Esquema criado

Abaixo estão os campos de detalhes do `Schema Created` evento.

Quando um esquema é criado, EventBridge envia um `Schema Version Created` evento `Schema Created` e um.

Os `detail-type` campos `source` e estão incluídos porque contêm valores específicos para EventBridge eventos. Para obter as definições dos outros campos de metadados incluídos em todos os eventos, consultethe section called “Referência de estrutura de eventos”.

```
{
  . . .,
  "detail-type": "Schema Created",
  "source": "aws.schemas",
  . . .,
  "detail": {
    "SchemaName" : "String",
    "SchemaType" : "String",
    "RegistryName" : "String",
    "CreationDate" : "DateTime",
    "Version" : "Number"
  }
}
```

`detail-type`

Identifica o tipo de evento.

Para esse evento, esse valor é`Schema Created`.

Obrigatório: Sim

`source`

Identifica o serviço que gerou o evento. Para EventBridge eventos, esse valor é`aws.schemas`.

Obrigatório: Sim

`detail`

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Obrigatório: Sim

Para esse evento, esses dados incluem:

`SchemaName`

O nome do esquema.

Obrigatório: Sim

`SchemaType`

O tipo de esquema.

Valores válidos: `OpenApi3` | `JSONSchemaDraft4`

Obrigatório: Sim

`RegistryName`

O nome do registro que contém o esquema.

Obrigatório: Sim

`CreationDate`

A data em que o esquema foi criado.

Obrigatório: Sim

`Version`

A versão do esquema.

Para `Schema Created` eventos, esse valor sempre será`1`.

Obrigatório: Sim

Example Exemplo de esquema criado: evento

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Schema Created",
  "source": "aws.schemas",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "resources": ["arn:aws:schemas:us-east-1::schema/myRegistry/mySchema"],
  "detail": {
    "SchemaName": "mySchema",
    "SchemaType": "OpenApi3",
    "RegistryName": "myRegistry",
    "CreationDate": "2019-11-29T20:08:55Z",
    "Version": "1"
  }
}
```

Versão do esquema criada

Abaixo estão os campos de detalhes do `Schema Version Created` evento.

Quando um esquema é criado, EventBridge envia um `Schema Version Created` evento `Schema Created` e um.

Os `detail-type` campos `source` e estão incluídos porque contêm valores específicos para EventBridge eventos. Para obter as definições dos outros campos de metadados incluídos em todos os eventos, consultethe section called “Referência de estrutura de eventos”.

```
{
  . . .,
  "detail-type": "Schema Version Created",
  "source": "aws.schemas",
  . . .,
  "detail": {
    "SchemaName" : "String",
    "SchemaType" : "String",
    "RegistryName" : "String",
    "CreationDate" : "DateTime",
    "Version" : "Number"
  }
```

```
}
```

`detail-type`

Identifica o tipo de evento.

Para esse evento, esse valor é`Schema Version Created`.

Obrigatório: Sim

`source`

Identifica o serviço que gerou o evento. Para EventBridge eventos, esse valor é`aws.schemas`.

Obrigatório: Sim

`detail`

Um objeto JSON contém informações sobre o evento. O serviço que gera o evento determina o conteúdo desse campo.

Obrigatório: Sim

Para esse evento, esses dados incluem:

`SchemaName`

O nome do esquema.

Obrigatório: Sim

`SchemaType`

O tipo de esquema.

Valores válidos: `OpenApi3` | `JSONSchemaDraft4`

Obrigatório: Sim

`RegistryName`

O nome do registro que contém o esquema.

Obrigatório: Sim

`CreationDate`

A data em que a versão do esquema foi criada.

Obrigatório: Sim

`Version`

A versão do esquema.

Obrigatório: Sim

Example Exemplo de evento criado pela versão do esquema

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Schema Version Created",
  "source": "aws.schemas",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "resources": ["arn:aws:schemas:us-east-1::schema/myRegistry/mySchema"],
  "detail": {
    "SchemaName": "mySchema",
    "SchemaType": "OpenApi3",
    "RegistryName": "myRegistry",
    "CreationDate": "2019-11-29T20:08:55Z",
    "Version": "5"
  }
}
```

Recebendo eventos de um parceiro SaaS com a Amazon EventBridge

Para receber eventos de aplicações e serviços de parceiros de SaaS, é necessário ter uma origem de evento de parceiro oferecida pelo parceiro. Depois, você pode criar um barramento de eventos de parceiros e associá-lo à origem do evento do parceiro correspondente.

O vídeo a seguir aborda as integrações de SaaS com EventBridge: Parceiros de software como serviço (SaaS)

Tópicos

- Integrações compatíveis de parceiros de SaaS

- Configurando a Amazon EventBridge para receber eventos de uma integração SaaS
- Como criar uma regra que corresponda a eventos de parceiros de SaaS
- Recebendo eventos usando URLs de AWS Lambda funções
- Como receber eventos do Salesforce

Integrações compatíveis de parceiros de SaaS

EventBridge oferece suporte às seguintes integrações de parceiros SaaS:

- Adobe
- Auth0
- Blitline
- BUIDLHub
- Buildkite
- CleverTap
- Datadog
- Epsagon
- Freshworks
- Genesys
- GS2
- Karte
- Kloudless
- Mackerel
- MongoDB
- New Relic
- OneLogin
- Opsgenie
- PagerDuty
- Payshield
- SaaSus Platform
- SailPoint
- Saviynt

- Segment
- Shopify
- SignalFx
- Site24x7
- Stax
- Stripe
- SugarCRM
- SugarCRM
- Symantec
- Thundra
- TriggerMesh
- Whispir
- Zendesk
- Amazon Seller Partner API

As origens de eventos de parceiros estão disponíveis nas regiões a seguir.

Código	Nome
`us-east-1`	Leste dos EUA (Norte da Virgínia)
`us-east-2`	Leste dos EUA (Ohio)
`us-west-1`	Oeste dos EUA (N. da Califórnia)
`us-west-2`	Oeste dos EUA (Oregon)
`ca-central-1`	Canadá (Central)
`eu-central-1`	Europa (Frankfurt)
`eu-central-2`	Europa (Zurique)
`eu-west-1`	Europa (Irlanda)
`eu-west-2`	Europa (Londres)

Código	Nome
eu-west-3	Europa (Paris)
eu-north-1	Europa (Estocolmo)
eu-south-1	Europa (Milão)
eu-south-2	Europa (Espanha)
af-south-1	África (Cidade do Cabo)
ap-south-1	Ásia-Pacífico (Mumbai)
ap-south-2	Ásia-Pacífico (Hyderabad)
ap-east-1	Ásia-Pacífico (Hong Kong)
ap-northeast-1	Ásia-Pacífico (Tóquio)
ap-northeast-2	Ásia-Pacífico (Seul)
ap-northeast-3	Asia Pacific (Osaka)
ap-southeast-1	Ásia-Pacífico (Singapura)
ap-southeast-2	Ásia-Pacífico (Sydney)
ap-southeast-3	Ásia-Pacífico (Jacarta)
ap-southeast-4	Ásia-Pacífico (Melbourne)
cn-north-1	China (Pequim)
cn-northwest-1	China (Ningxia)
me-central-1	Oriente Médio (Emirados Árabes Unidos)
me-south-1	Middle East (Bahrain)
sa-east-1	South America (São Paulo)

Código	Nome
`il-central-1`	Israel (Tel Aviv)

Configurando a Amazon EventBridge para receber eventos de uma integração SaaS

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, selecione Fontes de eventos de parceiro.
3. Encontre o parceiro que deseja e selecione Configurar para esse parceiro.
4. Para copiar o ID da conta para a área de transferência, escolha Copiar
5. No painel de navegação, selecione Fontes de eventos de parceiro.
6. Acesse o site do parceiro e siga as instruções para criar uma origem de evento do parceiro usando o ID da sua conta. A origem do evento criado estará disponível somente para sua conta.
7. Volte para o EventBridge console e escolha Partner event sources no painel de navegação.
8. Selecione o botão ao lado da origem do evento do parceiro e selecione Associar ao barramento de eventos.

   O status da origem do evento muda de `Pending` para `Active`, e o nome do barramento de eventos é atualizado para corresponder ao nome da origem do evento do parceiro. Agora é possível começar a criar regras que combinam eventos da origem de eventos do parceiro. Para ter mais informações, consulte Como criar uma regra que corresponda a eventos de parceiros de SaaS.

   > **Note**
   >
   > Qualquer evento publicado por um parceiro em uma origem de evento de parceiro que não tenha sido associado a um barramento de eventos será imediatamente cancelado. Esses eventos não persistirão em repouso em EventBridge.

Como criar uma regra que corresponda a eventos de parceiros de SaaS

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.

2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra na mesma Região e barramento de eventos.
5. Em Barramento de eventos, selecione o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, escolha Barramento de eventos padrão da AWS . Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Tipo de Regra, escolha Regra com Padrão de Evento.
7. Escolha Próximo.
8. Em Fonte do evento, escolha Outra.
9. (Opcional) Em Eventos de amostra, escolha o tipo de evento.
10. Em Padrão de evento, insira um padrão de evento JSON.
11. Selecione Next (Próximo).
12. Em Tipos de destino, escolha Serviço da AWS .
13. Em Selecionar um destino, escolha o AWS serviço para o qual você deseja enviar informações ao EventBridge detectar um evento que corresponda ao padrão do evento.
14. Os campos exibidos variam de acordo com o serviço escolhido. Insira as informações específicas desse tipo de destino conforme necessário.
15. Para muitos tipos de alvo, EventBridge precisa de permissões para enviar eventos ao alvo. Nesses casos, EventBridge pode criar a função do IAM necessária para que sua regra seja executada. Execute um destes procedimentos:

    - Para criar um perfil do IAM automaticamente, escolha Criar um novo perfil para este recurso específico.
    - Para usar um perfil do IAM que você criou anteriormente, escolha Usar perfil existente e selecione o perfil existente na lista suspensa.
16. (Opcional) Para Configurações Adicionais, proceda da seguinte forma:

    a. Em Tempo Máximo do Evento, insira um valor entre um minuto (00:01) e 24 horas (24:00).
    b. Em Tentativas de Repetição, insira um número entre 0 e 185.
    c. Para fila de mensagens mortas, escolha se deseja usar uma fila padrão do Amazon SQS como fila de mensagens mortas. EventBridge envia eventos que correspondam a essa

regra para a fila de mensagens mortas se não forem entregues com sucesso ao destino. Faça um dos procedimentos a seguir:

- Escolha None (Nenhum) para não usar uma fila de mensagens não entregues.
- Escolha Selecionar uma fila do Amazon SQS na conta atual da AWS para usar como a fila de mensagens não entregues e depois selecione na lista suspensa a fila a ser usada.
- Escolha Selecionar uma fila do Amazon SQS em outra AWS conta como uma fila de mensagens mortas e, em seguida, insira o ARN da fila a ser usada. Você deve anexar uma política baseada em recursos à fila que conceda EventBridge permissão para enviar mensagens para ela. Para ter mais informações, consulte Como conceder permissões para a fila de mensagens não entregues.

17. (Opcional) Selecione Adicionar outro destino para adicionar outro destino a essa regra.
18. Escolha Next (Próximo).
19. (Opcional) Insira uma ou mais tags para a regra. Para ter mais informações, consulte EventBridge Etiquetas da Amazon.
20. Escolha Próximo.
21. Analise os detalhes da regra e selecione Criar regra.

Recebendo eventos usando URLs de AWS Lambda funções

> **Note**
>
> Para que o Inbound Webhook possa ser acessado por nossos parceiros, estamos criando um Open Lambda em sua AWS conta que é protegido no nível do aplicativo Lambda, verificando a assinatura de autenticação enviada pelo parceiro terceirizado. Revise esta configuração com sua equipe de segurança. Para obter mais informações, consulte Modelo de segurança e autenticação para URLs de função do Lambda.

Seu barramento de EventBridge eventos da Amazon pode usar uma URL de AWS Lambda função criada por um AWS CloudFormation modelo para receber eventos de provedores de SaaS compatíveis. Com URLs da função, os dados do evento são enviados para uma função do Lambda. Em seguida, a função converte esses dados em um evento que pode ser ingerido EventBridge e enviado para um barramento de eventos para processamento. Quando o evento estiver em um barramento de eventos, será possível usar regras para filtrar os eventos, aplicar qualquer transformação de entrada configurada e, em seguida, roteá-lo para o destino correto.

> **Note**
>
> A criação de URLs de função do Lambda aumentará seus custos mensais. Para obter mais informações, consulte Definição de preços do AWS Lambda.

Para configurar uma conexão EventBridge, primeiro você seleciona o provedor de SaaS com o qual deseja configurar uma conexão. Em seguida, você fornece um segredo de assinatura criado com esse provedor e seleciona o ônibus de EventBridge eventos para o qual enviar eventos. Por fim, você usa um AWS CloudFormation modelo e cria os recursos necessários para concluir a conexão.

Atualmente, os seguintes provedores de SaaS estão disponíveis para uso com o uso de URLs de EventBridge funções Lambda:

- GitHub
- Twilio

Tópicos

Configure uma conexão para o GitHub:

Etapa 1: criar a AWS CloudFormation pilha

Primeiro, use o EventBridge console da Amazon para criar uma CloudFormation pilha:

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. Do painel de navegação, escolha Quick starts.
3. Em Webhooks de entrada usando fURLs do Lambda, escolha Conceitos básicos.
4. Em GitHub, escolha Configurar.
5. Em Etapa 1: selecionar um barramento de eventos, selecione um barramento de eventos na lista suspensa. Esse barramento de eventos recebe dados do URL da função do Lambda fornecida para o GitHub. Também é possível criar um barramento de eventos selecionando Novo barramento de eventos.
6. Em Etapa 2: Configurar usando CloudFormation, escolha Novo GitHub webhook.
7. Selecione Eu confirmo que o Webhook de entrada que criei estará acessível ao público e escolha Confirmar.
8. Insira um nome para a pilha.
9. Em parâmetros, verifique se o barramento de eventos correto está listado e, em seguida, especifique um token seguro para o GitHubWebhookSecret. Para obter mais informações sobre como criar um token seguro, consulte Como configurar seu token secreto na documentação do GitHub.
10. Em Capacidades e transformações, selecione cada uma das seguintes opções:

    - Eu reconheço que isso AWS CloudFormation pode criar recursos do IAM.

- Eu reconheço que isso AWS CloudFormation pode criar recursos do IAM com nomes personalizados.
- Eu reconheço que isso AWS CloudFormation pode exigir a seguinte capacidade: **`CAPABILITY_AUTO_EXPAND`**

11. Selecione Criar pilha.

Etapa 2: criar um webhook do GitHub

Em seguida, crie o webhook no GitHub. Será necessário o token seguro e da URL da função do Lambda criados na etapa 2 para concluir esta etapa. Para obter mais informações, consulte Como criar webhooks na documentação do GitHub.

Configure uma conexão para um Twilio:

Etapa 1: encontrar seu token de autenticação Twilio

Para configurar uma conexão entre Twilio e EventBridge, primeiro configure a conexão Twilio com o token de autenticação, ou segredo, da sua Twilio conta. Para obter mais informações, consulte Tokens de autenticação e como alterá-los na documentação do Twilio.

Etapa 2: criar a AWS CloudFormation pilha

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Quick starts.
3. Em Webhooks de entrada usando fURLs do Lambda, escolha Conceitos básicos.
4. Em Twilio, escolha Configurar.
5. Em Etapa 1: selecionar um barramento de eventos, selecione um barramento de eventos na lista suspensa. Esse barramento de eventos recebe dados do URL da função do Lambda fornecida para o Twilio. Também é possível criar um barramento de eventos selecionando Novo barramento de eventos.
6. Em Etapa 2: Configurar usando CloudFormation, escolha Novo Twilio webhook.
7. Selecione Eu confirmo que o Webhook de entrada que criei estará acessível ao público e escolha Confirmar.
8. Insira um nome para a pilha.
9. Em parâmetros, verifique se o barramento de eventos correto está listado e, em seguida, insira o TwilioWebhookSecret criado na etapa 1.

10. Em Capacidades e transformações, selecione cada uma das seguintes opções:
    - Eu reconheço que isso AWS CloudFormation pode criar recursos do IAM.
    - Eu reconheço que isso AWS CloudFormation pode criar recursos do IAM com nomes personalizados.
    - Eu reconheço que isso AWS CloudFormation pode exigir o seguinte recurso: CAPABILITY_AUTO_EXPAND
11. Selecione Criar pilha.

Etapa 3: criar um webhook do Twilio

Depois de configurar a URL da função do Lambda, é preciso fornecê-la ao Twilio para que os dados do evento possam ser enviados. Para obter mais informações, consulte Configurar seu URL público com o Twilio na documentação do Twilio.

Atualizar o segredo do webhook ou o token de autenticação

Atualizar segredo do GitHub

> **Note**
>
> O GitHub não é compatível com dois segredos ao mesmo tempo. Você pode passar por um tempo de inatividade dos recursos enquanto o GitHub segredo e o segredo na AWS CloudFormation pilha estão fora de sincronia. GitHubas mensagens enviadas enquanto os segredos estiverem fora de sincronia falharão devido a assinaturas incorretas. GitHubEspere até que os CloudFormation segredos e estejam sincronizados e tente novamente.

1. Criar um novo segredo do GitHub. Para obter mais informações, consulte Segredos criptografados na documentação do GitHub.
2. Abra o AWS CloudFormation console em https://console.aws.amazon.com/cloudformation.
3. Do painel de navegação, escolha Pilhas.
4. Escolha a pilha do webhook que inclui o segredo que deseja atualizar.
5. Selecione Atualizar.
6. Verifique se a opção Usar modelo atual está selecionada e escolha Avançar.
7. Em GitHubWebhookSecret, desmarque Usar valor existente, insira o novo GitHub segredo que você criou na etapa 1 e escolha Avançar.

8. Selecione Next (Próximo).
9. Escolha Atualizar pilha.

Pode levar até uma hora para que o segredo seja propagado. Para reduzir esse tempo de inatividade, é possível atualizar o contexto de execução do Lambda.

Atualizar segredo do Twilio

> **Note**
>
> O Twilio não é compatível com dois segredos ao mesmo tempo. Você pode passar por um tempo de inatividade dos recursos enquanto o Twilio segredo e o segredo na AWS CloudFormation pilha estão fora de sincronia. Twilioas mensagens enviadas enquanto os segredos estiverem fora de sincronia falharão devido a assinaturas incorretas. TwilioEspere até que os CloudFormation segredos e estejam sincronizados e tente novamente.

1. Criar um novo segredo do Twilio. Para obter mais informações, consulte Tokens de autenticação e como alterá-los na documentação do Twilio.
2. Abra o AWS CloudFormation console em https://console.aws.amazon.com/cloudformation.
3. Do painel de navegação, escolha Pilhas.
4. Escolha a pilha do webhook que inclui o segredo que deseja atualizar.
5. Selecione Atualizar.
6. Verifique se a opção Usar modelo atual está selecionada e escolha Avançar.
7. Em TwilioWebhookSecret, desmarque Usar valor existente, insira o novo Twilio segredo que você criou na etapa 1 e escolha Avançar.
8. Selecione Next (Próximo).
9. Escolha Atualizar pilha.

Pode levar até uma hora para que o segredo seja propagado. Para reduzir esse tempo de inatividade, é possível atualizar o contexto de execução do Lambda.

Atualizar função do Lambda

A função Lambda criada pela CloudFormation pilha cria o webhook básico. Se você quiser personalizar a função Lambda para um caso de uso específico, como registro personalizado, use o

CloudFormation console para acessar a função e, em seguida, use o console Lambda para atualizar o código da função Lambda.

Acesse a função do Lambda

1. Abra o AWS CloudFormation console em https://console.aws.amazon.com/cloudformation.
2. Do painel de navegação, escolha Pilhas.
3. Escolha a pilha do webhook que inclui a função do Lambda que deseja atualizar.
4. Escolha a guia Recursos.
5. Para abrir a função do Lambda no console Lambda, em Physical ID, escolha a ID da função do Lambda.

Agora que acessou a função do Lambda, use o console do Lambda para atualizar o código da função.

Atualizar o código de função do Lambda

1. Em Ações, escolha Exportar função.
2. Escolha Baixar pacote de implantação e salve o arquivo no computador.
3. Descompacte o arquivo.zip do pacote de implantação, atualize o arquivo `app.py` e compacte o pacote de implantação atualizado, certificando-se de que todos os arquivos no arquivo.zip original estejam incluídos.
4. No console do Lambda, escolha a guia Código.
5. Em Fonte de código), escolha Fazer upload de.
6. Escolha Arquivo .zip e Fazer upload.
   - No seletor de arquivos, selecione a nova versão da imagem e escolha Abrir e Salvar.
7. Em Ações, escolha Publicar nova versão.

Tipos de eventos disponíveis

Atualmente, os seguintes tipos de eventos são compatíveis com CloudFormation barramentos de eventos:

- GitHub— Todos os tipos de eventos são suportados.
- Twilio: webhooks pós-evento são compatíveis

Cotas, códigos de erro e novas tentativas de entrega

Cotas

O número de solicitações recebidas para o webhook é limitado pelos serviços subjacentes. AWS A tabela a seguir inclui as cotas relevantes.

Serviço	Cota
AWS Lambda	Padrão: 10 execuções simultâneas Para obter mais informações sobre cotas, incluindo a solicitação de aumentos de cotas, consulte Cotas do Lambda.
AWS Secrets Manager	Padrão: cinco mil solicitações por segundo Para obter mais informações sobre cotas, incluindo a solicitação de aumentos de cota, consulte Cotas de serviço do AWS Secrets Manager. **Note** O número de solicitações por segundo é minimizado usando o cliente de cache em Python do AWS Secrets Manager.
Amazon EventBridge	Tamanho máximo de entrada de 256 KB para PutEvents ações. EventBridge impõe cotas tarifárias baseadas na região. Para ter mais informações, consulte ???.

Códigos de erro

Cada AWS serviço retorna códigos de erro específicos quando ocorrem erros. A tabela a seguir inclui os códigos de erro relevantes.

Serviço	Código de erro	Descrição
AWS Lambda	429 “” TooManyRequestsExp tion	A cota de execução simultâne a foi excedida.
AWS Secrets Manager	500 "Erro interno do servidor"	A cota de solicitações por segundo foi excedida.
Amazon EventBridge	500 "Erro interno do servidor"	A cota tarifária é excedida para a região.

Reentrega do evento

Quando ocorrem erros, é possível tentar novamente a entrega dos eventos afetados. Cada provedor de SaaS tem procedimentos de repetição diferentes.

GitHub

Use a API de webhooks GitHub para verificar o status de entrega de qualquer chamada de webhook e reenviar o evento, se necessário. Para obter mais informações, consulte a seguinte documentação do GitHub:

- Organização: reenvie uma entrega para um webhook da organização
- Repositório: reenvie uma entrega para um webhook do repositório
- Aplicação: reenvie uma entrega para um webhook de aplicação

Twilio

Os usuários do Twilio podem personalizar as opções de repetição de eventos usando substituições de conexão. Para obter mais informações, consulte Webhooks (retornos de chamada HTTP): substituições de conexão na documentação do Twilio.

Como receber eventos do Salesforce

Você pode usar EventBridge a Amazon para receber eventos Salesforce das seguintes formas:

- Usando o recurso Salesforce's Event Bus Relay para receber eventos diretamente em um ônibus de eventos EventBridge parceiro.
- Ao configurar um fluxo na Amazon AppFlow que usa Salesforce como fonte de dados. A Amazon AppFlow então envia Salesforce eventos para EventBridge usando um ônibus de eventos parceiro.

É possível enviar informações do evento para o Salesforce usando destinos de API. Depois que o evento é enviado para oSalesforce, ele pode ser processado por Fluxos ou Acionadores do Apex. Para obter mais informações sobre como configurar um destino de API do Salesforce, consulte ???.

Tópicos

Recebendo eventos do Salesforce usando a Retransmissão do barramento de eventos

Etapa 1: configurar o Salesforce Event Bus Relay e uma fonte de eventos do EventBridge parceiro

Quando você cria uma configuração de retransmissão de eventos ativadaSalesforce, Salesforce cria uma fonte de eventos do parceiro EventBridge no estado pendente.

Para configurar a Retransmissão do barramento de eventos do Salesforce

1. Configurar uma ferramenta de API REST
2. (Opcional) Defina um evento de plataforma
3. Crie um pipe para um evento de plataforma personalizado
4. Crie um membro do pipe para associar o evento da plataforma personalizada
5. Crie uma credencial nomeada
6. Criar uma configuração de retransmissão de eventos

Etapa 2: ativar a fonte de eventos do Salesforce parceiro no EventBridge console e iniciar a retransmissão de eventos

1. Abra a página de fontes de eventos do parceiro no EventBridge console.
2. Selecione a origem do evento do parceiro Salesforce criado na etapa 1.
3. Escolha Associar ao barramento de eventos.
4. Valide o nome do barramento de eventos do parceiro.
5. Selecione Associar.
6. Inicie a Retransmissão de eventos

Agora que você configurou e iniciou o Event Bus Relay e configurou a fonte de eventos do parceiro, você pode criar uma EventBridge regra que reaja aos eventos para filtrar e enviar os dados para um destino.

Recebendo eventos Salesforce usando a Amazon AppFlow

A Amazon AppFlow encapsula os eventos Salesforce em um envelope de EventBridge eventos. O exemplo a seguir mostra um Salesforce evento recebido por um ônibus de eventos EventBridge parceiro.


```
{
    "version": "0",
    "id": "5c42b99e-e005-43b3-c744-07990c50d2cc",
    "detail-type": "AccountChangeEvent",
    "source": "aws.partner/appflow.test/salesforce.com/364228160620/CustomSF-Source-
Final",
    "account": "000000000",
    "time": "2020-08-20T18:25:51Z",
    "region": "us-west-2",
    "resources": [],
    "detail": {
        "ChangeEventHeader": {
            "commitNumber": 248197218874,
            "commitUser": "0056g000003XW7AAAW",
            "sequenceNumber": 1,
            "entityName": "Account",
            "changeType": "UPDATE",
            "changedFields": [
                "LastModifiedDate",
```

```
                "Region__c"
            ],
            "changeOrigin": "com/salesforce/api/soap/49.0;client=SfdcInternalAPI/",
            "transactionKey": "000035af-b239-0581-9f14-461e4187de11",
            "commitTimestamp": 1597947935000,
            "recordIds": [
                "0016g00000MLhLeAAL"
            ]
        },
        "LastModifiedDate": "2020-08-20T18:25:35.000Z",
        "Region__c": "America"
    }
}
```

Etapa 1: configurar AppFlow a Amazon para ser usada Salesforce como fonte de eventos parceira

Para enviar eventos para EventBridge, primeiro você precisa configurar a Amazon AppFlow para usá-la Salesforce como fonte de eventos parceira.

1. No AppFlowconsole da Amazon, escolha Criar fluxo.
2. Na seção Detalhes do fluxo, em Nome do fluxo, insira um nome para seu fluxo.
3. (Opcional) Insira uma descrição para o fluxo e escolha Próximo.
4. Em Detalhes da origem, escolha Salesforce no menu suspenso Nome da origem e escolha Conectar para criar uma nova conexão.
5. Na caixa de diálogo Conectar-se ao Salesforce, escolha Produção ou Sandbox para o ambiente da Salesforce.
6. No campo Nome da conexão, insira um nome exclusivo para a conexão e escolha Continuar.
7. Na caixa de diálogo Salesforce, faça o seguinte:
    a. Insira suas credenciais de login da Salesforce para fazer login na Salesforce.
    b. Selecione Salesforce eventos para os tipos de dados AppFlow a serem processados pela Amazon.
8. No menu suspenso Escolher Salesforce evento, selecione o tipo de evento para o qual enviar. EventBridge
9. Para um destino, selecione Amazon EventBridge.
10. Selecione Criar nova origem de eventos para parceiros.
11. (Opcional) Especifique um sufixo exclusivo para a origem do evento do parceiro.

12. Escolha Gerar origem de eventos do parceiro.
13. Escolha um bucket do Amazon S3 para armazenar arquivos de carga útil de eventos maiores que 256 KB.
14. Na seção Acionador de fluxo, verifique se a opção Executar fluxo no evento está selecionada. Esta configuração garante que o fluxo seja executado quando um novo evento Salesforce ocorrer.
15. Selecione Next (Próximo).
16. Para mapeamento de campo, selecione Mapear todos os campos diretamente. Como alternativa, é possível selecionar os campos que são de interesse na lista Nome do campo de origem.

    Para obter mais informações, consulte Mapear campos de dados.
17. Selecione Next (Próximo).
18. (Opcional) Configure filtros para campos de dados na Amazon AppFlow.
19. Selecione Next (Próximo).
20. Revise as configurações e escolha Criar fluxo.

Com o fluxo configurado, a Amazon AppFlow cria uma nova fonte de eventos de parceiros que você precisa associar a um ônibus de eventos de parceiros em sua conta.

Etapa 2: Configurar EventBridge para receber Salesforce eventos

Certifique-se de que o AppFlow fluxo da Amazon que é acionado a partir de Salesforce eventos com EventBridge como destino esteja configurado antes de seguir as instruções nesta seção.

Para configurar EventBridge para receber Salesforce eventos

1. Abra a página de fontes de eventos do parceiro no EventBridge console.
2. Selecione a origem do evento do parceiro Salesforce criado na etapa 1.
3. Escolha Associar ao barramento de eventos.
4. Valide o nome do barramento de eventos do parceiro.
5. Selecione Associar.
6. No AppFlow console da Amazon, abra o fluxo que você criou e escolha Ativar fluxo.
7. Abra a página Regras no EventBridge console.
8. Escolha Criar Regra.

9. Insira um nome exclusivo para a regra.
10. Na seção Padrão de evento, escolha a seção Definir padrão.
11. Em Padrão de correspondência de eventos, escolha Padrão predefinido por serviço.
12. Na seção Provedor de serviços, selecione Todos os eventos.
13. Em Selecionar barramento de eventos, escolha Barramento de evento personalizado ou parceiro.
14. Selecione o ônibus de eventos que você associou à fonte de eventos do AppFlow parceiro da Amazon.
15. Em Selecionar alvos, escolha o AWS serviço que deve agir quando a regra for executada. Uma regra pode ter até cinco destinos.
16. Escolha Criar.

O serviço de destino recebe todos os eventos Salesforce configurados para sua conta. Para filtrar os eventos ou enviar alguns eventos para destinos diferentes, você pode usar a filtragem baseada em conteúdo com padrões de eventos.

> **Note**
>
> Para eventos maiores que 256 KB, a Amazon AppFlow não envia o evento completo para. EventBridge Em vez disso, a Amazon AppFlow coloca o evento em um bucket do S3 na sua conta e, em seguida, envia um evento para EventBridge com um ponteiro para o bucket do Amazon S3. É possível usar o ponteiro para obter o evento completo do bucket.

Como depurar os eventos de entrega

Os problemas de entrega de eventos podem ser difíceis de identificar e EventBridge oferecem algumas maneiras de depurar e se recuperar de falhas na entrega de eventos.

Como EventBridge tenta realizar eventos novamente

Às vezes, um evento não é entregue com êxito ao destino especificado em uma regra. Isso pode acontecer, por exemplo:

- Se o recurso de destino não estiver disponível
- Devido às condições da rede

Quando um evento não é entregue com sucesso a um alvo devido a erros recuperáveis, EventBridge tenta enviar o evento novamente. São definidos os tempo de tentativa e o número de tentativas nas configurações da política de repetição do destino. Por padrão, EventBridge tenta enviar novamente o evento por 24 horas e até 185 vezes com um recuo exponencial e instabilidade, ou atraso aleatório.

Se um evento não for entregue após o esgotamento de todas as tentativas, o evento será cancelado e EventBridge não continuará sendo processado.

Usando filas de cartas mortas para processar eventos não entregues

Para evitar a perda de eventos após eles não serem entregues a um destino, você pode configurar uma fila de mensagens não entregues (DLQ) e enviar todos os eventos que falharam para processamento posterior.

EventBridge As DLQs são filas padrão do Amazon SQS usadas para armazenar eventos EventBridge que não puderam ser entregues com sucesso a um destino. Ao criar uma regra e adicionar um destino, é possível escolher se quer ou não usar uma DLQ. Ao configurar uma DLQ, é possível reter todos os eventos que não foram entregues com êxito. Em seguida, é possível resolver o problema que resultou na falha na entrega do evento e processar os eventos posteriormente.

Quando você configura uma DLQ para o destino de uma regra, EventBridge envia os eventos com invocações falhadas para a fila selecionada do Amazon SQS.

Os erros de eventos são tratados de diferentes maneiras. Alguns eventos são descartados ou enviados para uma DLQ sem nenhuma tentativa de repetição. Por exemplo, para erros que resultam da falta de permissões para um destino ou de um recurso de destino que não existe mais, todas as tentativas falham até que uma ação seja tomada para resolver o problema subjacente. Em vez de tentar novamente, EventBridge envia esses eventos diretamente para o DLQ, se você tiver um.

Quando a entrega de um evento falha, EventBridge publica um evento nas CloudWatch métricas da Amazon indicando que uma meta `invocation` falhou. Se você usa um DLQ, métricas adicionais são enviadas para CloudWatch incluir `InvocationsSentToDLQ` e. `InvocationsFailedToBeSentToDLQ`

Você também pode especificar DLQs para barramentos de eventos, se você usar AWS KMS chaves gerenciadas pelo cliente para criptografar eventos em repouso. Para ter mais informações, consulte ???.

Cada mensagem em sua DLQ incluirá os seguintes atributos personalizados:

- `RULE_ARN`

- TARGET_ARN
- ERROR_CODE

  A seguinte é uma amostra dos códigos de erro que uma DLQ pode retornar:

  - CONNECTION_FAILURE
  - CROSS_ACCOUNT_INGESTION_FAILED
  - CROSS_REGION_INGESTION_FAILED
  - ERROR_FROM_TARGET
  - EVENTS_IN_BATCH_REQUEST_REJECTED
  - EVENTS_IN_BATCH_REQUEST_REJECTED
  - FAILED_TO_ASSUME_ROLE
  - INTERNAL_ERROR
  - INVALID_JSON
  - INVALID_PARAMETER
  - NO_PERMISSIONS
  - NO_RESOURCE
  - RESOURCE_ALREADY_EXISTS
  - RESOURCE_LIMIT_EXCEEDED
  - RESOURCE_MODIFICATION_COLLISION
  - SDK_CLIENT_ERROR
  - THIRD_ACCOUNT_HOP_DETECTED
  - THIRD_REGION_HOP_DETECTED
  - THROTTLING
  - TIMEOUT
  - TRANSIENT_ASSUME_ROLE
  - UNKNOWN
- ERROR_MESSAGE
- EXHAUSTED_RETRY_CONDITION

  As seguintes condições podem ser retornadas:

  - MaximumRetryAttempts
  - MaximumEventAgeInSeconds

- `RETRY_ATTEMPTS`

O seguinte vídeo aborda as configurações de DLQs: Como usar filas de mensagens não entregues (DLQs)

Tópicos

- Considerações sobre o uso de uma fila de mensagens não entregues
- Como conceder permissões para a fila de mensagens não entregues
- Como reenviar eventos de uma fila de mensagens não entregues

Considerações sobre o uso de uma fila de mensagens não entregues

Considere o seguinte ao configurar um DLQ para. EventBridge

- Somente filas padrão são compatíveis. Você não pode usar uma fila FIFO para uma entrada de DLQ. EventBridge
- EventBridge inclui metadados de eventos e atributos de mensagem na mensagem, incluindo: o código de erro, a mensagem de erro, a condição de repetição esgotada, o ARN da regra, as tentativas de repetição e o ARN de destino. É possível usar esses valores para identificar um evento e a causa da falha.
- Permissões para DLQs na mesma conta:
  - Se você adicionar um destino a uma regra usando o console e escolher uma fila do Amazon SQS na mesma conta, uma política baseada em recursos que concede EventBridge acesso à fila será anexada à fila para você.
  - Se você usar a `PutTargets` operação da EventBridge API para adicionar ou atualizar um destino para uma regra e escolher uma fila do Amazon SQS na mesma conta, deverá conceder manualmente as permissões para a fila selecionada. Para saber mais, consulte Como conceder permissões para a fila de mensagens não entregues.
- Permissões para usar filas do Amazon SQS de uma conta diferente. AWS
  - Se criar uma regra no console, as filas de outras contas não serão exibidas para você selecionar. É preciso fornecer o ARN para a fila na outra conta e anexar manualmente uma política baseada em recursos para conceder permissão à fila. Para saber mais, consulte Como conceder permissões para a fila de mensagens não entregues.

  - Se criar uma regra usando a API, deverá anexar manualmente uma política baseada em recursos às filas do SQS em outra conta que seja usada como fila de mensagens não entregues. Para saber mais, consulte Como conceder permissões para a fila de mensagens não entregues.
- A fila do Amazon SQS que é usada deve estar na mesma região em que a regra foi criada.

Como conceder permissões para a fila de mensagens não entregues

Para entregar eventos com sucesso à fila, é EventBridge preciso ter permissão para fazer isso. Quando você especifica uma DLQ usando o EventBridge console, as permissões são adicionadas automaticamente. Isso inclui:

- Quando você configura uma DLQ para o destino de uma regra.
- Quando você configura uma DLQ para um barramento de eventos em que você especificou esse EventBridge uso, use an AWS KMS chave gerenciada pelo cliente para criptografar eventos em repouso.

  Para ter mais informações, consulte ???.

Se você especificar uma DLQ usando a API ou usar uma fila que esteja em uma AWS conta diferente, deverá criar manualmente uma política baseada em recursos que conceda as permissões necessárias e, em seguida, anexá-la à fila.

Exemplo de permissões de fila de mensagens mortas do Target

A política baseada em recursos a seguir demonstra como conceder as permissões necessárias para enviar mensagens de eventos EventBridge para uma fila do Amazon SQS. O exemplo de política EventBridge concede ao serviço permissões para usar a `SendMessage` operação para enviar mensagens para uma fila chamada "MyEventDLQ". A fila deve estar na região us-west-2 na conta 123456789012. AWS A `Condition` instrução permite somente solicitações provenientes de uma regra chamada "MyTestRule" criada na região us-west-2 na conta 123456789012. AWS

```
{
  "Sid": "Dead-letter queue permissions",
  "Effect": "Allow",
  "Principal": {
     "Service": "events.amazonaws.com"
  },
```

```
  "Action": "sqs:SendMessage",
  "Resource": "arn:aws:sqs:us-west-2:123456789012:MyEventDLQ",
  "Condition": {
    "ArnEquals": {
      "aws:SourceArn": "arn:aws:events:us-west-2:123456789012:rule/MyTestRule"
    }
  }
}
```

Exemplo de permissões de fila de mensagens mortas do Event Bus

A política baseada em recursos a seguir demonstra como conceder as permissões necessárias ao especificar uma DLQ para um barramento de eventos. Nesse caso, `aws:SourceArn` especifica o ARN do barramento de eventos que envia os eventos para a DLQ. Aqui, novamente neste exemplo, a fila deve estar na mesma região do barramento de eventos.

```
{
  "Sid": "Dead-letter queue permissions",
  "Effect": "Allow",
  "Principal": {
     "Service": "events.amazonaws.com"
  },
  "Action": "sqs:SendMessage",
  "Resource": "arn:aws:sqs:region:account-id:queue-name",
  "Condition": {
    "ArnEquals": {
      "aws:SourceArn": "arn:aws:events:region:account-id:event-bus/event-bus-arn"
    }
  }
}
```

Para anexar a política à fila, use o console do Amazon SQS, abra a fila, escolha a Política de acesso e edite a política. Você também pode usar o AWS CLI Para saber mais, consulte Permissões do Amazon SQS.

Como reenviar eventos de uma fila de mensagens não entregues

É possível remover mensagens de uma DLQ de duas maneiras:

- Evite escrever a lógica de consumidor do Amazon SQS: defina sua DLQ como uma origem de evento para a função do Lambda para drenar sua DLQ.

- Escreva a lógica de consumo do Amazon SQS — Use a API AWS , o SDK do Amazon SQS AWS CLI ou para escrever uma lógica de consumidor personalizada para sondar, processar e excluir as mensagens no DLQ.

Padrões de EventBridge eventos da Amazon

Os padrões de eventos têm a mesma estrutura que os eventos aos quais correspondem. As regras usam padrões de evento para selecionar eventos e enviá-los para os destinos. Um padrão de evento corresponde a um evento ou não corresponde.

> **Important**
>
> Em EventBridge, é possível criar regras que podem gerar higher-than-expected cobranças e estrangulamentos. Por exemplo, é possível criar, por engano, uma regra que leva a um loop infinito, em que uma regra é acionada recursivamente sem fim. Supõe-se que uma regra possa detectar que as ACLs foram alteradas em um bucket do Amazon S3 e acionar o software para alterá-las para o estado desejado. Se a regra não for gravada cuidadosamente, a alteração subsequente às ACLs disparará a regra novamente, criando um loop infinito.
>
> Para obter orientação sobre como escrever regras precisas e padrões de eventos para minimizar esses resultados inesperados, consulte ??? e ???.

O seguinte vídeo aborda os princípios básicos dos padrões de eventos: Como filtrar eventos

Tópicos

- Como criar padrões de eventos
- Eventos de exemplo e padrões de eventos
- Correspondência de valores nulos e strings vazias nos padrões de eventos da Amazon EventBridge
- Matrizes nos padrões de EventBridge eventos da Amazon
- Filtragem de conteúdo nos padrões de EventBridge eventos da Amazon
- Testando um padrão de evento usando o EventBridge Sandbox
- Melhores práticas ao definir padrões de EventBridge eventos da Amazon

O evento a seguir mostra um AWS evento simples do Amazon EC2.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ec2:us-west-1:123456789012:instance/i-1234567890abcdef0"
  ],
  "detail": {
    "instance-id": "i-1234567890abcdef0",
    "state": "terminated"
  }
}
```

O padrão de eventos a seguir processa todos os eventos `instance-termination` do Amazon EC2.

```
{
  "source": ["aws.ec2"],
  "detail-type": ["EC2 Instance State-change Notification"],
  "detail": {
    "state": ["terminated"]
  }
}
```

Como criar padrões de eventos

Para criar um padrão de evento, especifique os campos aos quais o padrão de evento deve corresponder. Especifique somente os campos usados para fazer a correspondência. O exemplo anterior do padrão de evento fornece valores somente para três campos: os campos de nível superior `"source"` e`"detail-type"`, e o `"state"` campo dentro do campo do `"detail"` objeto. EventBridge ignora todos os outros campos do evento ao aplicar a regra.

Para que um padrão de evento corresponda a um evento, o evento deve conter todos os nomes de campos listados no padrão do evento. Os nomes de campos também devem aparecer no evento com a mesma estrutura de aninhamento.

Ao gravar padrões de regras para corresponder, pode usar a API `TestEventPattern` ou o comando `test-event-pattern` da CLI para garantir que o padrão corresponda ao JSON correto. Para obter mais informações, consulte TestEventPattern.

Valores de eventos correspondentes

Em um padrão de evento, o valor correspondente está em uma matriz JSON, entre colchetes ("[", "]") para que possa fornecer vários valores. Por exemplo, para combinar eventos do Amazon EC2 ou AWS Fargate, você pode usar o seguinte padrão, que corresponde a eventos em que o valor do `"source"` campo é ou`"aws.ec2"`. `"aws.fargate"`

```
{
    "source": ["aws.ec2", "aws.fargate"]
}
```

Considerações ao criar padrões de eventos

A seguir, algumas considerações ao criar seus padrões de eventos:

- EventBridge ignora os campos do evento que não estão incluídos no padrão do evento. O efeito é que há um curinga `"*": "*"` para campos que não aparecem no padrão do evento.
- Os valores de correspondência nos padrões de evento seguem regras JSON. Você pode incluir strings entre aspas (“), números e as palavras-chave `true`, `false` e `null`.
- Para strings, EventBridge usa a character-by-character correspondência exata sem dobrar maiúsculas e minúsculas ou qualquer outra normalização de string.
- Para números, EventBridge usa representação de string. Por exemplo, 300, 300.0 e 3.0e2 não são considerados iguais.
- Se vários padrões forem especificados para o mesmo campo JSON, usará EventBridge somente o último.
- Lembre-se de que, ao EventBridge compilar padrões de eventos para uso, ele usa ponto (.) como caractere de junção.

  Isso significa que EventBridge tratará os seguintes padrões de eventos como idênticos:

```
## has no dots in keys
{ "detail" : { "state": { "status": [ "running" ] } } }

## has dots in keys
```

```
{ "detail" : { "state.status": [ "running" ] } }
```

E que os dois padrões de eventos corresponderão aos dois seguintes eventos:

```
## has no dots in keys
{ "detail" : { "state": { "status": "running" } } }

## has dots in keys
{ "detail" : { "state.status": "running"  } }
```

> **Note**
>
> Isso descreve EventBridge o comportamento atual e não se deve confiar que não mude.

- Os padrões de eventos contendo campos duplicados são inválidos. Se um padrão contiver campos duplicados, considerará EventBridge apenas o valor final do campo.

  Por exemplo, os seguintes padrões de eventos corresponderão ao mesmo evento:

```
## has duplicate keys
{
  "source": ["aws.s3"],
  "source": ["aws.sns"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail":  {
      "eventSource": ["s3.amazonaws.com"],
      "eventSource": ["sns.amazonaws.com"]
  }
}

## has unique keys
{
  "source": ["aws.sns"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail": { "eventSource": ["sns.amazonaws.com"] }
}
```

  E EventBridge trata os dois eventos a seguir como idênticos:

```
## has duplicate keys
{
```

```
  "source": ["aws.s3"],
  "source": ["aws.sns"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail":  [
    {
      "eventSource": ["s3.amazonaws.com"],
      "eventSource": ["sns.amazonaws.com"]
    }
  ]
}

## has unique keys
{
  "source": ["aws.sns"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail": [
    { "eventSource": ["sns.amazonaws.com"] }
  ]
}
```

> **Note**
>
> Isso descreve EventBridge o comportamento atual e não se deve confiar que não mude.

Operações de comparação para uso em padrões de eventos

Abaixo, um resumo de todos os operadores de comparação disponíveis em EventBridge.

Os operadores de comparação só funcionam em nós folha, com exceção de `$or` e `anything-but`.

Comparação	Exemplo	Sintaxe da regra
E	Local é “Nova York” e o dia é “Segunda-feira”	`"Location": [ "New York" ], "Day": ["Monday"]`
Qualquer coisa, menos	Estado é qualquer valor além de “inicializar”.	`"state": [ { "anything-but": "initializing" } ]`

Comparação	Exemplo	Sintaxe da regra
Tudo menos (começa com)	A região não está nos EUA.	`"Region": [ { "anything-but": {"prefix": "us-" } } ]`
Qualquer coisa, menos (termina com)	FileName não termina com uma extensão.png.	`"FileName": [ { "anything-but": { "suffix": ".png" } } ]`
Qualquer coisa, menos (ignorar maiúsculas e minúsculas)	Estado é qualquer valor além de “inicializar” ou qualquer outra variação de maiúsculas e minúsculas, como “INICIALIZAR”.	`"state": : [{ "anything-but": { "equals-ignore-case": "initializing" }}]}`
Qualquer coisa, menos usar um curinga	FileName não é um caminho de arquivo que inclua `/lib/`.	`"FilePath" : [{ "anything-but": { "wildcard": "*/lib/*" }}]`
Começa com	A região está nos EUA.	`"Region": [ {"prefix": "us-" } ]`
Começa com (ignorar maiúsculas e minúsculas)	O nome do serviço começa com as letras “eventb”, independentemente do caso.	`{"service" : [{ "prefix": { "equals-ignore-case": "eventb" }}]}`
Vazio	LastName está vazio.	`"LastName": [""]`
Igual	Name é “Alice”	`"Name": [ "Alice" ]`
É igual a (ignorar maiúsculas e minúsculas)	Name é “Alice”	`"Name": [ { "equals-ignore-case": "alice" } ]`
Termina com	FileName termina com uma extensão.png	`"FileName": [ { "suffix": ".png" } ]`

Comparação	Exemplo	Sintaxe da regra
Termina com (ignorar maiúsculas e minúsculas)	O nome do serviço termina com as letras “tbridge” ou qualquer outra variação de caixa, como “TBRIDGE”.	`{"service" : [{ "suffix": { "equals-ignore-case": "tBridge" }}]}`
Existe	ProductName existe	`"ProductName": [ { "exists": true } ]`
Não existe	ProductName não existe	`"ProductName": [ { "exists": false } ]`
Não	Weather é qualquer valor, exceto “Raining” (Chovendo)	`"Weather": [ { "anything-but": [ "Raining" ] } ]`
Nulo	UserID é null	`"UserID": [ null ]`
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 ] } ]`
Ou	PaymentType é “Crédito” ou “Débito”	`"PaymentType": [ "Credit", "Debit"]`
Ou (vários campos)	Location é “New York” ou Day é “Monday”.	`"$or": [ { "Location": [ "New York" ] }, { "Day": [ "Monday" ] } ]`
Curinga	Qualquer arquivo com extensão .png, localizado na pasta "dir"	`"FileName": [ { "wildcard": "dir/*.png" } ]`

Eventos de exemplo e padrões de eventos

É possível usar todos os tipos e valores de dados JSON para combinar eventos. Os exemplos a seguir mostram eventos e os padrões de eventos que correspondem a eles.

Correspondência de campos

É possível combinar o valor de um campo. Considere o evento do Amazon EC2 Auto Scaling a seguir.

```
{
  "version": "0",
  "id": "3e3c153a-8339-4e30-8c35-687ebef853fe",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:31:47Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "eventVersion": "",
    "responseElements": null
  }
}
```

Para o evento anterior, é possível usar o campo `"responseElements"` para corresponder.

```
{
  "source": ["aws.autoscaling"],
  "detail-type": ["EC2 Instance Launch Successful"],
  "detail": {
   "responseElements": [null]
  }
}
```

Valor para corresponder

Considere o evento do Amazon Macie, que está truncado, a seguir.

```
{
  "version": "0",
  "id": "0948ba87-d3b8-c6d4-f2da-732a1example",
  "detail-type": "Macie Finding",
  "source": "aws.macie",
  "account": "123456789012",
  "time": "2021-04-29T23:12:15Z",
```

```
  "region":"us-east-1",
  "resources": [

  ],
  "detail": {
    "schemaVersion": "1.0",
    "id": "64b917aa-3843-014c-91d8-937ffexample",
    "accountId": "123456789012",
    "partition": "aws",
    "region": "us-east-1",
    "type": "Policy:IAMUser/S3BucketEncryptionDisabled",
    "title": "Encryption is disabled for the S3 bucket",
    "description": "Encryption is disabled for the Amazon S3 bucket. The data in the
 bucket isn’t encrypted
        using server-side encryption.",
    "severity": {
        "score": 1,
        "description": "Low"
    },
    "createdAt": "2021-04-29T15:46:02Z",
    "updatedAt": "2021-04-29T23:12:15Z",
    "count": 2,
.
.
.
```

O padrão de evento a seguir corresponde a qualquer evento que tenha uma pontuação de gravidade de 1 e uma contagem de 2.

```
{
  "source": ["aws.macie"],
  "detail-type": ["Macie Finding"],
  "detail": {
    "severity": {
      "score": [1]
    },
    "count":[2]
  }
}
```

Correspondência de valores nulos e strings vazias nos padrões de eventos da Amazon EventBridge

> **Important**
>
> Em EventBridge, é possível criar regras que podem gerar higher-than-expected cobranças e estrangulamentos. Por exemplo, é possível criar, por engano, uma regra que leva a um loop infinito, em que uma regra é acionada recursivamente sem fim. Supõe-se que uma regra possa detectar que as ACLs foram alteradas em um bucket do Amazon S3 e acionar o software para alterá-las para o estado desejado. Se a regra não for gravada cuidadosamente, a alteração subsequente às ACLs disparará a regra novamente, criando um loop infinito.
>
> Para obter orientação sobre como escrever regras precisas e padrões de eventos para minimizar esses resultados inesperados, consulte ??? e ???.

Também é possível criar um padrão de evento que corresponde a um evento de campo que tem um valor nulo ou uma string vazia. Considere o exemplo de evento a seguir:

Veja as práticas recomendadas para evitar cobranças e controle de utilização acima do esperado

```
{
  "version": "0",
  "id": "3e3c153a-8339-4e30-8c35-687ebef853fe",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:31:47Z",
  "region": "us-east-1",
  "resources": [
   ],
  "detail": {
    "eventVersion": "",
    "responseElements": null
   }
}
```

Para corresponder eventos onde o valor de `eventVersion` é uma string vazia, use o padrão a seguir, que corresponderia com o exemplo de evento.

```
{
  "detail": {
     "eventVersion": [""]
  }
}
```

Para corresponder eventos onde o valor de `responseElements` é nulo, use o padrão a seguir, que corresponderia com o exemplo de evento.

```
{
  "detail": {
     "responseElements": [null]
  }
}
```

> **Note**
>
> Valores Null e Strings vazias não são permutáveis em correspondência padrão. Um padrão de evento que corresponde a strings vazias não corresponde aos valores de `null`.

Matrizes nos padrões de EventBridge eventos da Amazon

O valor de cada campo em um padrão de evento é uma matriz contendo um ou mais valores. Um padrão de evento corresponde ao evento se algum dos valores na matriz corresponder ao valor no evento. Se o valor no evento for uma matriz, o padrão do evento será correspondente se a interseção da matriz do padrão do evento e da matriz do evento for não vazia.

> **Important**
>
> Em EventBridge, é possível criar regras que podem gerar higher-than-expected cobranças e estrangulamentos. Por exemplo, é possível criar, por engano, uma regra que leva a um loop infinito, em que uma regra é acionada recursivamente sem fim. Supõe-se que uma regra possa detectar que as ACLs foram alteradas em um bucket do Amazon S3 e acionar o software para alterá-las para o estado desejado. Se a regra não for gravada cuidadosamente, a alteração subsequente às ACLs disparará a regra novamente, criando um loop infinito.
>
> Para obter orientação sobre como escrever regras precisas e padrões de eventos para minimizar esses resultados inesperados, consulte ??? e ???.

Por exemplo, considere um padrão de evento que inclui o seguinte campo:

```
"resources": [
   "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f",
   "arn:aws:ec2:us-east-1:111122223333:instance/i-b188560f",
   "arn:aws:ec2:us-east-1:444455556666:instance/i-b188560f",
]
```

O padrão de exemplo anterior corresponderia a um evento que inclui o campo a seguir porque o primeiro item na matriz de padrão do evento corresponde ao segundo item na matriz de evento.

```
"resources": [
   "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:eb56d16b-bbf0-401d-
b893-d5978ed4a025:autoScalingGroupName/ASGTerminate",
   "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f"
]
```

Filtragem de conteúdo nos padrões de EventBridge eventos da Amazon

A Amazon EventBridge oferece suporte à filtragem declarativa de conteúdo usando padrões de eventos. Com a filtragem de conteúdo, é possível escrever padrões de eventos complexos que só são acionados sob condições muito específicas. Por exemplo, é possível criar um padrão de evento que corresponda a um evento quando:

- Um campo do evento está dentro de um intervalo numérico específico.
- O evento vem de um endereço IP específico.
- Não existe um campo específico no JSON do evento.

> **Important**
>
> Em EventBridge, é possível criar regras que podem gerar higher-than-expected cobranças e estrangulamentos. Por exemplo, é possível criar, por engano, uma regra que leva a um loop infinito, em que uma regra é acionada recursivamente sem fim. Supõe-se que uma regra possa detectar que as ACLs foram alteradas em um bucket do Amazon S3 e acionar o software para alterá-las para o estado desejado. Se a regra não for gravada cuidadosamente, a alteração subsequente às ACLs disparará a regra novamente, criando um loop infinito.
>
> Para obter orientação sobre como escrever regras precisas e padrões de eventos para minimizar esses resultados inesperados, consulte ??? e ???.

Tipos de filtro

- Correspondência de prefixo
- Correspondência de sufixo
- Correspondência anything-but
- Correspondência numérica
- Correspondência de endereço IP
- Existe correspondência
- quals-ignore-caseCombinação E
- Como corresponder usando curingas

- Exemplo complexo com várias correspondências
- Exemplo complexo com correspondências de $or

Correspondência de prefixo

É possível fazer uma correspondência de um evento com um valor na origem do evento no prefixo. É possível usar a correspondência de prefixos para valores de string.

Por exemplo, haveria correspondência do padrão de evento a seguir em qualquer evento em que o campo `"time"` começasse com `"2017-10-02"`, como `"time": "2017-10-02T18:43:48Z"`.

```
{
  "time": [ { "prefix": "2017-10-02" } ]
}
```

Correspondência de prefixo ao ignorar maiúsculas e minús

Você também pode combinar um valor de prefixo, independentemente da maiúscula dos caracteres com os quais um valor começa, usando `equals-ignore-case` em conjunto com `prefix.`

Por exemplo, o padrão de evento a seguir corresponderia a qualquer evento em que o `service` campo começasse com a cadeia de caracteres`EventB`, mas também com `EVENTBeventb`,, ou com qualquer outra capitalização desses caracteres.

```
{
  "detail": {"service" : [{ "prefix": { "equals-ignore-case": "EventB" }}]}
}
```

Correspondência de sufixo

É possível combinar um evento dependendo do sufixo de um valor na origem do evento. É possível usar a correspondência de sufixos para valores de string.

Por exemplo, haveria correspondência do padrão de evento a seguir em qualquer evento em que o campo `"FileName"` terminasse com a extensão de arquivo `.png`.

```
{
  "FileName": [ { "suffix": ".png" } ]
```

```
}
```

Correspondência de sufixos ao ignorar maiúsculas e min

Você também pode combinar um valor de sufixo, independentemente da maiúscula dos caracteres com os quais um valor termina, usando `equals-ignore-case` em conjunto com `suffix`.

Por exemplo, o padrão de evento a seguir corresponderia a qualquer evento em que o `FileName` campo terminasse com a cadeia de caracteres`.png`, mas também `.PNG` a qualquer outra capitalização desses caracteres.

```
{
  "detail": {"FileName" : [{ "suffix": { "equals-ignore-case": ".png" }}]}
}
```

Correspondência anything-but

Qualquer coisa, exceto a correspondência, corresponde a qualquer coisa, exceto o que está especificado na regra.

É possível excluir correspondências com strings e valores numéricos, incluindo listas que contenham somente strings ou números.

O evento a seguir mostra tudo menos uma correspondência com uma lista de strings e números.

```
{
  "detail": {
    "state": [ { "anything-but": "initializing" } ]
  }
}

{
  "detail": {
    "x-limit": [ { "anything-but": 123 } ]
  }
}
```

O evento a seguir mostra tudo menos uma correspondência com uma lista de strings.

```
{
```

```
  "detail": {
    "state": [ { "anything-but": [ "stopped", "overloaded" ] } ]
  }
}
```

O evento a seguir mostra tudo menos uma correspondência com uma lista de números.

```
{
  "detail": {
    "x-limit": [ { "anything-but": [ 100, 200, 300 ] } ]
  }
}
```

Tudo menos combinar, ignorando maiúsculas e minúsculas

Você também pode usar `equals-ignore-case` em conjunto com`anything-but`, para combinar valores de string, independentemente da maiúscula dos caracteres.

O padrão de evento a seguir corresponde aos `state` campos que não contêm a string “initializing”, “INITIALIZING”, “Initializing” ou qualquer outra capitalização desses caracteres.

```
{
  "detail": {"state" : [{ "anything-but": { "equals-ignore-case": "initializing" }}]}
}
```

Você também pode usar `equals-ignore-case` em conjunto com `anything-but` para comparar com uma lista de valores:

```
{
  "detail": {"state" : [{ "anything-but": { "equals-ignore-case": ["initializing",
 "stopped"] }}]}
}
```

Tudo menos correspondência em prefixos

Você pode usar `prefix` em conjunto com `anything-but` para combinar valores de string que não começam com o valor especificado. Isso inclui valores únicos ou uma lista de valores.

O padrão de evento a seguir mostra tudo, exceto a correspondência, que corresponde a qualquer evento que não tenha o prefixo `"init"` no campo. `"state"`

```
{
  "detail": {
    "state": [ { "anything-but": { "prefix": "init" } } ]
  }
}
```

O padrão de eventos a seguir mostra tudo menos a correspondência usada com uma lista de valores de prefixo. Esse padrão de evento corresponde a qualquer evento que não tenha o prefixo `"init"` nem o `"stop"` `"state"` campo.

```
{
"detail": {
  "state" : [{ "anything-but": { "prefix": ["init", "stop"] } } ] }
  }
}
```

Tudo menos correspondência em sufixos

Você pode usar `suffix` em conjunto com `anything-but` para combinar valores de string que não terminam com o valor especificado. Isso inclui valores únicos ou uma lista de valores.

O padrão de evento a seguir corresponde a todos os valores do `FileName` campo que não terminam com`.txt`.

```
{
  "detail": {
    "FileName": [ { "anything-but": { "suffix": ".txt" } } ]
  }
}
```

O padrão de evento a seguir mostra tudo menos a correspondência usada com uma lista de valores de sufixo. Esse padrão de evento corresponde a todos os valores do `FileName` campo que não terminam com `.txt` ou`.rtf`.

```
{
  "detail": {
    "FileName": [ { "anything-but": { "suffix": [".txt", ".rtf"] } } ]
  }
}
```

Tudo menos combinar usando curingas

Você pode usar o caractere curinga (*) nos valores especificados para qualquer coisa, exceto a correspondência. Isso inclui valores únicos ou uma lista de valores.

O padrão de evento a seguir corresponde a todos os valores do `FileName` campo que não contêm`/lib/`.

```
{
"detail": {
  "FilePath" : [{ "anything-but": { "wildcard": "*/lib/*" }}]
  }
}
```

O padrão de eventos a seguir mostra tudo menos a correspondência usada com uma lista de valores, incluindo curingas. Esse padrão de evento corresponde a todos os valores do `FileName` campo que não contêm `/lib/` nem`/bin/`.

```
{
"detail": {
  "FilePath" : [{ "anything-but": { "wildcard": ["*/lib/*", "*/bin/*"] }}]
  }
}
```

Para ter mais informações, consulte ???.

Correspondência numérica

A correspondência numérica funciona com valores que são números JSON. Está limitada a valores entre -5.0e9 e +5.0e9 inclusive, com 15 dígitos de precisão ou seis dígitos à direita do ponto decimal.

O exemplo a seguir mostra a correspondência numérica para um padrão de evento que corresponde somente aos eventos que são verdadeiros para todos os campos.

```
{
  "detail": {
    "c-count": [ { "numeric": [ ">", 0, "<=", 5 ] } ],
    "d-count": [ { "numeric": [ "<", 10 ] } ],
    "x-limit": [ { "numeric": [ "=", 3.018e2 ] } ]
  }
```

```
}
```

Correspondência de endereço IP

É possível usar a correspondência de endereços IP para endereços IPv4 e IPv6. O padrão de eventos a seguir mostra o endereço IP correspondente aos endereços IP que começam com 10.0.0 e terminam com um número entre 0 e 255.

```
{
  "detail": {
    "sourceIPAddress": [ { "cidr": "10.0.0.0/24" } ]
  }
}
```

Existe correspondência

Existe correspondência funciona na presença ou ausência de um campo no JSON do evento.

A correspondência de existênciafunciona somente em nós folha. Ela não funciona em nós intermediários.

O padrão de evento a seguir corresponde a qualquer evento que tenha um campo `detail.state`.

```
{
  "detail": {
    "state": [ { "exists": true  } ]
  }
}
```

Os padrões de evento anterior seriam correspondentes ao evento de exemplo a seguir.

```
{
  "version": "0",
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2015-11-11T21:29:54Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"],
```

```
  "detail": {
    "instance-id": "i-abcd1111",
    "state": "pending"
  }
}
```

O padrão do evento anterior NÃO corresponde ao evento seguinte porque não tem um campo `detail.state`.

```
{
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "resources": [ "arn:aws:ec2:us-east-1:123456789012:instance/i-02ebd4584a2ebd341" ],
  "detail": {
    "c-count" : {
       "c1" : 100
    }
  }
}
```

quals-ignore-caseCombinação E

A quals-ignore-case correspondência E funciona em valores de string, independentemente do caso.

O padrão de evento a seguir corresponde a qualquer evento que tenha um campo `detail-type` que corresponda à string especificada, independentemente do caso.

```
{
  "detail-type": [ { "equals-ignore-case": "ec2 instance state-change notification" } ]
}
```

Os padrões de evento anterior seriam correspondentes ao evento de exemplo a seguir.

```
{
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "resources": [ "arn:aws:ec2:us-east-1:123456789012:instance/i-02ebd4584a2ebd341" ],
  "detail": {
    "c-count" : {
       "c1" : 100
    }
  }
}
```

Como corresponder usando curingas

É possível usar o caractere curinga (*) para encontrar o valor correspondente a valores de string em padrões de eventos.

> **Note**
>
> No momento, o caractere curinga é compatível somente com as regras do barramento de eventos.

Considerações ao usar curingas em seus padrões de eventos:

- É possível especificar qualquer número de caracteres curinga em um determinado valor de cadeia de caracteres; no entanto, caracteres curinga consecutivos não são compatíveis.
- EventBridge suporta o uso do caractere de barra invertida (\) para especificar os caracteres literais * e\ em filtros curinga:
  - A string * representa o caractere literal *
  - A string \\ representa o caractere literal \

  Não há compatibilidade para usar a barra invertida no escape de outros caracteres.

Curingas e complexidade do padrão de eventos

Há um limite para a complexidade de uma regra que usa curingas. Se uma regra for muito complexa, EventBridge retornará um `InvalidEventPatternException` ao tentar criar a regra. Se sua regra gerar esse erro, considere usar a orientação abaixo para reduzir a complexidade do padrão do evento:

- Reduza o número de caracteres curinga usados

  Use somente caracteres curinga quando realmente precisar comparar com vários valores possíveis. Por exemplo, considere o seguinte padrão de eventos, em que deseja comparar com os barramentos de eventos na mesma região:

```
{
"EventBusArn": [ { "wildcard": "*:*:*:*:*:event-bus/*" } ]
}
```

No caso acima, muitas das seções do ARN serão baseadas diretamente na região em que seus barramentos de eventos residem. Portanto, se estiver usando a região `us-east-1`, um padrão menos complexo que ainda corresponda aos valores desejados pode ser o seguinte exemplo:

```
{
"EventBusArn": [ { "wildcard": "arn:aws:events:us-east-1:*:event-bus/*" } ]
}
```

- Reduza as sequências de caracteres repetidas que ocorrem após um caractere curinga

  Ter a mesma sequência de caracteres aparecendo várias vezes após o uso de um curinga aumenta a complexidade do processamento do padrão do evento. Reformule seu padrão de eventos para minimizar sequências repetidas. Por exemplo, considere o seguinte exemplo, que corresponde ao arquivo de nome `doc.txt` de arquivo de qualquer usuário:

  ```
  {
  "FileName": [ { "wildcard": "/Users/*/dir/dir/dir/dir/dir/doc.txt" } ]
  }
  ```

  Se soubesse que o arquivo `doc.txt` só ocorreria no caminho especificado, poderia reduzir a sequência de caracteres repetidos desta forma:

  ```
  {
  "FileName": [ { "wildcard": "/Users/*/doc.txt" } ]
  }
  ```

Exemplo complexo com várias correspondências

É possível combinar várias regras de correspondência em um padrão de evento mais complexo. Por exemplo, o padrão de evento a seguir combina `anything-but` e `numeric`.

```
{
  "time": [ { "prefix": "2017-10-02" } ],
  "detail": {
    "state": [ { "anything-but": "initializing" } ],
    "c-count": [ { "numeric": [ ">", 0, "<=", 5 ] } ],
    "d-count": [ { "numeric": [ "<", 10 ] } ],
    "x-limit": [ { "anything-but": [ 100, 200, 300 ] } ]
  }
```

```
}
```

> **Note**
>
> Ao criar padrões de eventos, se incluir uma chave mais de uma vez, a última referência será aquela usada para avaliar eventos. Por exemplo, para o seguinte padrão:
>
> ```
> {
>   "detail": {
>     "location": [ { "prefix": "us-" } ],
>     "location": [ { "anything-but": "us-east" } ]
>   }
> }
> ```
>
> somente `{ "anything-but": "us-east" }` serão levados em consideração ao avaliar a `location`.

Exemplo complexo com correspondências de `$or`

Também é possível criar padrões de eventos complexos que verificam se os valores de algum campo coincidem em vários campos. Use `$or` para criar um padrão de evento que corresponda se algum dos valores de vários campos for correspondido.

Observe que pode incluir outros tipos de filtro, como correspondência numérica e matrizes, na correspondência de padrões para campos individuais em sua estrutura `$or`.

O seguinte padrão de evento corresponderá se alguma das condições a seguir for atendida:

- O campo `c-count` é maior que 0 ou menor que ou igual a 5.
- O campo `d-count` é inferior a 10.
- O campo `x-limit` é igual a 3.018e2.

```
{
  "detail": {
    "$or": [
      { "c-count": [ { "numeric": [ ">", 0, "<=", 5 ] } ] },
      { "d-count": [ { "numeric": [ "<", 10 ] } ] },
```

```
      { "x-limit": [ { "numeric": [ "=", 3.018e2 ] } ] }
    ]
  }
}
```

> **Note**
>
> As APIs que aceitam um padrão de evento (como `PutRule`, `CreateArchive`, `UpdateArchive` `eTestEventPattern`) lançarão uma `InvalidEventPatternException` se o uso de resultados `$or` em mais de mil combinações de regras.
> Para determinar o número de combinações de regras em um padrão de evento, multiplique o número total de argumentos de cada matriz `$or` no padrão de evento. Por exemplo, o padrão acima contém uma única matriz `$or` com três argumentos, então o número total de combinações de regras também é três. Se adicionasse outra matriz `$or` com dois argumentos, o total de combinações de regras seria então seis.

Testando um padrão de evento usando o EventBridge Sandbox

As regras usam padrões de evento para selecionar eventos e enviá-los para os destinos. Os padrões de eventos têm a mesma estrutura que os eventos aos quais correspondem. Um padrão de evento corresponde a um evento ou não corresponde.

Definir um padrão de evento normalmente faz parte do processo maior de criar uma nova regra ou da edição de uma existente. No entanto EventBridge, usando o Sandbox in, você pode definir rapidamente um padrão de evento e usar um evento de amostra para confirmar se o padrão corresponde aos eventos desejados, sem precisar criar ou editar uma regra. Depois de testar seu padrão de evento, você EventBridge tem a opção de criar uma nova regra usando esse padrão de evento diretamente da sandbox.

Para obter mais informações sobre padrões de eventos, consulte ???.

> **Important**
>
> Em EventBridge, é possível criar regras que podem gerar higher-than-expected cobranças e estrangulamentos. Por exemplo, é possível criar, por engano, uma regra que leva a um loop infinito, em que uma regra é acionada recursivamente sem fim. Supõe-se que

uma regra possa detectar que as ACLs foram alteradas em um bucket do Amazon S3 e acionar o software para alterá-las para o estado desejado. Se a regra não for gravada cuidadosamente, a alteração subsequente às ACLs disparará a regra novamente, criando um loop infinito.
Para obter orientação sobre como escrever regras precisas e padrões de eventos para minimizar esses resultados inesperados, consulte ??? e ???.

Para testar um padrão de evento usando a EventBridge sandbox

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Recursos do desenvolvedor, depois selecione Sandbox e, na página do Sandbox, escolha a guia Padrão do evento.
3. Em Origem do evento, escolha AWS eventos ou eventos de EventBridge parceiros.
4. Na seção Eventos de amostra, escolha um Tipo de evento de amostra com o qual deseja testar seu padrão de evento.

   Os seguintes tipos de eventos estão disponíveis:

   - AWS eventos — Selecione entre os eventos emitidos pelo suporte Serviços da AWS.
   - EventBridge eventos de parceiros — selecione entre os eventos emitidos por serviços terceirizados que oferecem suporte EventBridge, como o Salesforce.
   - Inserir o meu próprio: insira o seu próprio evento em texto JSON.

     Você também pode usar um evento AWS ou um evento de um parceiro como ponto de partida para criar seu próprio evento personalizado.

     1. Selecione AWS eventos ou eventos de EventBridge parceiros.
     2. Use o menu suspenso Exemplos de eventos para selecionar o evento que deseja usar como ponto de partida para seu evento personalizado.

        EventBridge exibe o evento de amostra.
     3. Selecione Copiar.
     4. Selecione Inserir meu próprio para o Tipo de evento.
     5. Exclua a estrutura de eventos de amostra no painel de edição JSON e cole o evento AWS ou o evento do parceiro em seu lugar.
     6. Edite o JSON do evento para criar seu próprio evento de amostra.

5. Escolha um Método de criação. Você pode criar um padrão de evento a partir de um EventBridge esquema ou modelo, ou pode criar um padrão de evento personalizado.

   Existing schema

   Para usar um EventBridge esquema existente para criar o padrão de evento, faça o seguinte:

   1. Na seção Método de criação, em Método, selecione Usar esquema.
   2. Na seção Padrão de evento, em Tipo de esquema, selecione Selecionar esquema do registro do esquema.
   3. Em Registro do esquema, escolha a caixa suspensa e insira o nome de um registro do esquema, como `aws.events`. Também é possível selecionar uma opção na lista suspensa que aparece.
   4. Em Esquema, escolha a caixa suspensa e insira o nome do esquema a ser usado. Por exemplo, `aws.s3@0bjectDeleted`. Também é possível selecionar uma opção na lista suspensa que aparece.
   5. Na seção Modelos, escolha o botão Editar ao lado de qualquer atributo para abrir suas propriedades. Defina os campos Relacionamento e Valor conforme necessário e escolha Definir para salvar o atributo.

      > **Note**
      >
      > Para obter informações sobre a definição de um atributo, escolha o ícone Informações ao lado do nome do atributo. Para obter uma referência sobre como definir propriedades de atributos em seu evento, abra a seção Observação da caixa de diálogo de propriedades de atributos.
      > Para excluir as propriedades de um atributo, escolha o botão Editar para esse atributo e escolha Limpar.

   6. Escolha Gerar padrão de evento em JSON para gerar e validar seu padrão de evento como texto JSON.
   7. Para testar o evento de amostra em relação ao seu padrão de teste, escolha Padrão de teste.

      EventBridge exibe uma caixa de mensagem informando se o evento de amostra corresponde ao padrão do evento.

      Também é possível escolher uma das seguintes opções:

- Copiar: copie o padrão do evento para a área de transferência do seu dispositivo.
- Aprimorar: facilita a leitura do texto JSON adicionando quebras de linha, tabulações e espaços.

Custom schema

Para escrever um esquema personalizado e convertê-lo em um padrão de evento, faça o seguinte:

1. Na seção Método de criação, em Método, escolha Usar esquema.
2. Na seção Padrão de evento, em Tipo de esquema, escolha Inserir esquema.
3. Insira o seu esquema na caixa de texto. É preciso formatar o esquema como texto JSON válido.
4. Na seção Modelos, escolha o botão Editar ao lado de qualquer atributo para abrir suas propriedades. Defina os campos Relacionamento e Valor conforme necessário e escolha Definir para salvar o atributo.

   > **Note**
   >
   > Para obter informações sobre a definição de um atributo, escolha o ícone Informações ao lado do nome do atributo. Para obter uma referência sobre como definir propriedades de atributos em seu evento, abra a seção Observação da caixa de diálogo de propriedades de atributos.
   > Para excluir as propriedades de um atributo, escolha o botão Editar para esse atributo e escolha Limpar.

5. Escolha Gerar padrão de evento em JSON para gerar e validar seu padrão de evento como texto JSON.
6. Para testar o evento de amostra em relação ao seu padrão de teste, escolha Padrão de teste.

   EventBridge exibe uma caixa de mensagem informando se o evento de amostra corresponde ao padrão do evento.

   Também é possível escolher uma das seguintes opções:

   - Copiar: copie o padrão do evento para a área de transferência do seu dispositivo.

- Aprimorar: facilita a leitura do texto JSON adicionando quebras de linha, tabulações e espaços.

Event pattern

Para escrever um padrão de evento personalizado no formato JSON, faça o seguinte:

1. Na seção Método de criação, em Método, escolha Padrão personalizado (editor JSON).
2. Em Padrão de evento, insira seu padrão de evento personalizado em texto formatado em JSON.
3. Para testar o evento de amostra em relação ao seu padrão de teste, escolha Padrão de teste.

   EventBridge exibe uma caixa de mensagem informando se o evento de amostra corresponde ao padrão do evento.

   Também é possível escolher uma das seguintes opções:

   - Copiar: copie o padrão do evento para a área de transferência do seu dispositivo.
   - Aprimorar: facilita a leitura do texto JSON adicionando quebras de linha, tabulações e espaços.
   - Formulário de padrão de evento: abre o padrão de evento no Criador de padrões. Se o padrão não puder ser renderizado no Pattern Builder no estado em que se encontra, EventBridge avisa você antes de abrir o Pattern Builder.

6. (Opcional) Para criar uma regra com esse padrão de evento e atribuir a regra a um barramento de eventos específico, escolha Criar regra com padrão.

   EventBridge leva você para a Etapa 1 de Criar regra, que você pode usar para criar uma regra e atribuí-la ao barramento de eventos de sua escolha.

   Observe que a etapa 2: criar padrão de evento contém as informações do padrão de evento já especificadas e que pode aceitar ou atualizar.

   Para obter mais informações sobre como criar regras, consulte ???.

Melhores práticas ao definir padrões de EventBridge eventos da Amazon

Abaixo estão algumas das práticas recomendadas a serem consideradas ao definir padrões de eventos em suas regras de barramento de eventos.

Evite escrever loops infinitos

Em EventBridge, é possível criar regras que levam a loops infinitos, onde uma regra é acionada repetidamente. Por exemplo, uma regra pode detectar que as ACLs foram alteradas em um bucket do S3 e acionar o software para alterá-las para o estado desejado. Se a regra não for gravada cuidadosamente, a alteração subsequente às ACLs disparará a regra novamente, criando um loop infinito.

Para evitar esses problemas, escreva os padrões de eventos para que suas regras sejam o mais precisas possível, para que correspondam apenas aos eventos que você realmente deseja enviar ao destino. No exemplo acima, seria criado um padrão de evento para corresponder aos eventos para que as ações acionadas não disparassem novamente a mesma regra. Por exemplo, crie um padrão de evento em sua regra que corresponda aos eventos somente se as ACLs estiverem em um estado incorreto, em vez de após qualquer alteração. Para obter mais informações, consulte ??? e ???.

Um loop infinito pode rapidamente causar cobranças acima do esperado. Isto também pode levar ao controle de utilização e atraso na entrega do evento. É possível monitorar o limite superior de suas taxas de invocação para receber avisos sobre picos inesperados no volume.

Use o orçamento para alertar você quando as cobranças excederem o limite especificado. Para obter mais informações, consulte Gerenciamento de seus custos com orçamentos.

Torne os padrões de eventos os mais precisos possível

Quanto mais preciso for o padrão de eventos, maior será a probabilidade dele corresponder somente aos eventos que você realmente deseja e evitar correspondências inesperadas quando novos eventos forem adicionados a uma origem de eventos ou eventos existentes forem atualizados para incluir novas propriedades.

Os padrões de eventos podem incluir filtros que correspondem a:

- Metadados do evento sobre o evento, como `source`, `detail-type`, `account` ou `region`.
- Dados do evento, ou seja, os campos dentro do objeto `detail`.

- Conteúdo do evento ou os valores reais dos campos dentro do objeto `detail`.

A maioria dos padrões é simples, como especificar somente filtros `source` e `detail-type`. No entanto, EventBridge os padrões incluem a flexibilidade de filtrar qualquer chave ou valor do evento. Além disso, é possível aplicar filtros de conteúdo, como filtros `prefix` e `suffix`, para melhorar a precisão de seus padrões. Para ter mais informações, consulte ???.

Especifique a origem do evento e o tipo de detalhe como filtros

É possível reduzir a geração de loops infinitos e a correspondência de eventos indesejados tornando seus padrões de eventos mais precisos usando os campos de metadados `source` e `detail-type`.

Quando precisar combinar valores específicos em dois ou mais campos, use o operador de comparação `$or`, em vez de listar todos os valores possíveis em uma única matriz de valores.

Para eventos que são entregues por meio de AWS CloudTrail, recomendamos que você use o `eventName` campo como filtro.

O exemplo de padrão de evento a seguir corresponde a `CreateQueue` ou `SetQueueAttributes` do serviço Amazon Simple Queue Service `CreateKey` ou a `DisableKeyRotation` eventos do AWS Key Management Service serviço.

```
{
  "detail-type": ["AWS API Call via CloudTrail"],
  "$or": [{
      "source": [
        "aws.sqs"
        ],
      "detail": {
        "eventName": [
          "CreateQueue",
          "SetQueueAttributes"
        ]
      }
    },
    {
      "source": [
        "aws.kms"
        ],
      "detail": {
        "eventName": [
```

```
          "CreateKey",
          "DisableKeyRotation"
        ]
      }
    }
  ]
}
```

Especifique conta e região como filtros

Incluir os campos `account` e `region` em seu padrão de eventos ajuda a limitar a correspondência de eventos entre contas ou regiões.

Especifique filtros de conteúdo

A filtragem baseada em conteúdo pode ajudar a melhorar a precisão do padrão de eventos ao mesmo tempo em que mantém a duração do padrão de eventos no mínimo. Por exemplo, a correspondência com base em um intervalo numérico pode ser útil em vez de listar todos os valores numéricos possíveis.

Para ter mais informações, consulte ???.

Defina seus padrões de eventos para considerar as atualizações da origem de eventos

Ao criar padrões de eventos, é necessário levar em consideração que os esquemas e domínios de eventos podem evoluir e se expandir com o tempo. Aqui, novamente, tornar seus padrões de eventos o mais precisos possível ajuda a limitar correspondências inesperadas se a origem do evento mudar ou se expandir.

Por exemplo, suponha que você esteja comparando eventos de um novo microsserviço que publica eventos relacionados a pagamentos. Inicialmente, o serviço usa o domínio `acme.payments` e publica um único evento, `Payment accepted`:

```
{
  "detail-type": "Payment accepted",
  "source": "acme.payments",
  "detail": {
    "type": "credit",
    "amount": "100",
```

```
    "date": "2023-06-10",
    "currency": "USD"
    }
  }
}
```

Nesse momento, é possível criar um padrão de evento simples que corresponda aos eventos aceitos por pagamento:

```
{ “source” : “acme.payments” }
```

No entanto, suponha que o serviço introduza posteriormente um novo evento para pagamentos rejeitados:

```
{
  "detail-type": "Payment rejected",
  "source": "acme.payments",
  "detail": {
  }
}
```

Neste caso, o padrão de evento simples que você criou agora corresponderá aos eventos `Payment accepted` e `Payment rejected`. EventBridge direciona os dois tipos de eventos para o destino especificado para processamento, possivelmente introduzindo falhas de processamento e custos adicionais de processamento.

Para definir o escopo do seu padrão de eventos somente `Payment accepted` para eventos, é preciso especificar tanto `source` como `detail-type`, no mínimo:

```
{
  "detail-type": "Payment accepted",
  "source": "acme.payments"
  }
}
```

Também é possível especificar conta e região em seu padrão de eventos, para limitar ainda mais quando eventos entre contas ou regiões correspondem a essa regra.

```
{
  "account": "012345678910",
```

```
  "source": "acme.payments",
  "region": "AWS-Region",
  "detail-type": "Payment accepted"
}
```

Validar padrões de eventos

Para garantir que as regras correspondam aos eventos desejados, é altamente recomendável que você valide seus padrões de eventos. Você pode validar seus padrões de eventos usando o EventBridge console ou a API:

- No EventBridge console, você pode criar e testar padrões de eventos como parte da criação de uma regra ou separadamente usando o Sandbox.
- Você pode testar seus padrões de eventos programaticamente usando a TestEventPatternação.

EventBridge Regras da Amazon

Você especifica o que EventBridge acontece com os eventos entregues em cada ônibus de eventos. Para fazer isso, você cria regras. Uma regra especifica quais eventos enviar para quais alvos serão processados. Uma única regra pode enviar um evento para vários destinos, que são executados em paralelo.

É possível criar dois tipos de regras:

- Regras que coincidem nos dados do evento

  Você pode criar regras que correspondam aos eventos recebidos com base nos critérios de dados do evento (chamado de padrão de evento). Um padrão de evento define a estrutura do evento e os campos aos quais uma regra corresponde para iniciar a ação de destino. Se um evento corresponder aos critérios definidos no padrão do evento, ele será EventBridge enviado para o (s) alvo (s) especificado (s).

  Para ter mais informações, consulte ???.
- Regras que são executadas de acordo com um cronograma

  Você também pode criar regras que enviem eventos para os destinos especificados em intervalos específicos. Por exemplo, para executar periodicamente uma Lambda função, você pode criar uma regra para ser executada de acordo com uma agenda.

  > **Note**
  >
  > EventBridge oferece o Amazon EventBridge Scheduler, um programador sem servidor que permite criar, executar e gerenciar tarefas a partir de um serviço gerenciado central. EventBridge O Scheduler é altamente personalizável e oferece escalabilidade aprimorada em relação às regras EventBridge programadas, com um conjunto mais amplo de operações e serviços de API de destino. AWS
  > Recomendamos que você use o EventBridge Scheduler para invocar alvos em uma agenda. Para ter mais informações, consulte ???.

O seguinte vídeo aborda os princípios básicos das regras: O que são regras

Regras EventBridge gerenciadas pela Amazon

Além das regras que você cria, AWS os serviços podem criar e gerenciar EventBridge regras em sua AWS conta que são necessárias para determinadas funções nesses serviços. Elas são chamadas de regras gerenciadas.

Quando um serviço cria uma regra gerenciada, ele também pode criar uma IAM política que conceda permissão a esse serviço para criar a regra. As políticas do IAM criadas desta forma têm um escopo limitado com permissões no nível do recurso para permitir a criação apenas das regras necessárias.

É possível excluir regras gerenciadas usando a opção Forçar exclusão, mas elas devem ser excluídas somente se você tiver certeza de que o outro serviço não precisará mais da regra. Do contrário, excluir uma regra gerenciada faz os recursos que dependem dela deixar de funcionar.

Como criar regras do Amazon EventBridge que reagem a eventos

Para agir em eventos recebidos pelo Amazon EventBridge, você pode criar regras. Quando um evento corresponde o padrão do evento definido na sua regra, o EventBridge envia o evento para o destino especificado e aciona a ação definida na regra.

O seguinte vídeo explora a criação de diferentes tipos de regras e como testá-las: Saber mais sobre regras.

Use o procedimento a seguir para criar uma regra do Amazon EventBridge que responda aos eventos.

Crie uma regra que reaja aos eventos

As etapas a seguir explicam como criar uma regra que o EventBridge usa para combinar eventos à medida que eles são enviados para o barramento de eventos especificado.

Etapas

- Defina a regra
- Criar o padrão de eventos
- Selecione destinos
- Configurar tags e regra de revisão

Defina a regra

Primeiro, insira um nome e uma descrição para a regra para identificá-la. Também é preciso definir o barramento de eventos em que sua regra procura eventos que correspondam a um padrão de eventos.

Para definir os detalhes da regra

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Informe um Nome para a regra e, opcionalmente, uma Descrição para a regra.

Uma regra não pode ter o mesmo nome que outra regra na mesma Região da AWS e no mesmo barramento de eventos.

5. Em Barramento de eventos, selecione o barramento de eventos para associar com essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, selecione Barramento de eventos padrão da AWS. Quando um AWS service (Serviço da AWS) na sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Rule type (Tipo de regra), escolha Rule with an event pattern (Regra com um padrão de evento).
7. Escolha Next (Próximo).

Criar o padrão de eventos

Em seguida, crie o padrão do evento. Para fazer isso, especifique a origem do evento, escolha a base para o padrão do evento e defina os atributos e valores aos quais corresponder. Também é possível gerar o padrão de evento em JSON e testá-lo em um evento de amostra.

Para criar o padrão de eventos

1. Em Event source (Origem do evento), escolha Eventos da AWS ou eventos de parceiro do EventBridge.
2. (Opcional) Na seção Eventos de amostra, escolha um Tipo de evento de amostra com o qual deseja testar seu padrão de evento.

   Os seguintes tipos de eventos estão disponíveis:

   - Eventos da AWS: elecione entre os eventos emitidos pelos Serviços da AWS compatíveis.
   - Eventos de parceiros do EventBridge: selecione entre os eventos emitidos por serviços de terceiros que são compatíveis com o EventBridge, como o Salesforce.
   - Inserir o meu próprio: insira o seu próprio evento em texto JSON.

   Também é possível usar um evento da AWS ou um evento de um parceiro como ponto de partida para criar seu próprio evento personalizado.

   1. Selecione Eventos da AWS ou Eventos de parceiros do EventBridge.
   2. Use o menu suspenso Exemplos de eventos para selecionar o evento que deseja usar como ponto de partida para seu evento personalizado.

O EventBridge exibe o evento de amostra.

3. Selecione Copiar.
4. Selecione Inserir meu próprio para o Tipo de evento.
5. Exclua a estrutura de eventos de amostra no painel de edição JSON e cole o evento da AWS ou do parceiro no lugar.
6. Edite o JSON do evento para criar seu próprio evento de amostra.

3. Escolha um Método de criação. Também é possível criar um padrão de evento a partir de um esquema ou modelo do EventBridge ou criar um padrão de evento personalizado.

   Existing schema

   Para usar um esquema do EventBridge existente para criar o padrão de evento, faça o seguinte:

   1. Na seção Método de criação, em Método, selecione Usar esquema.
   2. Na seção Padrão de evento, em Tipo de esquema, selecione Selecionar esquema do registro do esquema.
   3. Em Registro do esquema, escolha a caixa suspensa e insira o nome de um registro do esquema, como `aws.events`. Também é possível selecionar uma opção na lista suspensa que aparece.
   4. Em Esquema, escolha a caixa suspensa e insira o nome do esquema a ser usado. Por exemplo, `aws.s3@0bjectDeleted`. Também é possível selecionar uma opção na lista suspensa que aparece.
   5. Na seção Modelos, escolha o botão Editar ao lado de qualquer atributo para abrir suas propriedades. Defina os campos Relacionamento e Valor conforme necessário e escolha Definir para salvar o atributo.

      > **Note**
      >
      > Para obter informações sobre a definição de um atributo, escolha o ícone Informações ao lado do nome do atributo. Para obter uma referência sobre como definir propriedades de atributos em seu evento, abra a seção Observação da caixa de diálogo de propriedades de atributos.
      > Para excluir as propriedades de um atributo, escolha o botão Editar para esse atributo e escolha Limpar.

6. Escolha Gerar padrão de evento em JSON para gerar e validar seu padrão de evento como texto JSON.
7. (Opcional) Para testar o evento de amostra em relação ao seu padrão de teste, escolha Padrão de teste.

   O EventBridge exibe uma caixa de mensagem informando se seu evento de amostra corresponde ao padrão do evento.

   Também é possível escolher uma das seguintes opções:

   - Copiar: copie o padrão do evento para a área de transferência do seu dispositivo.
   - Aprimorar: facilita a leitura do texto JSON adicionando quebras de linha, tabulações e espaços.

Custom schema

Para escrever um esquema personalizado e convertê-lo em um padrão de evento, faça o seguinte:

1. Na seção Método de criação, em Método, escolha Usar esquema.
2. Na seção Padrão de evento, em Tipo de esquema, escolha Inserir esquema.
3. Insira o seu esquema na caixa de texto. É preciso formatar o esquema como texto JSON válido.
4. Na seção Modelos, escolha o botão Editar ao lado de qualquer atributo para abrir suas propriedades. Defina os campos Relacionamento e Valor conforme necessário e escolha Definir para salvar o atributo.

   > **Note**
   >
   > Para obter informações sobre a definição de um atributo, escolha o ícone Informações ao lado do nome do atributo. Para obter uma referência sobre como definir propriedades de atributos em seu evento, abra a seção Observação da caixa de diálogo de propriedades de atributos.
   > Para excluir as propriedades de um atributo, escolha o botão Editar para esse atributo e escolha Limpar.

5. Escolha Gerar padrão de evento em JSON para gerar e validar seu padrão de evento como texto JSON.

6. (Opcional) Para testar o evento de amostra em relação ao seu padrão de teste, escolha Padrão de teste.

   O EventBridge exibe uma caixa de mensagem informando se seu evento de amostra corresponde ao padrão do evento.

   Também é possível escolher uma das seguintes opções:

   - Copiar: copie o padrão do evento para a área de transferência do seu dispositivo.
   - Aprimorar: facilita a leitura do texto JSON adicionando quebras de linha, tabulações e espaços.

Event pattern

Para escrever um padrão de evento personalizado no formato JSON, faça o seguinte:

1. Na seção Método de criação, em Método, escolha Padrão personalizado (editor JSON).
2. Em Padrão de evento, insira seu padrão de evento personalizado em texto formatado em JSON.
3. (Opcional) Para testar o evento de amostra em relação ao seu padrão de teste, escolha Padrão de teste.

   O EventBridge exibe uma caixa de mensagem informando se seu evento de amostra corresponde ao padrão do evento.

   Também é possível escolher uma das seguintes opções:

   - Copiar: copie o padrão do evento para a área de transferência do seu dispositivo.
   - Aprimorar: facilita a leitura do texto JSON adicionando quebras de linha, tabulações e espaços.
   - Formulário de padrão de evento: abre o padrão de evento no Criador de padrões. Se o padrão não puder ser renderizado no Criador do Padrão no estado em que se encontra, o EventBridge avisa antes de abrir o Criador do Padrão.

4. Escolha Next (Próximo).

Selecione destinos

Escolha um ou mais destinos para receber eventos que correspondam ao padrão especificado. Os destinos podem incluir um barramento de eventos do EventBridge, destinos da API do EventBridge, incluindo parceiros de SaaS, como Salesforce, ou outros AWS service (Serviço da AWS).

Para selecionar destinos

1. Para Tipo de destino, escolha um dos seguintes tipos de destino:

   Event bus

   Para selecionar um barramento de eventos do EventBridge, selecione o Barramento de eventos do EventBridge e faça o seguinte:

   - Para usar um barramento de eventos da mesma forma Região da AWS que esta regra:
     1. Selecione Barramento de eventos na mesma conta e região.
     2. Em Barramento de eventos para destino, escolha a caixa suspensa e insira o nome do barramento de eventos. Também é possível selecionar o barramento de eventos na lista suspensa.

        Para obter mais informações, consulte ???.
   - Para usar um barramento de eventos em uma conta ou Região da AWS diferente como esta regra:
     1. Selecione Barramento de eventos em uma conta ou região diferente.
     2. Para Barramento de eventos como destino, insira o ARN do barramento de eventos que deseja usar.

        Para obter mais informações, consulte:

        - ???
        - ???

   API destination

   Para usar um destino da API EventBridge, selecione o Destino da API do EventBridge e faça o seguinte:

- Para usar um destino de API existente, selecione Usar um destino de API existente. Em seguida, selecione um destino de API na lista suspensa.
- Para criar um novo destino de API, selecione Criar um novo destino de API. Em seguida, forneça os seguintes detalhes para o destino:
  - Nome: insira um nome para o destino.

    Os nomes devem ser exclusivos em sua Conta da AWS. Nomes podem ter até 64 caracteres. Os caracteres válidos A-Z, a-z, 0-9 e . _ - (hífen).
  - (Opcional) Descrição: insira uma descrição para o destino.

    As descrições podem ter até 512 caracteres.
  - Endpoint de destino da API: o endpoint de URL para o destino.

    O URL do endpoint deve começar com **`https`**. É possível incluir **`*`** como um parâmetro de caminho curinga. É possível definir parâmetros de caminho a partir do atributo `HttpParameters` do destino.
  - Método HTTP: selecione o método HTTP usado ao invocar o endpoint.
  - (Opcional) Limite de taxa de invocação por segundo: insira o número máximo de invocações aceitas por segundo para este destino.

    Este valor deve ser maior que zero. Por padrão, este valor é definido como 300.
  - Conexão: escolha usar uma conexão nova ou existente:
    - Para usar uma conexão existente, selecione Usar uma conexão existente e selecione a conexão na lista suspensa.
    - Para criar uma nova conexão para esse destino, selecione Criar uma nova conexão e defina o nome, o tipo de destino e o tipo de autorização da conexão. Também é possível adicionar uma descrição opcional para essa conexão.

Para obter mais informações, consulte ???.

AWS service (Serviço da AWS)

Para usar um AWS service (Serviço da AWS), selecione AWS service (Serviço da AWS) e faça o seguinte:

1. Em Selecionar um destino, selecione um AWS service (Serviço da AWS) para usar como destino. Forneça as informações solicitadas para o serviço selecionado.

> **Note**
>
> Os campos exibidos variam de acordo com o serviço selecionado. Para obter mais informações sobre os destinos disponíveis, consulte Alvos disponíveis no EventBridge console.

2. Para muitos tipos de destino, o Eventbridge precisa de permissões para enviar eventos ao destino. Nesses casos, o Eventbridge pode criar a função do IAM necessária para sua função ser executada.

   Em Perfil de execução, realize um dos seguintes procedimentos:

   - Para criar um novo perfil de execução para esta regra:
     a. Selecione Criar um novo perfil para este recurso específico.
     b. Insira um nome para este perfil de execução ou use o nome gerado pelo EventBridge.
   - Para usar um perfil de execução existente para esta regra:
     a. Selecione Usar perfil existente.
     b. Insira ou selecione o nome do perfil de execução a ser usado na lista suspensa.

3. (Opcional) Para Configurações adicionais, especifique qualquer uma das configurações opcionais disponíveis para seu tipo de destino:

   Event bus

   Em Fila de mensagens não entregues, escolha se será usada uma fila padrão do Amazon SQS como uma fila de mensagens não entregues. O EventBridge envia eventos que correspondem a essa regra para a fila de mensagens mortas se eles não forem entregues com êxito ao destino. Faça um dos seguintes procedimentos:

   - Escolha None (Nenhum) para não usar uma fila de mensagens mortas.
   - Escolha Selecionar uma fila do Amazon SQS na conta atual da AWS para usar como a fila de mensagens não entregues e depois selecione na lista suspensa a fila a ser usada.
   - Escolha Selecione uma fila do Amazon SQS em outra conta da AWS como fila de mensagens não entregues e insira o ARN da fila a ser usada. Você deve anexar uma

política baseada em recurso à fila que conceda permissão ao EventBridge para enviar mensagens a ela.

Para obter mais informações, consulte Como conceder permissões para a fila de mensagens não entregues.

API destination

1. (Opcional) Em Configurar entrada de destino, escolha como deseja personalizar o texto enviado ao destino para eventos correspondentes. Escolha uma das seguintes opções:
   - Eventos correspondentes: o EventBridge envia todo o evento de origem original para o destino. Esse é o padrão.
   - Parte dos eventos correspondentes: o EventBridge envia apenas a parte especificada do evento de origem original para o destino.

     Em Especificar a parte do evento correspondente, especifique um caminho JSON que defina a parte do evento que deseja que o EventBridge envie para o destino.
   - Constante (texto JSON): o EventBridge envia somente o texto JSON especificado para o destino. Nenhuma parte do evento de origem original é enviada.

     Em Especificar a constante em JSON, especifique o texto JSON que deseja que o EventBridge envie para o destino em vez do evento.
   - Transformador de entrada: configure um transformador de entrada para personalizar o texto que deseja que o EventBridge envie para o destino. Para obter mais informações, consulte ???.

     a. Selecione Configurar transformador de entrada.

     b. Configure o transformador de entrada seguindo as etapas em ???.
2. (Opcional) Em Política de repetição, especifique como o EventBridge deve tentar enviar novamente um evento para um destino após a ocorrência de um erro.
   - Idade máxima do evento: Insira o tempo máximo (em horas, minutos e segundos) para que o EventBridge retenha eventos não processados. O padrão é 24 horas.
   - Tentativas de repetição: insira o número máximo de vezes que o EventBridge deve tentar enviar novamente um evento para o destino após a ocorrência de um erro. O padrão é 185 vezes.

3. Em Fila de mensagens não entregues, escolha se será usada uma fila padrão do Amazon SQS como uma fila de mensagens não entregues. O EventBridge envia eventos que correspondem a essa regra para a fila de mensagens mortas se eles não forem entregues com êxito ao destino. Faça um dos seguintes procedimentos:
   - Escolha None (Nenhum) para não usar uma fila de mensagens mortas.
   - Escolha Selecionar uma fila do Amazon SQS na conta atual da AWS para usar como a fila de mensagens não entregues e depois selecione na lista suspensa a fila a ser usada.
   - Escolha Selecione uma fila do Amazon SQS em outra conta da AWS como fila de mensagens não entregues e insira o ARN da fila a ser usada. Você deve anexar uma política baseada em recurso à fila que conceda permissão ao EventBridge para enviar mensagens a ela.

     Para obter mais informações, consulte Como conceder permissões para a fila de mensagens não entregues.

AWS service

Observe que o EventBridge pode não exibir todos os campos a seguir para um determinado serviço da AWS.

1. (Opcional) Em Configurar entrada de destino, escolha como deseja personalizar o texto enviado ao destino para eventos correspondentes. Escolha uma das seguintes opções:
   - Eventos correspondentes: o EventBridge envia todo o evento de origem original para o destino. Esse é o padrão.
   - Parte dos eventos correspondentes: o EventBridge envia apenas a parte especificada do evento de origem original para o destino.

     Em Especificar a parte do evento correspondente, especifique um caminho JSON que defina a parte do evento que deseja que o EventBridge envie para o destino.
   - Constante (texto JSON): o EventBridge envia somente o texto JSON especificado para o destino. Nenhuma parte do evento de origem original é enviada.

     Em Especificar a constante em JSON, especifique o texto JSON que deseja que o EventBridge envie para o destino em vez do evento.

- Transformador de entrada: configure um transformador de entrada para personalizar o texto que deseja que o EventBridge envie para o destino. Para obter mais informações, consulte ???.

  a. Selecione Configurar transformador de entrada.

  b. Configure o transformador de entrada seguindo as etapas em ???.

2. (Opcional) Em Política de repetição, especifique como o EventBridge deve tentar enviar novamente um evento para um destino após a ocorrência de um erro.

   - Idade máxima do evento: Insira o tempo máximo (em horas, minutos e segundos) para que o EventBridge retenha eventos não processados. O padrão é 24 horas.
   - Tentativas de repetição: insira o número máximo de vezes que o EventBridge deve tentar enviar novamente um evento para o destino após a ocorrência de um erro. O padrão é 185 vezes.

3. Em Fila de mensagens não entregues, escolha se será usada uma fila padrão do Amazon SQS como uma fila de mensagens não entregues. O EventBridge envia eventos que correspondem a essa regra para a fila de mensagens mortas se eles não forem entregues com êxito ao destino. Faça um dos seguintes procedimentos:

   - Escolha None (Nenhum) para não usar uma fila de mensagens mortas.
   - Escolha Selecionar uma fila do Amazon SQS na conta atual da AWS para usar como a fila de mensagens não entregues e depois selecione na lista suspensa a fila a ser usada.
   - Escolha Selecione uma fila do Amazon SQS em outra conta da AWS como fila de mensagens não entregues e insira o ARN da fila a ser usada. Você deve anexar uma política baseada em recurso à fila que conceda permissão ao EventBridge para enviar mensagens a ela.

     Para obter mais informações, consulte Como conceder permissões para a fila de mensagens não entregues.

4. (Opcional) Selecione Add another target (Adicionar outro destino) para adicionar outro destino a essa regra.
5. Escolha Next (Próximo).

Observe que o EventBridge pode não exibir todos os campos a seguir para um determinado serviço da AWS.

Configurar tags e regra de revisão

Por fim, insira as tags desejadas para a regra, revise e crie a regra.

Para configurar tags, revisar e criar a regra

1. (Opcional) Insira uma ou mais tags para a regra. Para obter mais informações, consulte EventBridge Etiquetas da Amazon.
2. Escolha Next (Próximo).
3. Analise os detalhes da nova regra. Para fazer mudanças a qualquer seção, escolha o botão Editar próximo à seção.

   Quando estiver satisfeito com os detalhes da regra, escolha Criar regra.

Como usar o Agendador do Amazon EventBridge com o Amazon EventBridge

O Agendador do Amazon EventBridge é um agendador sem servidor que permite criar, executar e gerenciar tarefas de um serviço gerenciado central. Com o Agendador do EventBridge, você pode criar programações usando expressões cron e rate para padrões recorrentes ou configurar invocações únicas. É possível configurar janelas de tempo flexíveis para entrega, bem como definir limites de repetição e o tempo máximo de retenção para invocações de API com falha.

O Agendador do EventBridge é altamente personalizável e oferece escalabilidade aprimorada em relação às Regras programadas do EventBridge, com um conjunto mais amplo de operações da API de destino e serviços da AWS. Recomendamos que você use o Agendador do EventBridge para invocar alvos em uma programação.

Tópicos

Configurar o perfil de execução

Quando você cria uma nova programação, o Agendador EventBridge deve ter permissão para invocar sua operação de API de destino em seu nome. Você concede essas permissões ao Agendador EventBridge usando um perfil de execução. A política de permissão que você anexa ao perfil de execução da sua programação define as permissões necessárias. Essas permissões dependem da API de destino que você deseja que o Agendador EventBridge invoque.

Quando você usa o console do Agendador EventBridge para criar uma programação, como no procedimento a seguir, o Agendador EventBridge configura automaticamente um perfil de execução com base no destino selecionado. Se você quiser criar uma agenda usando um dos SDKs do Agendador EventBridge, a AWS CLI ou o AWS CloudFormation, você deve ter um perfil de execução existente que conceda as permissões que o Agendador EventBridge exige para invocar um alvo. Para obter mais informações sobre como configurar manualmente um perfil de execução para sua programação, consulte Setting up an execution role no Guia do usuário do Agendador EventBridge.

Criar uma programação

Criar uma programação usando o console

1. Abra o console do Agendador do Amazon EventBridge em https://console.aws.amazon.com/scheduler/home.
2. Na página Programações, escolha Criar uma programação.
3. Na página Especificar detalhes da programação, na seção Nome e descrição da programação, faça o seguinte:

   a. Em Nome da programação, insira um nome para a programação. Por exemplo, **`MyTestSchedule`**.

   b. (Opcional) Em Descrição, insira uma descrição para a sua programação. Por exemplo, **`My first schedule`**.

   c. Para Grupo de programação, escolha um grupo de programação na lista suspensa. Se você não tiver um grupo, escolha padrão. Para criar um grupo de programação, escolha criar sua própria programação.

   Você usa grupos de programação para adicionar tags a eles.

4. • Escolha suas opções de programação.

Ocorrência	Fazer isso...	
Programação única Uma programação única invoca um destino somente uma vez na data e hora que você especificar.	Para Data e hora, faça o seguinte: • Insira uma data válida no formato `YYYY/MM/DD` . • Insira um carimbo de data e hora no formato 24 horas, `hh:mm`. • Para Fuso horário, escolha o fuso horário.	
Programação recorrente	a. Em Tipo de cronograma, siga um destes procedimentos:	

Ocorrência	Fazer isso...	
Uma programação recorrente invoca uma meta em uma taxa que você especifica usando uma expressão cron ou expressão de taxa.	• Para usar uma expressão cron para definir o cronograma, escolha Cronograma baseado em cron e insira a expressão cron. • Para usar uma expressão rate para definir o cronograma, escolha Cronograma com base em rate e insira a expressão rate. Para obter mais informações sobre expressões cron e rate, consulte Schedule types on EventBridge Scheduler no Guia do usuário do Agendador do Amazon EventBridge. b. Para Janela de tempo flexível, escolha Desativado para desativar a opção ou escolher uma das janelas de tempo predefinidas. Por exemplo, se você escolher 15 minutos e	

Ocorrência	Fazer isso...	
	definir uma programação recorrente para invocar sua meta uma vez a cada hora, a programaç ão será executada em até 15 minutos após o início de cada hora.	

5. (Opcional) Se você escolher Programação recorrente na etapa anterior, na seção Período, faça o seguinte:

   a. Para Fuso horário, escolha um fuso horário.

   b. Para Data e hora de início, insira uma data válida no formato `YYYY/MM/DD` e, em seguida, especifique um carimbo de data e hora no formato 24 horas, `hh:mm`.

   c. Para Data e hora de término, insira uma data válida no formato `YYYY/MM/DD` e, em seguida, especifique um carimbo de data e hora no formato 24 horas, `hh:mm`.

6. Escolha Next (Próximo).

7. Na página Selecionar destino, escolha a operação de API da AWS que o Agendador do EventBridge invoca:

   a. Em API de destino, escolha Destinos modelados.

   b. Escolha Amazon EventBridge PutEvents.

   c. Em PutEvents, especifique o seguinte:

      - No barramento de eventos do EventBridge, escolha o barramento do evento no menu suspenso. Por exemplo, **`default`**.

        Também é possível criar um novo barramento de eventos no console do EventBridge escolhendo Criar novo barramento de eventos.

      - Em Tipo de detalhe, insira o tipo de detalhe dos eventos que deseja combinar. Por exemplo, **`Object Created`**.

      - Em Origem, insira o nome do serviço que é a origem dos eventos.

        Para eventos de serviço da AWS, especifique o prefixo do serviço como origem. Não inclua o prefixo `aws.`. Por exemplo, para eventos do Amazon S3, insira **`s3`**.

Para determinar o prefixo de um serviço, consulte A tabela de chaves de condição na Referência de autorização de serviço. Para obter mais informações sobre valores de eventos de tipo de origem e detalhe, consulte ???.

- (Opcional): Em Detalhe, insira um padrão de evento para filtrar ainda mais os eventos que o Agendador do EventBridge envia para o EventBridge.

  Para obter mais informações, consulte ???.

8. Escolha Next (Próximo).
9. Na página Settings (Configurações), faça o seguinte:

   a. Para ativar a programação, em Estado da programação, alterne para Ativar programação.

   b. Para configurar uma política de novas tentativas para a sua programação, em Política de novas tentativas e fila de mensagens não entregues (DLQ), faça o seguinte:

      - Alterne para Tentar novamente.
      - Para Idade máxima do evento, insira as horas e os minutos máximos que o Agendador do EventBridge deve manter um evento não processado.
      - O tempo máximo é de 24 horas.
      - Para Máximo de tentativas, insira o número máximo de vezes que o Agendador do EventBridge tentará novamente a programação se o destino retornar um erro.

        O valor máximo é 185 novas tentativas.

      Com políticas de novas tentativas, se uma programação falhar em invocar seu destino, o Agendador do EventBridge executará novamente a programação. Se configurado, você deve definir o tempo máximo de retenção e as novas tentativas para a programação.

   c. Escolha onde o Agendador do EventBridge armazena os eventos não entregues.

Opção Filas de mensagens não entregues (DLQ)	Fazer isso...
Não armazene	Selecione None.
Armazenar o evento na mesma Conta da AWS em	a. Escolha Selecione uma fila do Amazon SQS

Opção Filas de mensagens não entregues (DLQ)	Fazer isso...	
que você está criando a programação	na minha Conta da AWScomo uma DLQ. b. Escolha o nome do recurso da Amazon (ARN) da fila do Amazon SQS.	
Armazenar o evento em uma Conta da AWS diferente de onde você está criando a programação	a. Escolha Especifique uma fila do Amazon SQS em outras Contas da AWS como uma DLQ. b. Insira o nome do recurso da Amazon (ARN) da fila do Amazon SQS.	

d. Para usar uma chave gerenciada pelo cliente para criptografar sua entrada de destino, em Criptografia, escolha Personalizar as configurações de criptografia (avançado).

   Se você escolher essa opção, insira uma chave ARN do KMS existente ou escolha Criar um AWS KMS key para navegar até o console do AWS KMS. Para obter mais informações sobre como o Agendador do EventBridge criptografa seus dados em repouso, consulte Encryption at rest no Guia do usuário do Agendador do Amazon EventBridge.

e. Para que o Agendador do EventBridge crie um novo perfil de execução para você, escolha Criar um novo perfil para esta programação. Depois, insira um nome para Nome do perfil. Se você escolher essa opção, o Agendador do EventBridge anexará as permissões necessárias para seu destino de exemplo ao perfil.

10. Escolha Next (Próximo).
11. Na página Revisar e criar uma programação, revise os detalhes da sua programação. Em cada seção, escolha Editar para voltar a essa etapa e editar seus detalhes.
12. Escolha Criar programação.

    Você pode ver uma lista das suas programações novas e existentes na página Programações. Na coluna Status, verifique se sua nova programação está Ativada.

Recursos relacionados

Para obter mais informações sobre o Agendador do EventBridge, consulte o seguinte:

- Guia do usuário do Agendador do EventBridge
- Referência da API do Agendador do EventBridge
- Preços do Agendador do EventBridge

Como criar uma regra do Amazon EventBridge que é executada de acordo com uma programação.

Uma regra pode ser executada em resposta a um evento ou em determinados intervalos de tempo. Por exemplo, para executar periodicamente uma função do AWS Lambda, é possível criar uma regra para ser executada de acordo com uma agenda.

> **Note**
>
> O EventBridge oferece o Agendador do Amazon EventBridge, um agendador com tecnologia sem servidor que permite criar, executar e gerenciar tarefas de um serviço gerenciado central. O Agendador do EventBridge é altamente personalizável e oferece escalabilidade aprimorada em relação às regras programadas do EventBridge, com um conjunto mais amplo de operações de API de destino e serviços da AWS.
> Recomendamos que você use o Agendador do EventBridge para invocar alvos em uma programação. Para obter mais informações, consulte ???.

No EventBridge, é possível criar dois tipos de regras programadas:

- Regras que são executadas a uma taxa regular

  O EventBridge executa essas regras em intervalos regulares; por exemplo, a cada 20 minutos.

  Para especificar a taxa de uma regra programada, você define uma expressão rate.
- Regras que são executadas em horários específicos

  O EventBridge executa essas regras em horários e datas específicos; por exemplo, 8:00 da manhã PST na primeira segunda-feira de cada mês.

Para especificar a hora e as datas em que uma regra programada é executada, você define uma expressão cron.

As expressões rate são mais simples de definir, enquanto as expressões cron oferecem controle detalhado do cronograma. Por exemplo, com uma expressão cron, é possível definir uma regra que é executada em um horário especificado em um determinado dia de cada semana ou mês. Por outro lado, as expressões rate executam uma regra em uma frequência regular, como uma vez por hora ou uma vez por dia.

Todos os eventos programados usam o fuso horário UTC+0 e a precisão mínima para uma programação é de um minuto.

> **Note**
>
> O EventBridge não fornece precisão no segundo nível em expressões de programação. A melhor resolução ao usar uma expressão cron é um minuto. Por conta da natureza distribuída do EventBridge e aos serviços de destino, pode haver um atraso de diversos segundos entre o momento em que a regra programada é acionada e o momento em que o serviço de destino executa o recurso de destino.

O seguinte vídeo fornece uma visão geral das tarefas de programação: Como criar tarefas programadas com o EventBridge

Tópicos

- Criar uma regra que seja executada em uma programação
- Referência de expressões cron
- Referência de expressões rate

Criar uma regra que seja executada em uma programação

As etapas a seguir explicam como criar uma regra do EventBridge que seja executada regularmente.

> **Note**
>
> Só é possível criar regras programadas usando o barramento de eventos padrão.

Etapas

- Defina a regra
- Defina a programação
- Selecione destinos
- Configurar tags e regra de revisão

Defina a regra

Primeiro, insira um nome e uma descrição para a regra para identificá-la.

Para definir os detalhes da regra

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Informe um Nome para a regra e, opcionalmente, uma Descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra regra na mesma Região da AWS e no mesmo barramento de eventos.
5. Em Selecionar barramento de eventos, escolha o barramento de eventos padrão. Só é possível criar regras programadas usando o barramento de eventos padrão.
6. Para que a regra entre em vigor assim que você a criar, verifique se a opção Habilitar a regra no barramento de eventos selecionado está ativada.
7. Em Rule type (Tipo de regra), escolha Schedule (Programação).

   Neste momento, é possível continuar com a criação de uma regra que é executada de acordo com uma programação ou usar o Agendador do Amazon EventBridge.
8. Escolha como deseja continuar:

   - Use o EventBridge Scheduler para criar sua agenda

> **Note**
>
> O Agendador do EventBridge é um programador com tecnologia sem servidor que permite criar, executar e gerenciar tarefas de um serviço gerenciado central. Ele fornece uma funcionalidade de programação única e recorrente, independente das regras e barramentos de eventos. O Agendador do EventBridge é altamente personalizável e oferece escalabilidade aprimorada em relação às regras programadas do EventBridge, com um conjunto mais amplo de operações de API de destino e serviços da AWS.
> Recomendamos que você use o Agendador do EventBridge para invocar alvos em uma programação. Para obter mais informações, consulte O que é o Agendador do Amazon EventBridge? no Manual do usuário do Agendador do Amazon EventBridge.

1. Selecione Continuar no Agendador do EventBridge

   O EventBridge abre o console do Agendador do EventBridge na página Criar agendamento.
2. Crie a programação no console do Agendador do EventBridge.

- Continue usando o EventBridge para criar uma regra programada para o barramento de eventos padrão
  1. Selecione Continuar para criar a regra.

Defina a programação

Em seguida, defina o padrão de programação.

Para definir o padrão de programação

1. Em Padrão de programação, escolha se deseja que a programação seja executada em um horário específico ou em uma taxa normal:

   Specific time

   1. Escolha um cronograma refinado que seja executado em um horário específico, tal como 8:00 a.m. PST na primeira segunda-feira de cada mês.
   2. Para a expressão Cron, especifique os campos para definir a expressão cron que o EventBridge deve usar para determinar quando executar essa regra programada.

Depois de especificar todos os campos, o EventBridge exibirá as próximas dez datas em que o EventBridge executará essa regra programada. Também é possível escolher se deseja exibir essas datas em UTC ou no Fuso horário local.

Para obter mais informações sobre como criar uma expressão cron, consulte ???.

Regular rate

1. Escolha Uma programação que seja executada em uma taxa regular, como a cada 10 minutos.
2. Em Expressão rate, especifique os campos Valor e Unidade para definir a taxa na qual o EventBridge deve executar essa regra programada.

   Para obter mais informações sobre como criar uma expressão rate, consulte ???.

2. Escolha Next (Próximo).

Selecione destinos

Escolha um ou mais destinos para receber eventos que correspondam ao padrão especificado. Os destinos podem incluir um barramento de eventos do EventBridge, destinos da API do EventBridge, incluindo parceiros de SaaS, como Salesforce, ou outros AWS service (Serviço da AWS).

Para selecionar destinos

1. Para Tipo de destino, escolha um dos seguintes tipos de destino:

   Event bus

   Para selecionar um barramento de eventos do EventBridge, selecione o Barramento de eventos do EventBridge e faça o seguinte:

   - Para usar um barramento de eventos da mesma forma Região da AWS que esta regra:
     1. Selecione Barramento de eventos na mesma conta e região.
     2. Em Barramento de eventos para destino, escolha a caixa suspensa e insira o nome do barramento de eventos. Também é possível selecionar o barramento de eventos na lista suspensa.

        Para obter mais informações, consulte ???.

- Para usar um barramento de eventos em uma conta ou Região da AWS diferente como esta regra:
  1. Selecione Barramento de eventos em uma conta ou região diferente.
  2. Para Barramento de eventos como destino, insira o ARN do barramento de eventos que deseja usar.

     Para obter mais informações, consulte:

     - ???
     - ???

API destination

Para usar um destino da API EventBridge, selecione o Destino da API do EventBridge e faça o seguinte:

- Para usar um destino de API existente, selecione Usar um destino de API existente. Em seguida, selecione um destino de API na lista suspensa.
- Para criar um novo destino de API, selecione Criar um novo destino de API. Em seguida, forneça os seguintes detalhes para o destino:
  - Nome: insira um nome para o destino.

    Os nomes devem ser exclusivos em sua Conta da AWS. Nomes podem ter até 64 caracteres. Os caracteres válidos A-Z, a-z, 0-9 e . _ - (hífen).
  - (Opcional) Descrição: insira uma descrição para o destino.

    As descrições podem ter até 512 caracteres.
  - Endpoint de destino da API: o endpoint de URL para o destino.

    O URL do endpoint deve começar com **`https`**. É possível incluir **`*`** como um parâmetro de caminho curinga. É possível definir parâmetros de caminho a partir do atributo `HttpParameters` do destino.
  - Método HTTP: selecione o método HTTP usado ao invocar o endpoint.
  - (Opcional) Limite de taxa de invocação por segundo: insira o número máximo de invocações aceitas por segundo para este destino.

    Este valor deve ser maior que zero. Por padrão, este valor é definido como 300.

- Conexão: escolha usar uma conexão nova ou existente:
  - Para usar uma conexão existente, selecione Usar uma conexão existente e selecione a conexão na lista suspensa.
  - Para criar uma nova conexão para esse destino, selecione Criar uma nova conexão e defina o nome, o tipo de destino e o tipo de autorização da conexão. Também é possível adicionar uma descrição opcional para essa conexão.

Para obter mais informações, consulte ???.

AWS service (Serviço da AWS)

Para usar um AWS service (Serviço da AWS), selecione AWS service (Serviço da AWS) e faça o seguinte:

1. Em Selecionar um destino, selecione um AWS service (Serviço da AWS) para usar como destino. Forneça as informações solicitadas para o serviço selecionado.

> **Note**
>
> Os campos exibidos variam de acordo com o serviço selecionado. Para obter mais informações sobre os destinos disponíveis, consulte Alvos disponíveis no EventBridge console.

2. Para muitos tipos de destino, o Eventbridge precisa de permissões para enviar eventos ao destino. Nesses casos, o Eventbridge pode criar a função do IAM necessária para sua função ser executada.

   Em Perfil de execução, realize um dos seguintes procedimentos:

   - Para criar um novo perfil de execução para esta regra:
     a. Selecione Criar um novo perfil para este recurso específico.
     b. Insira um nome para este perfil de execução ou use o nome gerado pelo EventBridge.
   - Para usar um perfil de execução existente para esta regra:
     a. Selecione Usar perfil existente.
     b. Insira ou selecione o nome do perfil de execução a ser usado na lista suspensa.

3. (Opcional) Para Configurações adicionais, especifique qualquer uma das configurações opcionais disponíveis para seu tipo de destino:

   Event bus

   Em Fila de mensagens não entregues, escolha se será usada uma fila padrão do Amazon SQS como uma fila de mensagens não entregues. O EventBridge envia eventos que correspondem a essa regra para a fila de mensagens mortas se eles não forem entregues com êxito ao destino. Faça um dos seguintes procedimentos:

   - Escolha None (Nenhum) para não usar uma fila de mensagens mortas.
   - Escolha Selecionar uma fila do Amazon SQS na conta atual da AWS para usar como a fila de mensagens não entregues e depois selecione na lista suspensa a fila a ser usada.
   - Escolha Selecione uma fila do Amazon SQS em outra conta da AWS como fila de mensagens não entregues e insira o ARN da fila a ser usada. Você deve anexar uma política baseada em recurso à fila que conceda permissão ao EventBridge para enviar mensagens a ela.

     Para obter mais informações, consulte Como conceder permissões para a fila de mensagens não entregues.

   API destination

   1. (Opcional) Em Configurar entrada de destino, escolha como deseja personalizar o texto enviado ao destino para eventos correspondentes. Escolha uma das seguintes opções:
      - Eventos correspondentes: o EventBridge envia todo o evento de origem original para o destino. Esse é o padrão.
      - Parte dos eventos correspondentes: o EventBridge envia apenas a parte especificada do evento de origem original para o destino.

        Em Especificar a parte do evento correspondente, especifique um caminho JSON que defina a parte do evento que deseja que o EventBridge envie para o destino.
      - Constante (texto JSON): o EventBridge envia somente o texto JSON especificado para o destino. Nenhuma parte do evento de origem original é enviada.

        Em Especificar a constante em JSON, especifique o texto JSON que deseja que o EventBridge envie para o destino em vez do evento.

- Transformador de entrada: configure um transformador de entrada para personalizar o texto que deseja que o EventBridge envie para o destino. Para obter mais informações, consulte ???.

  a. Selecione Configurar transformador de entrada.

  b. Configure o transformador de entrada seguindo as etapas em ???.

2. (Opcional) Em Política de repetição, especifique como o EventBridge deve tentar enviar novamente um evento para um destino após a ocorrência de um erro.
   - Idade máxima do evento: Insira o tempo máximo (em horas, minutos e segundos) para que o EventBridge retenha eventos não processados. O padrão é 24 horas.
   - Tentativas de repetição: insira o número máximo de vezes que o EventBridge deve tentar enviar novamente um evento para o destino após a ocorrência de um erro. O padrão é 185 vezes.
3. Em Fila de mensagens não entregues, escolha se será usada uma fila padrão do Amazon SQS como uma fila de mensagens não entregues. O EventBridge envia eventos que correspondem a essa regra para a fila de mensagens mortas se eles não forem entregues com êxito ao destino. Faça um dos seguintes procedimentos:
   - Escolha None (Nenhum) para não usar uma fila de mensagens mortas.
   - Escolha Selecionar uma fila do Amazon SQS na conta atual da AWS para usar como a fila de mensagens não entregues e depois selecione na lista suspensa a fila a ser usada.
   - Escolha Selecione uma fila do Amazon SQS em outra conta da AWS como fila de mensagens não entregues e insira o ARN da fila a ser usada. Você deve anexar uma política baseada em recurso à fila que conceda permissão ao EventBridge para enviar mensagens a ela.

     Para obter mais informações, consulte Como conceder permissões para a fila de mensagens não entregues.

AWS service

Observe que o EventBridge pode não exibir todos os campos a seguir para um determinado serviço da AWS.

1. (Opcional) Em Configurar entrada de destino, escolha como deseja personalizar o texto enviado ao destino para eventos correspondentes. Escolha uma das seguintes opções:

- Eventos correspondentes: o EventBridge envia todo o evento de origem original para o destino. Esse é o padrão.
- Parte dos eventos correspondentes: o EventBridge envia apenas a parte especificada do evento de origem original para o destino.

  Em Especificar a parte do evento correspondente, especifique um caminho JSON que defina a parte do evento que deseja que o EventBridge envie para o destino.
- Constante (texto JSON): o EventBridge envia somente o texto JSON especificado para o destino. Nenhuma parte do evento de origem original é enviada.

  Em Especificar a constante em JSON, especifique o texto JSON que deseja que o EventBridge envie para o destino em vez do evento.
- Transformador de entrada: configure um transformador de entrada para personalizar o texto que deseja que o EventBridge envie para o destino. Para obter mais informações, consulte ???.

  a. Selecione Configurar transformador de entrada.

  b. Configure o transformador de entrada seguindo as etapas em ???.

2. (Opcional) Em Política de repetição, especifique como o EventBridge deve tentar enviar novamente um evento para um destino após a ocorrência de um erro.
   - Idade máxima do evento: Insira o tempo máximo (em horas, minutos e segundos) para que o EventBridge retenha eventos não processados. O padrão é 24 horas.
   - Tentativas de repetição: insira o número máximo de vezes que o EventBridge deve tentar enviar novamente um evento para o destino após a ocorrência de um erro. O padrão é 185 vezes.
3. Em Fila de mensagens não entregues, escolha se será usada uma fila padrão do Amazon SQS como uma fila de mensagens não entregues. O EventBridge envia eventos que correspondem a essa regra para a fila de mensagens mortas se eles não forem entregues com êxito ao destino. Faça um dos seguintes procedimentos:
   - Escolha None (Nenhum) para não usar uma fila de mensagens mortas.
   - Escolha Selecionar uma fila do Amazon SQS na conta atual da AWS para usar como a fila de mensagens não entregues e depois selecione na lista suspensa a fila a ser usada.
   - Escolha Selecione uma fila do Amazon SQS em outra conta da AWS como fila de mensagens não entregues e insira o ARN da fila a ser usada. Você deve anexar uma

política baseada em recurso à fila que conceda permissão ao EventBridge para enviar mensagens a ela.

Para obter mais informações, consulte Como conceder permissões para a fila de mensagens não entregues.

4. (Opcional) Selecione Add another target (Adicionar outro destino) para adicionar outro destino a essa regra.
5. Escolha Next (Próximo).

Configurar tags e regra de revisão

Por fim, insira as tags desejadas para a regra, revise e crie a regra.

Para configurar tags, revisar e criar a regra

1. (Opcional) Insira uma ou mais tags para a regra. Para obter mais informações, consulte EventBridge Etiquetas da Amazon.
2. Escolha Next (Próximo).
3. Analise os detalhes da nova regra. Para fazer mudanças a qualquer seção, escolha o botão Editar próximo à seção.

   Quando estiver satisfeito com os detalhes da regra, escolha Criar regra.

Referência de expressões cron

Expressões cron têm seis campos obrigatórios, que são separados por um espaço em branco.

Sintaxe

```
cron(fields)
```

Campo	Valores	Wildcards (Curingas)
minutos	0-59	, - * /
Horas	0-23	, - * /

Campo	Valores	Wildcards (Curingas)
Dia do mês	1-31	, - * ? / L W
Mês	1-12 ou JAN-DEZ	, - * /
Dia da semana	1-7 ou DOM-SÁB	, - * ? L #
Ano	1970-2199	, - * /

Curingas

- A , (vírgula) curinga inclui valores adicionais. No campo Mês, JAN, FEV, MAR inclui janeiro, fevereiro e março.
- O - (traço) curinga especifica intervalos. No campo Dia, 1-15 inclui os dias 1 a 15 do mês especificado.
- O * (asterisco) curinga inclui todos os valores no campo. No campo Hours (Horas), * inclui todas as horas. Não é possível usar * nos campos Dia do mês e Dia da semana. Se você usá-lo em um deles, utilize ? no outro.
- O curinga / (barra) especifica incrementos. No campo Minutos, você pode inserir 1/10 para especificar cada décimo minuto a partir do primeiro minuto da hora (por exemplo, o 11º, 21º e 31º minuto, etc.).
- O curinga ? (interrogação) especifica qualquer um. No campo Dia do mês, você pode inserir 7 e qualquer dia da semana for aceitável, pode inserir ? no campo 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 seria 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.

> **Note**
>
> Se você usar um caractere “#”, poderá definir apenas uma expressão no campo do dia da semana. Por exemplo, o valor `"3#1,6#3"` não é válido porque é interpretado como duas expressões.

Limitações

- Você não pode especificar os campos Dia do mês e Dia da semana na mesma expressão cron. Se especificar um valor ou * (asterisco) em um dos campos, deverá usar ? (ponto de interrogação) no outro.
- As expressões Cron que levam a taxas mais rápidas do que 1 minuto não têm suporte.

Exemplos

Você pode usar as seguintes sequências de caracteres cron de exemplo ao criar uma regra com programação.

Minutos	Horas	Dia do mês	Mês	Dia da semana	Ano	Significado
0	10	*	*	?	*	Executada às 10h00 (UTC+0) todos os dias
15	12	*	*	?	*	Executada às 12h15 (UTC+0) todos os dias
0	18	?	*	SEG-SEX	*	Executada às 18h (UTC

Minutos	Horas	Dia do mês	Mês	Dia da semana	Ano	Significado
						+0) de segunda a sexta
0	8	1	*	?	*	Executada às 8h (UTC +0) todo primeiro dia do mês
0/15	*	*	*	?	*	Executada a cada 15 minutos
0/10	*	?	*	SEG-SEX	*	Executada a cada 10 minutos de segunda a sexta
0/5	8-17	?	*	SEG-SEX	*	Executada a cada cinco minutos, de segunda a sexta, entre 8h e 17h55 (UTC+0)

Minutos	Horas	Dia do mês	Mês	Dia da semana	Ano	Significado
0/30	20-2	?	*	SEG-SEX	*	Executada a cada 30 minutos, de segunda a sexta-feira, das 22h do dia inicial às 2h do dia seguinte (UTC) Executada das 12h às 2h na manhã de segunda-feira (UTC).

O seguinte exemplo cria uma regra que é executada todos os dias às 12h UTC+0.

```
aws events put-rule --schedule-expression "cron(0 12 * * ? *)" --name MyRule1
```

O seguinte exemplo cria uma regra que é executada todos os dias, às 14h05 e 14h35 UTC+0.

```
aws events put-rule --schedule-expression "cron(5,35 14 * * ? *)" --name MyRule2
```

O exemplo a seguir cria uma regra executada às 10h15 UTC+0 na última sexta-feira de cada mês durante os anos de 2019 a 2022.

```
aws events put-rule --schedule-expression "cron(15 10 ? * 6L 2019-2022)" --name MyRule3
```

Referência de expressões rate

Uma expressão rate começa quando a regra de evento programado é criada e é executada em uma programação definida.

As expressões rate têm dois campos obrigatórios separados por um espaço em branco.

Sintaxe

```
rate(value unit)
```

value

Um número positivo.

unidade

A unidade de tempo. Diferentes unidades são necessárias para valores de 1, como `minute`, e valores acima de 1, como `minutes`.

Valores válidos: minuto | minutos | hora | horas | dia | dias

Limitações

Se o valor for igual a 1, a unidade deverá ser singular. Se o valor for maior que 1, a unidade deverá ser plural. Por exemplo, as taxas (uma hora) e (cinco horas) não são válidas, mas as taxas (uma hora) e (cinco horas) são válidas.

Exemplos

Os exemplos a seguir mostram como usar expressões rate com o comando da AWS CLI `put-rule`. O primeiro exemplo aciona a regra a cada minuto, o segundo exemplo aciona a regra a cada cinco minutos, o próximo a aciona uma vez por hora e o terceiro exemplo a aciona uma vez por dia.

```
aws events put-rule --schedule-expression "rate(1 minute)" --name MyRule2
```

```
aws events put-rule --schedule-expression "rate(5 minutes)" --name MyRule3
```

```
aws events put-rule --schedule-expression "rate(1 hour)" --name MyRule4
```

```
aws events put-rule --schedule-expression "rate(1 day)" --name MyRule5
```

Como desabilitar ou excluir uma regra do Amazon EventBridge

Para impedir que uma regra processe eventos ou seja executada de acordo com uma agenda, é possível excluir ou desativar a regra. As etapas a seguir explicam como excluir ou desativar uma regra do EventBridge.

Para excluir uma desativar uma regra

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).

   Em Event bus (Barramento de eventos), selecione o barramento de eventos associado à regra.
3. Faça um dos seguintes procedimentos:

   a. Para excluir uma regra, selecione o botão próximo à regra e escolha Ações, Excluir, Excluir.

      Se a regra for uma regra gerenciada, insira o nome da regra para confirmar que se trata de uma regra gerenciada e que a exclusão pode interromper a funcionalidade no serviço que criou a regra. Para continuar, insira o nome da regra e selecione Force delete (Forçar exclusão).

   b. Para desabilitar temporariamente uma regra, selecione o botão ao lado da regra e escolha Disable (Desabilitar), Disable (Desabilitar).

      Não é possível desabilitar uma regra gerenciada.

Práticas recomendadas ao definir regras do Amazon EventBridge

Abaixo estão algumas das práticas recomendadas a serem consideradas ao criar regras para seus barramentos de eventos.

Defina um único destino para cada regra

Embora você possa especificar até cinco destinos para uma determinada regra, o gerenciamento de regras é mais fácil quando você especifica um único destino para cada regra. Se mais de um destino precisar receber o mesmo conjunto de eventos, recomendamos duplicar a regra para entregar os mesmos eventos a destino diferentes. Este encapsulamento simplifica a manutenção das regras: se as necessidades dos destinos do evento divergirem com o tempo, você poderá atualizar cada regra e seu padrão de evento independentemente das outras.

Configurar permissões de regras

É possível permitir que componentes ou serviços de aplicações que consomem eventos tenham o controle do gerenciamento de suas próprias regras. Uma abordagem arquitetônica comum adotada pelos clientes é isolar esses componentes ou serviços da aplicação usando contas separadas da AWS. Para habilitar o fluxo de eventos de uma conta para outra, é preciso criar uma regra em um barramento de eventos que encaminhe eventos para um barramento de eventos em outra conta. É possível permitir que equipes ou serviços de aplicações que consomem eventos tenham o controle do gerenciamento de suas próprias regras. Isso é feito ao especificar as permissões adequadas para as contas por meio de políticas de recursos. Isto funciona em todas as contas e regiões.

Para obter mais informações, consulte ???.

Por exemplo de políticas de recursos, consulte Padrões de design de várias contas com o Amazon EventBridge no GitHub.

Melhor desempenho de regras

Monitore suas regras para garantir que elas estejam funcionando conforme o esperado:

- Monitorar a métrica `TriggeredRules` em busca de pontos de dados ausentes ou anomalias pode ajudá-lo a detectar discrepâncias em um publicador que fez uma alteração significativa. Para obter mais informações, consulte ???.
- O alarme sobre anomalias ou a contagem máxima esperada também pode ajudar a detectar quando uma regra está correspondendo a novos eventos. Isto pode acontecer quando publicadores de eventos, incluindo serviços da AWS e parceiros de SaaS, introduzem novos eventos ao habilitar novos casos de uso e atributos. Quando esses novos eventos são inesperados e levam a um volume maior do que a taxa de processamento do destino downstream, podem resultar em um acúmulo de eventos.

  Este processamento de eventos inesperados também pode levar a cobranças de cobrança indesejadas.

  Também pode acionar o controle de utilização regras quando a conta ultrapassa sua cota de serviço de destino e agregada de invocações por segundo. O EventBridge ainda tentará entregar eventos compatíveis com regras limitadas e tentar novamente por até 24 horas ou conforme descrito na política de repetição personalizada do destino. É possível detectar e alarmar regras limitadas usando a métrica `ThrottledRules`

- Para casos de uso de baixa latência, também é possível monitorar o uso da latência usando `IngestionToInvocationStartLatency`, que fornece uma indicação da integridade do seu barramento de eventos. Qualquer período prolongado de alta latência acima de 30 segundos pode indicar uma interrupção do serviço ou controle de utilização de regras.

Como o Amazon EventBridge e os modelos do AWS Serverless Application Model

Também é possível criar e testar regras manualmente no console do EventBridge, o que pode ajudar no processo de desenvolvimento à medida que você refina os padrões de eventos. No entanto, quando estiver tudo pronto para implantar sua aplicação, será mais fácil usar uma estrutura como a de AWS SAM para lançar todos os seus recursos com tecnologia sem servidor de forma consistente.

Será usado esta aplicação de exemplo para analisar como é possível usar modelos do AWS SAM para criar recursos do EventBridge. O arquivo template.yaml neste exemplo é um modelo do AWS SAM que define quatro funções do AWS Lambda e mostra duas maneiras diferentes de integrar as funções do Lambda com o EventBridge.

Para obter uma explicação passo a passo dessa aplicação de exemplo, consulte ???.

Existem duas abordagens para se usar o EventBridge e os modelos do AWS SAM. Para integrações simples em que uma função do Lambda é invocada por uma regra, a abordagem de Modelo combinado é recomendada. Se existe uma lógica de roteamento complexa ou está se conectando a recursos fora do seu modelo do AWS SAM, a abordagem modelo separado é a melhor escolha.

Abordagens:

- Modelo combinado
- Modelo separado

Modelo combinado

A primeira abordagem usa a propriedade `Events` para configurar a regra do EventBridge. O código de exemplo a seguir define um evento que invoca sua função do Lambda.

> **Note**
>
> Este exemplo cria automaticamente a regra no barramento de eventos padrão, que existe em todas as contas da AWS. Para associar a regra a um barramento de eventos personalizado, você pode adicionar o `EventBusName` ao modelo.

```
atmConsumerCase3Fn:
```

```
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: atmConsumer/
    Handler: handler.case3Handler
    Runtime: nodejs12.x
    Events:
      Trigger:
        Type: CloudWatchEvent
        Properties:
          Pattern:
            source:
              - custom.myATMapp
            detail-type:
              - transaction
            detail:
              result:
                - "anything-but": "approved"
```

Este código YAML é equivalente a um padrão de evento no console do EventBridge. No YAML, só é preciso definir o padrão do evento e o AWS SAM cria automaticamente um perfil do IAM com as permissões necessárias.

Modelo separado

Na segunda abordagem para definir uma configuração do EventBridge no AWS SAM, os recursos são separados com mais clareza no modelo.

1. Primeiro, defina a função do Lambda:

   ```
   atmConsumerCase1Fn:
     Type: AWS::Serverless::Function
     Properties:
       CodeUri: atmConsumer/
       Handler: handler.case1Handler
       Runtime: nodejs12.x
   ```

2. Em seguida, defina a regra usando um recurso da `AWS::Events::Rule`. As propriedades definem o padrão do evento e também podem especificar destinos. É possível definir explicitamente vários destinos.

   ```
   EventRuleCase1:
     Type: AWS::Events::Rule
   ```

```
  Properties:
    Description: "Approved transactions"
    EventPattern:
      source:
        - "custom.myATMapp"
      detail-type:
        - transaction
      detail:
        result:
          - "approved"
    State: "ENABLED"
    Targets:
      -
        Arn:
          Fn::GetAtt:
            - "atmConsumerCase1Fn"
            - "Arn"
        Id: "atmConsumerTarget1"
```

3. Por fim, defina um recurso `AWS::Lambda::Permission` que conceda permissão ao EventBridge para invocar o destino.

```
PermissionForEventsToInvokeLambda:
  Type: AWS::Lambda::Permission
  Properties:
    FunctionName:
      Ref: "atmConsumerCase1Fn"
    Action: "lambda:InvokeFunction"
    Principal: "events.amazonaws.com"
    SourceArn:
      Fn::GetAtt:
        - "EventRuleCase1"
        - "Arn"
```

Gerar um modelo do AWS CloudFormation com as regras do Amazon EventBridge

O AWS CloudFormation permite configurar e gerenciar recursos da AWS entre contas e regiões de forma centralizada e repetível tratando a infraestrutura como código. O CloudFormation faz isso permitindo que você crie modelos que definem os recursos que deseja provisionar e gerenciar.

O EventBridge permite gerar modelos com as regras existentes na conta, o que ajuda você a começar a desenvolver modelos do CloudFormation. É possível selecionar uma única regra ou várias regras para incluir no modelo. É possível utilizar esses modelos como base para criar pilhas de recursos sob o gerenciamento do CloudFormation.

Para obter mais informações sobre o CloudFormation, consulte o Guia do usuário do AWS CloudFormation.

> **Note**
>
> O EventBridge não inclui regras gerenciadas no modelo gerado.

Também é possível gerar um modelo de barramento de eventos existente, incluindo as regras contidas no barramento de eventos.

Como gerar um modelo do AWS CloudFormation com uma ou mais regras

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Em Selecionar barramento de eventos, escolha o barramento de eventos que contém as regras que deseja incluir no modelo.
4. Em Regras, escolha as regras a serem incluídas no modelo do AWS CloudFormation gerado.

   Para uma única regra, também é possível escolher o nome da regra para exibir a página de detalhes da regra.
5. Escolha Modelo do CloudFormation e escolha em qual formato você deseja que o EventBridge gere o modelo: JSON ou YAML.

   O EventBridge exibe o modelo, gerado no formato selecionado.
6. O EventBridge oferece a opção de baixar o arquivo de modelo ou copiar o modelo para a área de transferência.

   - Para baixar o arquivo de modelo, escolha Baixar.
   - Para copiar o modelo para a área de transferência, escolha Copiar.
7. Para sair do modelo, escolha Cancelar.

Depois de personalizar o modelo do AWS CloudFormation conforme necessário para o caso de uso, será possível utilizá-lo para criar pilhas no AWS CloudFormation.

Considerações ao utilizar modelos do CloudFormation gerados no Amazon EventBridge

Considere os seguintes fatores ao utilizar um modelo do CloudFormation gerado no EventBridge:

- O EventBridge não inclui nenhuma senha no modelo gerado.

  É possível editar o modelo para incluir parâmetros de modelo que permitam aos usuários especificar senhas ou outras informações sensitivas ao utilizar o modelo para criar ou atualizar uma pilha do CloudFormation.

  Além disso, os usuários podem utilizar o Secrets Manager para criar um segredo na região desejada e editá-lo para utilizar parâmetros dinâmicos.
- Os destinos no modelo gerado permanecem exatamente como foram especificados no barramento de eventos original. Isso poderá resultar em problemas entre regiões se você não editar adequadamente o modelo antes de utilizá-lo para criar pilhas em outras regiões.

  Além disso, o modelo gerado não cria os destinos downstream automaticamente.

EventBridge Metas da Amazon

Um destino é um recurso ou endpoint que EventBridge envia um evento para quando o evento corresponde ao padrão de evento definido para uma regra. A regra processa os dados do evento e envia as informações pertinentes ao destino. Para entregar dados do evento a um alvo, EventBridge precisa de permissão para acessar o recurso de destino. É possível definir até cinco destinos para cada regra.

Quando destinos novos são adicionados a uma regra, e essa regra associada é executada logo em seguida, destinos novos ou atualizados podem não ser invocados imediatamente. Permita um curto período para que as alterações entrem em vigor.

O seguinte vídeo aborda as noções básicas dos destinos: O que é um destino

Alvos disponíveis no EventBridge console

Você pode configurar os seguintes alvos para eventos no EventBridge console:

- Destino da API
- API Gateway
- AWS AppSync;
- Fila de trabalhos em lote
- CloudWatch grupo de registros
- CodeBuild projeto
- CodePipeline
- Chamada de API `CreateSnapshot` do Amazon EBS
- EC2 Image Builder
- Chamada de API `RebootInstances` do EC2
- Chamada de API `StopInstances` do EC2
- Chamada de API `TerminateInstances` do EC2
- Tarefa do ECS
- Barramento de eventos em uma conta ou região diferente

- Barramento de eventos na mesma conta e região
- Fluxo de entrega do Firehose
- Fluxo de trabalho do Glue
- Plano de resposta do Incident Manager
- Modelo de avaliação do Inspector
- Fluxo do Kinesis
- Função do Lambda (ASYNC)
- Consultas de API de dados do cluster do Amazon Redshift
- Consultas de API de dados de grupos de trabalho do Amazon Redshift sem servidor
- SageMaker Pipeline
- Tópico do Amazon SNS

  EventBridge não oferece suporte aos tópicos FIFO (primeiro a entrar, primeiro a sair) do Amazon SNS.
- Fila do Amazon SQS
- Máquina de estado do Step Functions (ASYNC)
- Automação do Systems Manager
- Systems Manager OpsItem
- Run Command do Systems Manager

Parâmetros de destino

Alguns destinos não enviam as informações da carga do evento para o destino. Em vez disso, eles tratam o evento como um gatilho para invocar uma API específica. EventBridge usa os parâmetros do Target para determinar o que acontece com esse alvo. Incluindo o seguinte:

- Destinos da API (os dados enviados para o destino da API devem corresponder à estrutura da API. É preciso usar o objeto `InputTransformer` para garantir que os dados sejam estruturados corretamente. Se quiser incluir a carga original do evento, faça referência a ela no `InputTransformer`.)
- API Gateway (os dados enviados para o API Gateway devem corresponder à estrutura da API. É preciso usar o objeto `InputTransformer` para garantir que os dados sejam

estruturados corretamente. Se quiser incluir a carga original do evento, faça referência a ela no `InputTransformer`.)

- Amazon EC2 Image Builder
- `RedshiftDataParameters` (Clusters da API de dados do Amazon Redshift)
- `SageMakerPipelineParameters`(Amazon SageMaker Runtime Model Building Pipelines)

> **Note**
>
> EventBridge não suporta toda a sintaxe do JSON Path e a avalia em tempo de execução. A sintaxe compatível inclui:
>
> - notação de pontos (por exemplo, `$.detail`)
> - traços
> - sublinhados
> - caracteres alfanuméricos
> - índices de matriz
> - curingas (*)

Parâmetros dinâmicos do caminho

Alguns parâmetros de destino são compatíveis com a sintaxe de caminho JSON dinâmico opcional. Esta sintaxe permite especificar caminhos JSON em vez de valores estáticos (por exemplo`$.detail.state`). O valor inteiro precisa ser um caminho JSON, não apenas parte dele. Por exemplo, `RedshiftParameters.Sql` pode ser `$.detail.state`, mas não pode ser `"SELECT * FROM $.detail.state"`. Estes caminhos são substituídos dinamicamente em runtime por dados da própria carga do evento no caminho especificado. Os parâmetros do caminho dinâmico não podem referenciar valores novos ou transformados resultantes da transformação de entrada. A sintaxe compatível com caminhos JSON de parâmetros dinâmicos é a mesma da transformação da entrada. Para mais informações, consulte ???.

A sintaxe dinâmica pode ser usada em todos os campos de string, não enumerados, desses parâmetros:

- `EcsParameters`
- `HttpParameters` (exceto chaves `HeaderParameters`)

- `RedshiftDataParameters`
- `SageMakerPipelineParameters`

Permissões

Para fazer chamadas de API nos recursos que você possui, é EventBridge necessária a permissão apropriada. Para recursos do Amazon SNS AWS Lambda e do Amazon, EventBridge usa políticas baseadas em recursos. Para instâncias do EC2, fluxos de dados do Kinesis e máquinas de estado Step Functions EventBridge , usa funções do IAM que você especifica no parâmetro em`RoleARN`. `PutTargets` É possível invocar um endpoint do API Gateway com autorização do IAM configurada, mas o perfil é opcional se você não tiver configurado a autorização. Para ter mais informações, consulte Amazon EventBridge e AWS Identity and Access Management.

Se outra conta estiver na mesma região e tiver concedido permissão para você, será possível enviar eventos para essa conta. Para ter mais informações, consulte Enviar e receber EventBridge eventos da Amazon entre AWS contas.

Se seu destino estiver criptografado, deverá incluir a seção a seguir em sua política de chave do KMS.

```
{
    "Sid": "Allow EventBridge to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": "events.amazonaws.com"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*"
}
```

EventBridge especificidades do alvo

AWS Batch filas de trabalho

Alguns parâmetros AWS Batch `submitJob` podem ser configurados via BatchParameters.

Outros podem ser especificados na carga útil do evento. Se a carga útil do evento (transmitida ou via InputTransformers) contiver as seguintes chaves, elas serão mapeadas para os parâmetros de `submitJob` solicitação:

- `ContainerOverrides: containerOverrides`

  > **Note**
  >
  > Isto inclui somente comando, ambiente, memória e vcpus

- `DependsOn: dependsOn`

  > **Note**
  >
  > Isto inclui apenas jobId

- `Parameters: parameters`

CloudWatch Grupo de registros

Se você não usar um InputTransformercom um destino de CloudWatch registros, a carga útil do evento será usada como a mensagem de registro e a origem do evento como o carimbo de data/hora. Se você usar um InputTransformer, o modelo deverá ser:

```
{"timestamp":<timestamp>,"message":<message>}
```

EventBridge agrupa as entradas enviadas para um fluxo de log; portanto, EventBridge pode entregar um ou vários eventos a um stream de log, dependendo do tráfego.

CodeBuild projeto

Se você usar InputTransformerspara moldar o evento de entrada em um Target para corresponder à CodeBuild StartBuildRequestestrutura, os parâmetros serão mapeados de 1 para 1 e transmitidos para. `codeBuild.StartBuild`

Tarefa do Amazon ECS

Se você usar InputTransformerspara moldar o evento de entrada para um Target de acordo com a RunTask TaskOverrideestrutura do Amazon ECS, os parâmetros serão mapeados de 1 para 1 e transmitidos para. `ecs.RunTask`

Plano de resposta do Incident Manager

Se o evento correspondente veio de CloudWatch Alarmes, os detalhes da alteração do estado do alarme serão preenchidos nos detalhes do gatilho da StartIncidentRequest chamada para o Incident Manager.

Configurar destinos

Saiba como definir as configurações dos EventBridge alvos.

Destinos:

- Destinos da API
- EventBridge Metas da Amazon para o Amazon API Gateway
- AWS AppSync metas para a Amazon EventBridge
- Conexões para destinos de endpoint HTTP
- Enviar e receber EventBridge eventos da Amazon entre AWS contas
- Enviando e recebendo EventBridge eventos da Amazon entre AWS regiões
- Enviar e receber EventBridge eventos da Amazon entre ônibus de eventos na mesma conta e região

Destinos da API

Os destinos de EventBridge API da Amazon são endpoints HTTP que você pode invocar como alvo de uma regra, semelhante à forma como você invoca um AWS serviço ou recurso como destino. Usando destinos de API, você pode rotear eventos entre AWS serviços, aplicativos integrados de software como serviço (SaaS) e seus aplicativos externos AWS usando chamadas de API. Quando você especifica um destino de API como destino de uma regra, EventBridge invoca o endpoint HTTP para qualquer evento que corresponda ao padrão de evento especificado na regra e, em seguida, entrega as informações do evento com a solicitação. Com EventBridge, você pode usar qualquer método HTTP, exceto CONNECT e TRACE para a solicitação. Os métodos HTTP mais comuns a serem usados são PUT e POST. Também é possível usar transformadores de entrada para personalizar o evento de acordo com os parâmetros de um endpoint HTTP específico. Para ter mais informações, consulte Transformação EventBridge de insumos da Amazon.

> **Note**
>
> Os destinos de API não oferecem suporte a destinos privados, como endpoints VPC de interface, incluindo APIs HTTPS privadas em nuvens privadas virtuais (VPC) usando Network and Application Load Balancer e VPC endpoints de interface.
> Para ter mais informações, consulte ???.

> **Important**
>
> EventBridge as solicitações para um endpoint de destino da API devem ter um tempo limite máximo de execução do cliente de 5 segundos. Se o endpoint de destino levar mais de 5 segundos para responder, o EventBridge tempo limite da solicitação será atingido. EventBridge as novas tentativas atingiram o tempo limite das solicitações até os máximos configurados em sua política de repetição. Por padrão, os máximos são 24 horas e 185 vezes. Após o número máximo de tentativas, os eventos são enviados para sua fila de mensagens não entregues, se você tiver uma. Caso contrário, o evento será descartado.

O seguinte vídeo demonstra o uso do destino da API: Como usar destinos de API

Neste tópico:

- Criar um destino de API
- Como criar regras que enviam eventos para um destino de API
- Perfil vinculado ao serviço para destinos de API
- Cabeçalhos em solicitações para destinos da API
- Códigos de erro de destino da API
- Como a taxa de invocação afeta a entrega do evento
- Envio de CloudEvents eventos para destinos de API
- Parceiros de destino da API

Criar um destino de API

Cada destino de API requer uma conexão. Uma conexão determina o tipo de autorização e as credenciais a serem utilizadas para autorização com um endpoint HTTP de destino de API. Também é possível escolher uma conexão existente ou criar uma conexão ao criar um destino de API. Para mais informações, consulte ???.

Para criar um destino de API usando o EventBridge console

1. Faça login AWS usando uma conta que tenha permissões para gerenciar EventBridge e abrir o EventBridgeconsole.
2. No painel de navegação à esquerda, escolha Destinos da API.
3. Role para baixo até a tabela de destinos da API e escolha Criar destino da API.
4. Na página Criar destino da API, insira um Nome para o destino da API. É possível usar até 64 letras maiúsculas ou minúsculas, números, caracteres de ponto (.), traço (-) ou sublinhado (_).

   O nome deve ser exclusivo na sua conta na sua região atual.
5. Insira uma Descrição para o destino da API.
6. Insira um Endpoint de destino da API para o destino da API. O endpoint de destino da API é um destino de endpoint de invocação HTTP para eventos. As informações de autorização que você inclui na conexão usada para esse destino de API são usadas para autorizar esse endpoint. O URL deve usar HTTPS.
7. Insira o método HTTP a ser usado para se conectar ao endpoint de destino da API.
8. (Opcional) No campo Limite de taxa de invocação por segundo, insira o número máximo de invocações por segundo a serem enviadas ao endpoint de destino da API.

O limite de taxa definido pode afetar a forma como os eventos EventBridge são entregues. Para ter mais informações, consulte Como a taxa de invocação afeta a entrega do evento.

9. Em Conexão, execute um dos seguintes procedimentos:
   - Escolha Usar uma conexão existente e selecione a conexão a ser usada para esse destino de API.
   - Escolha Criar uma nova conexão e insira os detalhes da conexão a ser criada. Para obter mais informações, consulte Conexões.
10. Selecione Create (Criar).

Como criar regras que enviam eventos para um destino de API

Depois de criar um destino de API, é possível selecioná-lo como destino de uma regra. Para usar um destino de API como destino, você deve fornecer um perfil do IAM com as permissões corretas. Para mais informações, consulte ???.

Selecionar um destino de API como um destino faz parte da criação da regra.

Para criar uma regra que envia eventos para um destino de API usando o console

1. Siga as etapas no procedimento ???.
2. Na ??? etapa, quando solicitado a escolher um destino de API como tipo de destino:
   a. Selecione o destino EventBridge da API.
   b. Execute um destes procedimentos:
      - Escolha Usar um destino de API existente e selecione um destino de API existente
      - Escolha Criar um novo destino de API e especifique a configuração necessária para definir seu novo destino de API.

        Para obter mais informações sobre como especificar as configurações necessárias, consulte???.
   c. (Opcional): Para especificar parâmetros de cabeçalho para o evento, em Parâmetros de cabeçalho, escolha Adicionar parâmetro de cabeçalho.

      Em seguida, especifique a chave e o valor do parâmetro do cabeçalho.

d. (Opcional): Para especificar os parâmetros da sequência de caracteres de consulta para o evento, em Parâmetros da sequência de consulta, escolha Adicionar parâmetro da sequência de caracteres de consulta.

   Em seguida, especifique a chave e o valor para o parâmetro da sequência de caracteres de consulta.

3. Conclua a criação da regra seguindo as etapas do procedimento.

Perfil vinculado ao serviço para destinos de API

Quando você cria uma conexão para um destino de API, uma função vinculada ao serviço chamada AWS ServiceRoleForAmazonEventBridgeApiDestinationsé adicionada à sua conta. EventBridge usa a função vinculada ao serviço para criar e armazenar um segredo no Secrets Manager. Para conceder as permissões necessárias à função vinculada ao serviço, EventBridge anexe a AmazonEventBridgeApiDestinationsServiceRolePolicypolítica à função. A política limita as permissões concedidas somente às necessárias para que o perfil interaja com o segredo da conexão. Nenhuma outra permissão está incluída, e o perfil só pode interagir com as conexões em sua conta para gerenciar o segredo.

A política a seguir é a `AmazonEventBridgeApiDestinationsServiceRolePolicy`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:UpdateSecret",
                "secretsmanager:DescribeSecret",
                "secretsmanager:DeleteSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:events!connection/*"
        }
    ]
}
```

Para obter mais informações sobre os perfis vinculados ao serviço, consulte Como usar perfis vinculados ao serviço na documentação do IAM.

A função `AmazonEventBridgeApiDestinationsServiceRolePolicy` vinculada ao serviço é suportada nas seguintes regiões: AWS

- 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 (Mumbai)
- 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)
- Europa (Paris)
- Europa (Estocolmo)
- América do Sul (São Paulo)
- China (Ningxia)
- China (Pequim)

Cabeçalhos em solicitações para destinos da API

A seção a seguir detalha como EventBridge manipula cabeçalhos HTTP em solicitações para destinos de API.

Cabeçalhos incluídos nas solicitações para destinos de API

Além dos cabeçalhos de autorização definidos para a conexão usada para um destino de API, EventBridge inclui os seguintes cabeçalhos em cada solicitação.

Chave de cabeçalho	Valor de cabeçalho
User-Agent	Amazon//EventBridgeApiDestinations
Content-Type	Se nenhum valor personalizado de Content-Type for especificado, EventBridge inclua o seguinte valor padrão como Content-Type: application/json; charset=utf-8
Intervalo	bytes=0-1048575
Accept-Encoding	gzip,deflate
Conexão	feche
Content-Length	Um cabeçalho de entidade que indica o tamanho do corpo da entidade, em bytes, enviado ao destinatário.
Host	Um cabeçalho de solicitação que especifica o host e o número da porta do servidor para o qual a solicitação está sendo enviada.

Cabeçalhos que não podem ser substituídos em solicitações para destinos de API

EventBridge não permite que você substitua os seguintes cabeçalhos:

- User-Agent
- Intervalo

Os cabeçalhos são EventBridge removidos das solicitações para os destinos da API

EventBridge remove os seguintes cabeçalhos para todas as solicitações de destino da API:

- A-IM
- Accept-Charset
- Accept-Datetime
- Accept-Encoding
- Cache-Control
- Conexão
- Content-Encoding
- Content-Length
- Conteúdo-MD5
- Data
- Expect
- Encaminhado
- De
- Host
- HTTP2-Settings
- If-Match
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Max-Forwards
- Origem
- Pragma
- Proxy-Authorization
- Intervalo
- Referer
- TE
- Trailer

- Transfer-Encoding
- User-Agent
- Upgrade
- Via
- Aviso

Códigos de erro de destino da API

Quando EventBridge tenta entregar um evento para um destino de API e ocorre um erro, EventBridge faça o seguinte:

- Eventos associados aos códigos de erro 409, 429 e 5xx são repetidos.
- Eventos associados aos códigos de erro 1xx, 2xx, 3xx e 4xx (excluindo 429) não são repetidos.

EventBridge Os destinos da API leem o cabeçalho de resposta HTTP padrão `Retry-After` para descobrir quanto tempo esperar antes de fazer uma solicitação de acompanhamento. EventBridge escolhe o valor mais conservador entre a política de repetição definida e o `Retry-After` cabeçalho. Se `Retry-After` o valor for negativo, EventBridge interrompe a tentativa de entrega desse evento.

Como a taxa de invocação afeta a entrega do evento

Se a taxa de invocação for definida por segundo para um valor muito menor do que o número de invocações geradas, os eventos podem não ser entregues dentro do tempo de repetição de 24 horas para eventos. Por exemplo, se definir a taxa de invocação para 10 invocações por segundo, mas milhares de eventos por segundo forem gerados, você rapidamente terá um acúmulo de eventos para entregar que excede 24 horas. Para garantir que nenhum evento seja perdido, configure uma fila de mensagens não entregues para enviar eventos com invocações com falha para que seja possível processar os eventos posteriormente. Para ter mais informações, consulte Usando filas de cartas mortas para processar eventos não entregues.

Envio de CloudEvents eventos para destinos de API

CloudEvents é uma especificação independente de fornecedor para formatação de eventos, com o objetivo de fornecer interoperabilidade entre serviços, plataformas e sistemas. Você pode usar EventBridge para transformar eventos de AWS serviço CloudEvents antes de serem enviados para um destino, como um destino de API.

> **Note**
>
> O procedimento a seguir explica como transformar eventos de origem em modo CloudEventsestruturado. Na CloudEvents especificação, uma mensagem de modo estruturado é aquela em que todo o evento (atributos e dados) é codificado na carga útil do evento.

Para obter mais informações sobre a CloudEvents especificação, consulte cloudevents.io.

Para transformar AWS eventos no CloudEvents formato usando o console

Para transformar eventos no CloudEvents formato antes da entrega a um destino, você começa criando uma regra de barramento de eventos. Como parte da definição da regra, você usa um transformador de entrada para ter eventos de EventBridge transformação antes de enviar para o destino especificado.

1. Siga as etapas no procedimento ???.
2. Na ??? etapa, quando solicitado a escolher um destino de API como tipo de destino:
   a. Selecione o destino EventBridge da API.
   b. Execute um destes procedimentos:
      - Escolha Usar um destino de API existente e selecione um destino de API existente
      - Escolha Criar um novo destino de API e especifique a configuração necessária para definir seu novo destino de API.

        Para obter mais informações sobre como especificar as configurações necessárias, consulte???.
   c. Especifique os parâmetros de cabeçalho Content-Type necessários para os CloudEvents eventos:
      - Em Parâmetros de cabeçalho, escolha Adicionar parâmetro de cabeçalho.
      - Para chave, especifique`Content-Type`.

        Para valor, especifique`application/cloudevents+json; charset=UTF-8`.
3. Especifique uma função de execução para seu alvo.

4. Defina um transformador de entrada para transformar os dados do evento de origem no CloudEvents formato:

   a. Em Configurações adicionais, para Configurar entrada de destino, escolha Transformador de entrada.

      Em seguida, escolha Configurar transformador de entrada.

   b. Em Transformador de entrada de destino, especifique o caminho de entrada.

      No caminho de entrada abaixo, o atributo region é um atributo de extensão personalizado do CloudEvents formato. Como tal, não é necessário para cumprir a CloudEvents especificação.

      CloudEvents permite usar e criar atributos de extensão não definidos na especificação principal. Para obter mais informações, incluindo uma lista de atributos de extensão conhecidos, consulte Atributos de CloudEvents extensão na documentação de CloudEvents especificação em GitHub.

      ```
      {
        "detail": "$.detail",
        "detail-type": "$.detail-type",
        "id": "$.id",
        "region": "$.region",
        "source": "$.source",
        "time": "$.time"
      }
      ```

   c. Em Modelo, insira o modelo para transformar os dados do evento de origem no CloudEvents formato.

      No modelo abaixo, não `region` é estritamente obrigatório, pois o `region` atributo no caminho de entrada é um atributo de extensão da CloudEvents especificação.

      ```
      {
        "specversion":"1.0",
        "id":<id>,
        "source":<source>,
        "type":<detail-type>,
        "time":<time>,
        "region":<region>,
        "data":<detail>
      ```

```
}
```

5. Conclua a criação da regra seguindo as etapas do procedimento.

Parceiros de destino da API

Use as informações fornecidas pelos seguintes AWS parceiros para configurar um destino de API e uma conexão para seu serviço ou aplicativo.

Observabilidade da nuvem da Cisco

URL do endpoint de invocação de destino da API:

```
https://tenantName.observe.appdynamics.com/rest/awsevents/aws-eventbridge-integration/endpoint
```

Tipos de autorização compatíveis:

Credenciais de cliente OAuth

Os tokens OAuth são atualizados quando uma resposta 401 ou 407 é retornada

Parâmetros adicionais de autorização necessários:

ID AppDynamics do cliente Cisco e segredo do cliente

Endpoint OAuth:

```
https://tenantName.observe.appdynamics.com/auth/tenantId/default/oauth2/token
```

Os seguintes parâmetros do par de chave/valor do OAuth:

Tipo	Chave	Valor
Campo corporal	grant_type	client_credentials
Cabeçalho	Content-Type	aplicativo/x-www-form-urlencoded; charset=utf-8

AppDynamics Documentação da Cisco:

AWS ingestão de eventos

Operações de API comumente usadas:

Não aplicável

Informações adicionais

A escolha AppDynamics da Cisco no menu suspenso Destinos do parceiro preenche previamente as informações necessárias do OAuth, incluindo os pares de chave/valor do cabeçalho e do corpo necessários para chamadas de API.

Para obter informações adicionais, consulte a ingestão de AWS eventos na AppDynamics documentação da Cisco.

Confluente

URL do endpoint de invocação de destino da API:

Normalmente, o seguinte formato:

`https://random-id.region.aws.confluent.cloud:443/kafka/v3/clusters/cluster-id/topics/topic-name/records`

Para obter mais informações, consulte Encontre o endereço do endpoint REST e o ID do cluster na documentação do Confluent.

Tipos de autorização compatíveis:

Basic

Parâmetros adicionais de autorização necessários:

Não aplicável

Documentação confluente:

Produzir registros

Proxy REST confluente para Apache Kafka

Operações de API comumente usadas:

`POST`

Informações adicionais

Para transformar os dados do evento em uma mensagem que o endpoint possa processar, crie um transformador de entrada de destino.

- Para gerar um registro sem especificar uma chave de particionamento do Kafka, use o modelo a seguir para seu transformador de entrada. Nenhum caminho de entrada é necessário.

```
{
  "value":{
      "type":"JSON",
      "data":aws.events.event.json
  },
}
```

- Para gerar um registro usando um campo de dados do evento como chave de particionamento do Kafka, siga o caminho de entrada e o exemplo do modelo abaixo. Este exemplo define o caminho de entrada para o `orderId` campo e, em seguida, especifica esse campo como a chave de partição.

  Primeiro, defina o caminho de entrada para o campo de dados do evento:

```
{
  "orderId":"$.detail.orderId"
}
```

  Em seguida, use o modelo do transformador de entrada para especificar o campo de dados como a chave de partição:

```
{
  "value":{
    "type":"JSON",
    "data":aws.events.event.json
  },
  "key":{
    "data":"<orderId>",
    "type":"STRING"
  }
}
```

Coralogix

URL do endpoint de invocação de destino da API

Para obter uma lista completa de endpoints, consulte Referência da API do Coralogix.

Tipos de autorização compatíveis

Chave de API

Parâmetros adicionais de autorização necessários

Cabeçalho `"x-amz-event-bridge-access-key"`, o valor é a chave da API Coralogix

Documentação da Coralogix

EventBridgeAutenticação da Amazon

Operações de API comumente usadas

EUA: https://ingress.coralogix.us/aws/event-bridge

Cingapura: https://ingress.coralogixsg.com/aws/event-bridge

Irlanda: https://ingress.coralogix.com/aws/event-bridge

Estocolmo: https://ingress.eu2.coralogix.com/aws/event-bridge

Índia: https://ingress.coralogix.in/aws/event-bridge

Informações adicionais

Os eventos são armazenados como entradas de log com `applicationName=[AWS Account]` e `subsystemName=[event.source]`.

Datadog

URL do endpoint de invocação de destino da API

Para obter uma lista completa de endpoints, consulte Referência da API do Datadog.

Tipos de autorização compatíveis

Chave de API

Parâmetros adicionais de autorização necessários

Nenhum

Documentação da Datadog

Autenticação

Operações de API comumente usadas

POST https://api.datadoghq.com/api/v1/events

POST https://http-intake.logs.datadoghq.com/v1/input

Informações adicionais

Os URLs de endpoint diferem dependendo da localização da sua organização do Datadog. Para obter o URL correto para sua organização, consulte a documentação

Freshworks

URL do endpoint de invocação de destino da API

Para obter uma lista de endpoints, consulte https://developers.freshworks.com/documentation/

Tipos de autorização compatíveis

Básico, chave de API

Parâmetros adicionais de autorização necessários

Não aplicável

Documentação da Freshworks

Autenticação

Operações de API comumente usadas

https://developers.freshdesk.com/api/#create_ticket

https://developers.freshdesk.com/api/#update_ticket

https://developer.freshsales.io/api/#create_lead

https://developer.freshsales.io/api/#update_lead

Informações adicionais

Nenhum

MongoDB

URL do endpoint de invocação de destino da API

https://data.mongodb-api.com/app/*App ID*/endpoint/

Tipos de autorização compatíveis

Chave de API

E-mail/senha

Autenticação JWT personalizada

Parâmetros adicionais de autorização necessários

Nenhum

Documentação da MongoDB

API de dados do Atlas

Endpoints

Endpoints HTTPS personalizados

Autenticação

Operações de API comumente usadas

Nenhum

Informações adicionais

Nenhum

New Relic

URL do endpoint de invocação de destino da API

Para obter mais informações, consulte Nossos datacenters nas regiões da UE e dos EUA.

Eventos

EUA: https://insights-collector.newrelic.com/v1/accounts/*YOUR_NEW_RELIC_ACCOUNT_ID*/events

UE: https://insights-collector.eu01.nr-data.net/v1/accounts/*YOUR_NEW_RELIC_ACCOUNT_ID*/events

Métricas

EUA: https://metric-api.newrelic.com/metric/v1

UE: https://metric-api.eu.newrelic.com/metric/v1

Logs

EUA: https://log-api.newrelic.com/log/v1

UE: https://log-api.eu.newrelic.com/log/v1

Rastreamentos

EUA: https://trace-api.newrelic.com/trace/v1

UE: https://trace-api.eu.newrelic.com/trace/v1

Tipos de autorização compatíveis

Chave de API

Documentação da New Relic

Métrica de API

API de eventoshttps://docs.newrelic.com/docs/telemetry-data-platform/ingest-manage-data/ingest-apis/introduction-event-api/

Registrar API

API de rastreamento

Operações de API comumente usadas

Métrica de API

API de eventoshttps://docs.newrelic.com/docs/telemetry-data-platform/ingest-manage-data/ingest-apis/introduction-event-api/

Registrar API

API de rastreamento

Informações adicionais

Limites métricos de API

Limites da API de eventos

Limites de registro da API

Limites de rastreamento de API

Operata

URL do endpoint de invocação de destino da API:

https://api.operata.io/v2/aws/events/contact-record

Tipos de autorização compatíveis:

Basic

Parâmetros adicionais de autorização necessários:

Nenhum

Documentação do Operata:

Como criar, visualizar, alterar e revogar tokens de API?

AWS Integração do Operata usando Amazon EventBridge Scheduler Pipes

Operações de API comumente usadas:

`POST https://api.operata.io/v2/aws/events/contact-record`

Informações adicionais

O `username` é o ID do grupo Operata e a senha é seu token de API.

Salesforce

URL do endpoint de invocação de destino da API

Assunto — https://myDomainName.my.salesforce.com/services/data/ *versionNumber / sobjects//* SobjectEndpoint*

Eventos de plataforma personalizados — *https://myDomainName.my.salesforce.com/services/data/versionNumber /sobjects/ /* customPlatformEndpoint*

Para obter uma lista completa de endpoints, consulte Referência da API do Salesforce

Tipos de autorização compatíveis

Credenciais de cliente OAuth

Tokens OAUTH são atualizados quando uma resposta 401 ou 407 é retornada.

Parâmetros adicionais de autorização necessários

ID do cliente da aplicação Salesforce conectada e segredo do cliente.

Um dos seguintes endpoints de autorização:

- Produção — https://*MyDomainName*.my.salesforce.com. /services/oauth2/token
- Sandbox sem domínios aprimorados — https://***MyDomainName***-- ***SandboxName***.my.salesforce.com/services /oauth2/token
- Sandbox com domínios aprimorados — https://***MyDomainName***-- .sandbox.my.salesforce.com/services/oauth2/token *SandboxName*

O seguinte par de chave/valor:

Chave	Valor
grant_type	client_credentials

Documentação da Salesforce

Guia do desenvolvedor da API REST

Operações de API comumente usadas

Como trabalhar com metadados de objeto

Trabalhar com registros

Informações adicionais

Para ver um tutorial explicando como usar o EventBridge console para criar uma conexãoSalesforce, um destino de API e uma regra para a qual encaminhar informaçõesSalesforce, consulte???.

Slack

URL do endpoint de invocação de destino da API

Para ver uma lista de endpoints e outros recursos, consulte Como usar a API Web do Slack

Tipos de autorização compatíveis

OAuth 2.0

Tokens OAUTH são atualizados quando uma resposta 401 ou 407 é retornada.

Ao criar uma aplicação Slack e a instala no seu espaço de trabalho, um token portador do OAuth será criado em seu nome para ser usado para autenticar chamadas pela sua conexão de destino da API.

Parâmetros adicionais de autorização necessários

Não aplicável

Documentação da Slack

Configuração básica da aplicação

Como instalar com OAuth

Como recuperar mensagens

Enviar mensagens

Como enviar mensagens usando webhooks de entrada

Operações de API comumente usadas

https://slack.com/api/chat.postMessage

Informações adicionais

Ao configurar sua EventBridge regra, há duas configurações a serem destacadas:

- Inclua um parâmetro de cabeçalho que defina o tipo de conteúdo como "application/json; charset=utf-8".
- Use um transformador de entrada para mapear o evento de entrada para a saída esperada para a API do Slack, ou seja, certifique-se de que a carga enviada para a API do Slack tenha pares de chave/valor de "pipe" e "texto".

Shopify

URL do endpoint de invocação de destino da API

Para obter uma lista de endpoints e outros recursos e métodos, consulte Endpoints e solicitações.

Tipos de autorização compatíveis

OAuth, chave de API

> **Note**
>
> Tokens OAUTH são atualizados quando uma resposta 401 ou 407 é retornada.

Parâmetros adicionais de autorização necessários

Não aplicável

Documentação da Shopify

Visão geral de autenticação e autorização

Operações de API comumente usadas

POST - /admin/api/2022-01/products.json

GET - admin/api/2022-01/products/{product_id}.json

PUT - admin/api/2022-01/products/{product_id}.json

EXCLUIR: admin/api/2022-01/products/{product_id}.json

Informações adicionais

Crie uma aplicação

Entrega de EventBridge webhook da Amazon

Tokens de acesso para aplicações personalizadas no administrador da Shopify

Produto

API de administração da Shopify

Splunk

URL do endpoint de invocação de destino da API

https://*SPLUNK_HEC_ENDPOINT*:*optional_port*/services/collector/raw

Tipos de autorização compatíveis

Básico, chave de API

Parâmetros adicionais de autorização necessários

Nenhum

Documentação da Splunk

Para os dois tipos de autorização, você precisa de um ID de token HEC. Para obter mais informações, consulte Configurar e usar o Coletor de eventos HTTP no Splunk Web.

Operações de API comumente usadas

POST https://*SPLUNK_HEC_ENDPOINT*:*optional_port*/services/collector/raw

Informações adicionais

Chave de API — Ao configurar o endpoint para EventBridge, o nome da chave da API é “Autorização” e o valor é o ID do token Splunk HEC.

Básico (nome de usuário/senha) — Ao configurar o endpoint para EventBridge, o nome de usuário é “Splunk” e a senha é o ID do token Splunk HEC.

Sumo Logic

URL do endpoint de invocação de destino da API

Os URLs do endpoint HTTP Log e Metric Source serão diferentes para cada usuário. Para obter mais informações, consulte Origem de logs e métricas HTTP.

Tipos de autorização compatíveis

O Sumo Logic não exige autenticação em suas origens HTTP porque há uma chave exclusiva embutida na URL. Por este motivo, é preciso tratar o URL como um segredo.

Quando você configura o destino EventBridge da API, é necessário um tipo de autorização. Para atender a esse requisito, selecione Chave de API e atribua a ela um nome de chave de "chave fictícia" e um valor de chave de "valor fictício".

Parâmetros adicionais de autorização necessários

Não aplicável

Documentação da Sumo Logic

Sumo Logicjá criou fontes hospedadas para coletar registros e métricas de vários AWS serviços e você pode usar as informações do site deles para trabalhar com essas fontes. Para obter mais informações, consulte Amazon Web Services.

Se você estiver gerando eventos personalizados a partir de um aplicativo e quiser enviá-los Sumo Logic como registros ou métricas, use destinos de EventBridge API e endpoints Sumo Logic HTTP Log and Metric Source.

- Para se inscrever e criar uma instância Sumo Logic gratuita, consulte Comece seu teste gratuito hoje.
- Para obter mais informações sobre o uso do Sumo Logic, consulte Origem de registros e métricas HTTP.

Operações de API comumente usadas

PUBLICAR https://endpoint4.collection.us2.sumologic.com/receiver/v1/http/*UNIQUE_ID_PER_COLLECTOR*

Informações adicionais

Nenhum

TriggerMesh

URL do endpoint de invocação de destino da API

Use as informações no tópico Origem do evento para HTTP para formular a URL do endpoint. Um URL de endpoint inclui o nome da origem do evento e o namespace do usuário no seguinte formato:

https://*source-name*.*user-namespace*.cloud.triggermesh.io

Inclui os parâmetros de autorização básicos na solicitação do endpoint.

Tipos de autorização compatíveis

Basic

Parâmetros adicionais de autorização necessários

Nenhum

Documentação da TriggerMesh

Origem do evento para HTTP

Operações de API comumente usadas

Não aplicável

Informações adicionais

Nenhum

Zendesk

URL do endpoint de invocação de destino da API

https://developer.zendesk.com/rest_api/docs/support/tickets

Tipos de autorização compatíveis

Básico, chave de API

Parâmetros adicionais de autorização necessários

Nenhum

Documentação da Zendesk

Segurança e autenticação

Operações de API comumente usadas

POST https://*your_Zendesk_subdomain*/api/v2/tickets

Informações adicionais

As solicitações de API são EventBridge contabilizadas em relação aos seus limites de API do Zendesk. Para obter informações sobre os limites do Zendesk para seu plano, consulte Limites de uso.

Para proteger melhor sua conta e seus dados, recomendamos usar uma chave de API em vez da autenticação básica de credenciais de login.

EventBridge Metas da Amazon para o Amazon API Gateway

É possível usar o Amazon API Gateway para criar, publicar, manter e monitorar APIs. A Amazon EventBridge oferece suporte ao envio de eventos para um endpoint do API Gateway. Ao especificar um endpoint do API Gateway como destino, cada evento enviado ao destino é mapeado para uma solicitação enviada ao endpoint.

> **Important**
>
> EventBridge suporta o uso de endpoints regionais e otimizados para o API Gateway Edge como destinos. No momento, endpoints privados não são compatíveis. Para saber mais sobre os endpoints do cluster, consulte https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-api-endpoint-types.html.

É possível usar um destino do API Gateway para os seguintes casos de uso:

- Para invocar uma API especificada pelo cliente hospedada no API Gateway com base em eventos AWS ou de terceiros.
- Para invocar um endpoint periodicamente em um cronograma.

As informações do evento EventBridge JSON são enviadas como o corpo da solicitação HTTP para o seu endpoint. É possível especificar os outros atributos da solicitação no campo `HttpParameters` do destino da seguinte forma:

- `PathParameterValues` lista os valores que correspondem sequencialmente a qualquer variável de caminho no ARN do endpoint, por exemplo `"arn:aws:execute-api:us-east-1:112233445566:myapi/dev/POST/pets/*/*"`.
- `QueryStringParameters`representa os parâmetros da sequência de caracteres de consulta EventBridge anexados ao endpoint invocado.
- O `HeaderParameters` define cabeçalhos HTTP a serem adicionados à solicitação.

> **Note**
>
> Por questões de segurança, as seguintes chaves de cabeçalho HTTP não são permitidas:
>
> - Qualquer prexico com `X-Amz` ou `X-Amzn`

- `Authorization`
- `Connection`
- `Content-Encoding`
- `Content-Length`
- `Host`
- `Max-Forwards`
- `TE`
- `Transfer-Encoding`
- `Trailer`
- `Upgrade`
- `Via`
- `WWW-Authenticate`
- `X-Forwarded-For`

Parâmetros dinâmicos

Ao invocar um destino do API Gateway, você pode adicionar dados dinamicamente aos eventos enviados ao destino. Para ter mais informações, consulte the section called “Parâmetros de destino”.

Repetições de invocação

Como acontece com todos os alvos, EventBridge repete algumas invocações que falharam. Para o API Gateway, EventBridge repita as respostas enviadas com um código de status HTTP 5xx ou 429 por até 24 horas com recuo exponencial e instabilidade. Depois disso, EventBridge publica uma `FailedInvocations` métrica na Amazon CloudWatch. EventBridge não repete outros erros HTTP 4xx.

Timeout (Tempo limite)

EventBridge As solicitações de regra do API Gateway devem ter um tempo limite máximo de execução do cliente de 5 segundos. Se o API Gateway demorar mais de 5 segundos para responder, EventBridge exceda o tempo limite da solicitação e tente novamente.

EventBridge As solicitações do Pipes API Gateway têm um tempo limite máximo de 29 segundos, o máximo do API Gateway.

AWS AppSync metas para a Amazon EventBridge

AWS AppSync permite que os desenvolvedores conectem seus aplicativos e serviços a dados e eventos com APIs GraphQL e Pub/Sub seguras, sem servidor e de alto desempenho. Com AWS AppSync, você pode publicar atualizações de dados em tempo real em seus aplicativos com mutações do GraphQL. EventBridge suporta a chamada de uma operação de mutação válida do GraphQL para eventos correspondentes. Quando você especifica uma mutação AWS AppSync da API como alvo, AWS AppSync processa o evento por meio de uma operação de mutação, que pode então acionar assinaturas vinculadas à mutação.

> **Note**
>
> EventBridge suporta AWS AppSync APIs públicas do GraphQL. EventBridge atualmente não oferece suporte a APIs AWS AppSync privadas.

Você pode usar um destino da API AWS AppSync GraphQL para os seguintes casos de uso:

- Para enviar por push, transformar e armazenar dados de eventos em suas fontes de dados configuradas.
- Para enviar notificações em tempo real para clientes de aplicações conectados.

> **Note**
>
> AWS AppSync os destinos só suportam a chamada de APIs do AWS AppSync GraphQL usando o `AWS_IAM` tipo de autorização.

Para obter mais informações sobre as APIs do AWS AppSync GraphQL, consulte o GraphQL e a AWS AppSync arquitetura no Guia do desenvolvedor.AWS AppSync

Para especificar um AWS AppSync destino para uma EventBridge regra usando o console

1. Crie ou edite a regra.
2. Em Destino, especifique o destino escolhendo o Serviço da AWS e AWS AppSync.
3. Especifique a operação de mutação a ser analisada e executada, junto com o conjunto de seleção.

- Escolha a AWS AppSync API e, em seguida, a mutação da API GraphQL a ser invocada.
- Em Configurar parâmetros e conjunto de seleção, escolha criar um conjunto de seleção utilizando o mapeamento de chave-valor ou um transformador de entrada.

  Key-value mapping

  Para utilizar o mapeamento de chave-valor para criar o conjunto de seleção:

  - Especifique as variáveis para os parâmetros da API. Cada variável pode ter um valor estático ou uma expressão dinâmica de caminho JSON para a carga útil do evento.
  - Em Conjunto de seleção, escolha as variáveis que deseja incluir na resposta.

  Input transformer

  Para utilizar um transformador de entrada para criar o conjunto de seleção:

  - Especifique um caminho de entrada que defina as variáveis a serem utilizadas.
  - Especifique um modelo de entrada para definir e formatar as informações que deseja passar para o destino.

  Para ter mais informações, consulte ???.

4. Em Perfil de execução, escolha se deseja criar um perfil ou utilizar um perfil existente.
5. Conclua a criação ou a edição da regra.

Exemplo: AWS AppSync metas para a Amazon EventBridge

No exemplo a seguir, explicaremos como especificar um AWS AppSync destino para uma EventBridge regra, incluindo a definição de uma transformação de entrada para formatar eventos para entrega.

Suponha que você tenha uma API AWS AppSync GraphQL`Ec2EventAPI`,, definida pelo seguinte esquema:

```
type Event {
    id: ID!
    statusCode: String
    instanceId: String
}

type Mutation {
    pushEvent(id: ID!, statusCode: String!, instanceId: String): Event
```

```
}

type Query {
    listEvents: [Event]
}

type Subscription {
    subscribeToEvent(id: ID, statusCode: String, instanceId: String): Event
        @aws_subscribe(mutations: ["pushEvent"])
}
```

Clientes de aplicações que utilizam essa API podem assinar o `subscribeToEvent`, cuja assinatura é acionada pela mutação `pushEvent`.

Você pode criar uma EventBridge regra com um destino que envia eventos para a AppSync API por meio da `pushEvent` mutação. Quando a mutação é invocada, qualquer cliente assinante recebe o evento.

Para especificar essa API como o destino de uma EventBridge regra, você faria o seguinte:

1. Defina o nome do recurso da Amazon (ARN) do destino da regra como o ARN do endpoint de GraphQL da API `Ec2EventAPI`.
2. Especifique a operação de mutação de GraphQL como um parâmetro de destino:

   ```
   mutation CreatePushEvent($id: ID!, $statusCode: String, $instanceId: String) {
     pushEvent(id: $input, statusCode: $statusCode, instanceId: $instanceId) {
       id
       statusCode
       instanceId
     }
   }
   ```

   O conjunto de seleção de mutações deve incluir todos os campos que você deseja assinar em sua assinatura de GraphQL.
3. Configure um transformador de entrada para especificar como os dados dos eventos correspondentes são utilizados na operação.

   Suponha que você tenha selecionado o evento de entrada `“EC2 Instance Launch Successful”`:

```
{
  "version": "0",
  "id": "3e3c153a-8339-4e30-8c35-687ebef853fe",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:31:47Z",
  "region": "us-east-1",
  "resources": ["arn:aws:autoscaling:us-
east-1:123456789012:autoScalingGroup:eb56d16b-bbf0-401d-b893-
d5978ed4a025:autoScalingGroupName/sampleLuanchSucASG", "arn:aws:ec2:us-
east-1:123456789012:instance/i-b188560f"],
  "detail": {
    "StatusCode": "InProgress",
    "AutoScalingGroupName": "sampleLuanchSucASG",
    "ActivityId": "9cabb81f-42de-417d-8aa7-ce16bf026590",
    "Details": {
      "Availability Zone": "us-east-1b",
      "Subnet ID": "subnet-95bfcebe"
    },
    "RequestId": "9cabb81f-42de-417d-8aa7-ce16bf026590",
    "EndTime": "2015-11-11T21:31:47.208Z",
    "EC2InstanceId": "i-b188560f",
    "StartTime": "2015-11-11T21:31:13.671Z",
    "Cause": "At 2015-11-11T21:31:10Z a user request created an AutoScalingGroup
 changing the desired capacity from 0 to 1.  At 2015-11-11T21:31:11Z an instance was
 started in response to a difference between desired and actual capacity, increasing
 the capacity from 0 to 1."
  }
}
```

É possível definir as seguintes variáveis para uso no modelo, utilizando caminho de entrada do transformador de entrada do destino:

```
{
  "id": "$.id",
  "statusCode": "$.detail.StatusCode",
  "EC2InstanceId": "$.detail.EC2InstanceId"
}
```

Componha o modelo do transformador de entrada para definir as variáveis que EventBridge passam para a operação de AWS AppSync mutação. O modelo deve ser avaliado como JSON. Determinado o nosso caminho de entrada, é possível compor o seguinte modelo:

```
{
  "id": <id>,
  "statusCode": <statusCode>,
  "instanceId": <EC2InstanceId>
}
```

Conexões para destinos de endpoint HTTP

Uma conexão define o método de autorização e as credenciais a EventBridge serem usadas na conexão com um determinado endpoint HTTP. Quando você define as configurações de autorização e cria uma conexão, ele cria um segredo AWS Secrets Manager para armazenar com segurança as informações de autorização. Você também pode adicionar parâmetros adicionais para incluir na conexão, conforme apropriado para seu destino de endpoint HTTP.

Use conexões com:

- Destinos da API

  Ao criar um destino da API, é especificada uma conexão a ser usada para isso. Você pode escolher uma conexão existente na sua conta ou criar uma conexão ao criar um destino de API.

Métodos de autorização para conexões

EventBridge as conexões suportam os seguintes métodos de autorização:

- Basic
- Chave de API

  Para autorização básica e de chave de API, EventBridge preenche os cabeçalhos de autorização necessários para você.
- OAuth

  Para autorização do OAuth, EventBridge também troca o ID e o segredo do cliente por um token de acesso e, em seguida, os gerencia com segurança.

Tokens OAUTH são atualizados quando uma resposta 401 ou 407 é retornada.

Ao criar uma conexão, também é possível incluir o cabeçalho, o corpo e os parâmetros de consulta necessários para autorização com um endpoint. Você pode usar a mesma conexão para mais de um endpoint HTTP se a autorização para o endpoint for a mesma.

Quando você cria uma conexão e adiciona parâmetros de autorização, EventBridge cria uma entrada secreta AWS Secrets Manager. O custo de armazenar e de avaliar o segredo do Secrets Manager está incluído na cobrança pelo uso de um destino da API. Para saber mais sobre as melhores práticas para usar segredos com destinos de API, consulte AWS::Events::ApiDestinationo Guia CloudFormation do usuário.

> **Note**
>
> Para criar ou atualizar uma conexão com êxito, é preciso usar uma conta que tenha permissão para usar o Secrets Manager. A permissão necessária está incluída no AmazonEventBridgeFullAccess política. A mesma permissão é concedida ao perfil vinculado ao serviço criado em sua conta para a conexão.

Criação de conexões para destinos de endpoint HTTP

Para criar uma conexão para uso com endpoints HTTP usando o console EventBridge

1. Faça login AWS usando uma conta que tenha permissões para gerenciar EventBridge e abrir o EventBridge console.
2. No painel de navegação à esquerda, escolha Destinos da API.
3. Role para baixo até a tabela de destinos da API e escolha a guia Conexões.
4. Escolha Criar conexão.
5. Na página Criar conexão, insira um Nome da conexão para ela.
6. Insira uma Descrição para a conexão.
7. Em Tipo de autorização, selecione o tipo de autorização a ser usada para autorizar conexões com o endpoint HTTP especificado para o destino da API que usa essa conexão. Execute um destes procedimentos:
   - Escolha Básico (nome de usuário/senha), e insira o Nome de usuário e a Senha a serem usados para autorizar com o endpoint HTTP.

- Escolha Credenciais do cliente OAuth e insira o endpoint de autorização, o método HTTP, a ID do cliente e o segredo do cliente a serem usados para autorizar com o endpoint.

  Em Parâmetros HTTP do OAuth, adicione quaisquer parâmetros adicionais a serem incluídos para autorização com o endpoint de autorização. Selecione um Parâmetro na lista suspensa e insira uma Chave e um Valor. Para incluir um parâmetro adicional, escolha Adicionar parâmetro.

  Em Parâmetros de invocação HTTP, adicione os parâmetros adicionais a serem incluídos na solicitação de autorização. Para adicionar um parâmetro, escolha um Parâmetro na lista suspensa e insira uma Chave e um Valor. Para incluir um parâmetro adicional, escolha Adicionar parâmetro.
- Escolha a chave de API e insira o nome da chave de API e o Valor associado a ser usado na autorização da chave de API.

  Em Parâmetros de invocação HTTP, adicione os parâmetros adicionais a serem incluídos na solicitação de autorização. Para adicionar um parâmetro, escolha um Parâmetro na lista suspensa e insira uma Chave e um Valor. Para incluir um parâmetro adicional, escolha Adicionar parâmetro.

8. Selecione Create (Criar).

Editando conexões usando o EventBridge console

Você pode editar conexões existentes.

Para editar uma conexão usando o EventBridge console

1. Faça login AWS usando uma conta que tenha permissões para gerenciar EventBridge e abrir o EventBridge console.
2. No painel de navegação à esquerda, escolha Destinos da API.
3. Role para baixo até a tabela de destinos da API e escolha a guia Conexões.
4. Na tabela Conexões, escolha a conexão a ser editada.
5. Na página Detalhes da conexão, escolha Editar.
6. Atualize os valores da conexão e escolha Atualizar.

Desautorizando conexões usando o console EventBridge

Ao desautorizar uma conexão, ela remove todos os parâmetros de autorização. A remoção dos parâmetros de autorização remove o segredo da conexão, para que ele possa ser reutilizado sem precisar criar uma nova conexão.

> **Note**
>
> Você deve atualizar todos os endpoints HTTP que usam a conexão não autorizada para usar uma conexão diferente para enviar solicitações com êxito ao endpoint HTTP.

Para desautorizar uma conexão

1. Faça login AWS usando uma conta que tenha permissões para gerenciar EventBridge e abrir o EventBridge console.
2. No painel de navegação à esquerda, escolha Destinos da API.
3. Role para baixo até a tabela de destinos da API e escolha a guia Conexões.
4. Na tabela Conexões, escolha a conexão.
5. Na página Detalhes da conexão, escolha Desautorizar.
6. Na caixa de diálogo Remover autorização da conexão?, insira o nome da conexão e escolha Remover autorização.

O status da conexão muda para Desautorizando até que o processo seja concluído. Em seguida, o status muda para Desautorizado. Agora é possível editar a conexão para adicionar novos parâmetros de autorização.

Enviar e receber EventBridge eventos da Amazon entre AWS contas

Você pode configurar EventBridge para enviar e receber eventos entre os barramentos de eventos nas AWS contas. Ao configurar EventBridge para enviar ou receber eventos entre contas, você pode especificar quais AWS contas podem enviar ou receber eventos do barramento de eventos em sua conta. Também é possível permitir ou negar eventos de regras específicas associadas ao barramento de eventos ou eventos de origens específicas. Para obter mais informações, consulte Simplificando o acesso entre contas com as políticas de recursos da Amazon EventBridge

> **Note**
>
> Se você usar AWS Organizations, você pode especificar uma organização e conceder acesso a todas as contas dessa organização. Além disso, o barramento de eventos de envio deve ter perfis do IAM associados ao enviar eventos para outra conta. Para obter mais informações, consulte O que é o AWS Organizations no Guia do usuário do AWS Organizations .

> **Note**
>
> Se estiver usando um plano de resposta do Incident Manager como destino, todos os planos de resposta compartilhados com sua conta estarão disponíveis por padrão.

Você pode enviar e receber eventos entre barramentos de eventos em AWS contas dentro da mesma região em todas as regiões e entre contas em regiões diferentes, desde que a região de destino seja uma região de destino entre regiões compatível.

As etapas a serem configuradas EventBridge para enviar ou receber eventos de um barramento de eventos em uma conta diferente incluem o seguinte:

- Na conta do destinatário, edite as permissões em um barramento de eventos para permitir que AWS contas específicas, uma organização ou todas as AWS contas enviem eventos para a conta do destinatário.
- Na conta remetente, configure uma ou mais regras que têm o barramento de eventos da conta destinatária como destino.

  Se a conta do remetente herdar permissões para enviar eventos de uma AWS organização, a conta do remetente também deverá ter uma função do IAM com políticas que permitam enviar eventos para a conta do destinatário. Se você usar o AWS Management Console para criar a regra que tem como alvo o barramento de eventos na conta do destinatário, a função será criada automaticamente. Se você usar o AWS CLI, deverá criar a função manualmente.
- Na conta destinatária, configure uma ou mais regras que correspondam aos eventos oriundos da conta remetente.

Os eventos enviados de uma conta para outra são cobrados na conta de envio como eventos personalizados. A conta de recebimento não é cobrada. Para obter mais informações, consulte Amazon EventBridge Pricing.

Se uma conta destinatária puder definir uma regra que envia eventos recebidos de uma conta remetente para uma terceira conta, esses eventos não serão enviados para a terceira conta.

Se você tiver três ônibus de eventos na mesma conta e configurar uma regra no primeiro ônibus de eventos para encaminhar eventos do segundo ônibus de eventos para um terceiro ônibus de eventos, esses eventos não serão enviados para o terceiro ônibus de eventos.

O vídeo a seguir aborda eventos de roteamento entre contas: Roteamento de eventos para ônibus em outras contas AWS

Conceda permissões para permitir eventos de outras AWS contas

Para receber eventos de outras contas ou organizações, primeiro você deve editar as permissões no barramento de eventos padrão da conta. O barramento de eventos padrão aceita eventos de AWS serviços, outras AWS contas autorizadas e `PutEvents` chamadas. As permissões para um barramento de eventos são concedidas ou negadas usando uma política baseada em recursos anexada ao barramento de eventos. Na política, você pode conceder permissões a outras AWS contas usando o ID da conta ou a uma AWS organização usando o ID da organização. Para saber mais sobre permissões de barramento de eventos, consulte Permissões de barramentos de evento do Amazon EventBridge Pipes.

> **Note**
>
> EventBridge agora exige que todas as novas metas de barramento de eventos entre contas adicionem funções do IAM. Isto se aplica somente aos destinos de barramentos de eventos criados após 2 de março de 2023. As aplicações criadas sem um perfil do IAM antes dessa data não são afetados. No entanto, é recomendado adicionar perfis do IAM para conceder aos usuários acesso a recursos em outra conta, pois isso garante que os limites da organização usando as políticas de controle de serviços (SCPs) sejam aplicados para determinar quem pode enviar e receber eventos de contas em sua organização.

> **Important**
>
> Se você optar por receber eventos de todas as AWS contas, tenha o cuidado de criar regras que correspondam apenas aos eventos a serem recebidos de outras pessoas. Para criar regras mais seguras, certifique-se de que o padrão de evento para cada regra contenha um campo `Account` com os IDs de uma ou mais contas das quais receber eventos. As regras que têm um padrão de evento contendo um campo Conta não correspondem aos eventos enviados de contas que não estão listadas no campo `Account`. Para ter mais informações, consulte EventBridge Eventos da Amazon.

Regras para eventos entre AWS contas

Se sua conta estiver configurada para receber eventos de ônibus de eventos em outras AWS contas, você poderá criar regras que correspondam a esses eventos. Defina o padrão do evento da regra para corresponder aos barramentos de eventos que está recebendo em outra conta.

A não ser que especifique `account` no padrão de evento de uma regra, qualquer uma das regras da conta, nova ou existente, que corresponda a eventos recebidos de outras contas é disparada com base nesses eventos. Se estiver recebendo barramentos de eventos de outra conta e quiser que uma regra dispare somente naquele padrão de evento gerado da sua própria conta, adicione `account` e especifique seu próprio ID de conta para o padrão de evento da regra.

Se você configurou sua AWS conta para aceitar eventos de ônibus de eventos em todas as AWS contas, é altamente recomendável que você adicione `account` a todas as EventBridge regras da sua conta. Isso evita que regras em sua conta sejam acionadas em eventos de contas desconhecidas AWS . Quando você especifica o campo `account` na regra, pode especificar os IDs de conta de mais de uma conta da AWS no campo.

Para que uma regra seja acionada em um evento correspondente de qualquer barramento de eventos na AWS conta à qual você tenha concedido permissões, não especifique * no `account` campo da regra. Isso não corresponderia a nenhum evento, porque * nunca é exibido no campo `account` de um evento. Em vez disso, basta omitir o campo `account` da regra.

Criação de regras que enviam eventos entre AWS contas

Especificar um barramento de eventos em outra conta como destino faz parte da criação da regra.

Para criar uma regra que envie eventos para uma AWS conta diferente usando o console

1. Siga as etapas no procedimento ???.
2. Na etapa ???, quando solicitado a escolher um tipo de destino:
   a. Selecione o ônibus EventBridge do evento.
   b. Selecione Barramento de eventos em uma conta ou região diferente.
   c. Para Barramento de eventos como destino, insira o ARN do barramento de eventos que deseja usar.
3. Conclua a criação da regra ao seguir as etapas de procedimento.

Enviando e recebendo EventBridge eventos da Amazon entre AWS regiões

Você pode configurar EventBridge para enviar e receber eventos entre AWS regiões. Também é possível permitir ou negar eventos de regiões específicas, regras específicas associadas ao barramento de eventos ou eventos de origens específicas. Para obter mais informações, consulte Introdução ao roteamento de eventos entre regiões com a Amazon EventBridge

As seguintes regiões são regiões de destino compatíveis:

- 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 (Tóquio)
- Ásia-Pacífico (Seul)
- Asia Pacific (Osaka)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Sydney)

- Ásia-Pacífico (Melbourne)
- Canadá (Central)
- Oeste do Canadá (Calgary)
- Europa (Frankfurt)
- Europa (Espanha)
- Europa (Zurique)
- Europa (Estocolmo)
- Europa (Milão)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- Israel (Tel Aviv)
- Oriente Médio (Emirados Árabes Unidos)
- Middle East (Bahrain)
- South America (São Paulo)

O vídeo a seguir aborda eventos de roteamento entre regiões usando o https://console.aws.amazon.com/events/, AWS CloudFormation, e AWS Serverless Application Model: Roteamento de eventos entre regiões

Criação de regras que enviam eventos para uma AWS região diferente

Especificar um barramento de eventos em outra AWS região como destino faz parte da criação da regra.

Para criar uma regra que envie eventos para uma AWS conta diferente usando o console

1. Siga as etapas no procedimento ???.
2. Na etapa ???, quando solicitado a escolher um tipo de destino:
    a. Selecione o ônibus EventBridge do evento.
    b. Selecione Barramento de eventos em uma conta ou região diferente.

c. Para Barramento de eventos como destino, insira o ARN do barramento de eventos que deseja usar.

3. Conclua a criação da regra ao seguir as etapas de procedimento.

Enviar e receber EventBridge eventos da Amazon entre ônibus de eventos na mesma conta e região

Você pode configurar EventBridge para enviar e receber eventos entre barramentos de eventos na mesma AWS conta e região.

Ao configurar EventBridge para enviar ou receber eventos entre barramentos de eventos, você usa funções do IAM no barramento de eventos do remetente para dar permissão ao barramento de eventos do remetente para enviar eventos ao barramento de eventos do receptor. Políticas baseadas em recursos são usadas no barramento de eventos do destinatário para dar permissão ao barramento de eventos do destinatário para receber eventos do barramento de eventos do remetente. Também é possível permitir ou negar eventos de determinados barramentos de eventos, regras específicas associadas ao barramento de eventos ou eventos de origens específicas. Para obter mais informações sobre permissões de barramento de eventos, consulte.Permissões de barramentos de evento do Amazon EventBridge Pipes

As etapas a serem configuradas EventBridge para enviar ou receber eventos entre os barramentos de eventos em sua conta incluem o seguinte:

- Para usar um perfil do IAM existente, é preciso conceder as permissões do barramento de eventos do remetente ao barramento de eventos do destinatário ou as permissões do barramento de eventos do destinatário ao barramento de eventos do remetente.
- Na conta remetente, configure uma ou mais regras que têm o barramento de eventos do destinatário como o destino e crie um perfil do IAM. Para obter um exemplo da política que deve ser anexada ao perfil, consulte ???.
- No barramento de eventos do destinatário, edite as permissões para permitir que os eventos sejam transmitidos de outro barramento de eventos.
- No evento do destinatário, configure uma ou mais regras que correspondam aos eventos oriundos do barramento de eventos do remetente.

> **Note**
>
> EventBridge não é possível rotear eventos recebidos de um ônibus de eventos do remetente para um terceiro ônibus de eventos.

Os eventos enviados de um barramento de eventos para outro são cobrados como eventos personalizados. Para obter mais informações, consulte Definição de preço do Amazon EventBridge.

Criação de regras que enviam eventos para um barramento de eventos diferente na mesma AWS conta e região

Para enviar eventos para outro barramento de eventos, é necessário criar uma regra com um barramento de eventos como destino. Especificar um barramento de eventos na mesma AWS conta e região como destino faz parte da criação da regra.

Para criar uma regra que envie eventos para um barramento de eventos diferente na mesma AWS conta e região usando o console

1. Siga as etapas no procedimento ???.
2. Na etapa ???, quando solicitado a escolher um tipo de destino:
   a. Selecione o ônibus EventBridge do evento.
   b. Selecione Event bus na mesma AWS conta e região.
   c. Em barramento de eventos como destino, selecione um tipo de barramento de eventos na lista suspensa.
3. Conclua a criação da regra ao seguir as etapas de procedimento.

Transformação EventBridge de insumos da Amazon

Você pode personalizar o texto de um evento antes de EventBridge passar as informações para o destino de uma regra. Ao usar o transformador de entrada no console ou na API, são definidas variáveis que usam o caminho JSON para referenciar valores na origem original do evento. O evento transformado é enviado para um destino em vez do evento original. No entanto, os parâmetros do caminho dinâmico devem fazer referência ao evento original, não ao evento transformado. É possível definir até 100 variáveis, atribuindo a cada um valor da entrada. Depois, use essas variáveis no modelo de entrada como <*variable-name*>.

Para obter um tutorial sobre como usar o transformador de entrada, consulte ???.

> **Note**
>
> EventBridge não suporta toda a sintaxe do JSON Path e a avalia em tempo de execução. A sintaxe compatível inclui:
>
> - notação de pontos (por exemplo, `$.detail`)
> - traços
> - sublinhados
> - caracteres alfanuméricos
> - índices de matriz
> - curingas (*)

Neste tópico:

Variáveis predefinidas

Há variáveis predefinidas que podem ser usadas sem definir um caminho JSON. Estas variáveis são reservadas e não é possível criar variáveis com esses nomes:

- `aws.events.rule-arn`— O Amazon Resource Name (ARN) da EventBridge regra.
- `aws.events.rule-name`— O nome da EventBridge regra.
- `aws.events.event.ingestion-time`— A hora em que o evento foi recebido por EventBridge. Este é um carimbo de data/hora ISO 8601. Essa variável é gerada por EventBridge e não pode ser substituída.
- `aws.events.event`: a carga útil do evento original como JSON (sem o campo `detail`). Só pode ser usado como um valor para um campo JSON, pois o conteúdo não tem escape.
- `aws.events.event.json`: a carga útil completa do evento original como JSON (com o campo `detail`). Só pode ser usado como um valor para um campo JSON, pois o conteúdo não tem escape.

Exemplos de transformação de entrada

Veja a seguir um exemplo de evento do Amazon EC2.

```
{
  "version": "0",
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2015-11-11T21:29:54Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"
  ],
  "detail": {
    "instance-id": "i-0123456789",
    "state": "RUNNING"
  }
}
```

Ao definir uma regra no console, selecione a opção Input Transformer (Transformador de entrada) em Configure input (Configurar entrada). Essa opção exibe duas caixas de texto: uma para o Input Path (Caminho de Entrada) e outra para o Input Template (Modelo de entrada).

O Caminho de entrada é usado para definir variáveis. Use o caminho JSON para fazer referência a itens em seu evento e armazenar esses valores em variáveis. Por exemplo, é possível criar um caminho de entrada para fazer referência a valores no evento de exemplo inserindo o seguinte na primeira caixa de texto: Também é possível usar colchetes e índices para obter itens de matrizes.

> **Note**
>
> EventBridge substitui os transformadores de entrada em tempo de execução para garantir uma saída JSON válida. Por isso, coloque aspas nas variáveis que se referem aos parâmetros de caminho JSON, mas não coloque aspas nas variáveis que se referem a objetos ou matrizes JSON.

```
{
  "timestamp" : "$.time",
  "instance" : "$.detail.instance-id",
  "state" : "$.detail.state",
  "resource" : "$.resources[0]"
}
```

Isto definirá duas variáveis, `<timestamp>`, `<instance>`, `<state>` e `<resource>`. É possível pode fazer referência a essas variáveis ao criar o modelo de entrada.

O modelo de entrada é um modelo para as informações que você deseja passar para seu destino. É possível criar um modelo que transmita uma string ou JSON para o destino. Usando o evento anterior e o caminho de entrada, os exemplos de modelo de entrada a seguir transformarão o evento na saída de exemplo antes de fazer o roteamento dele para um destino.

Descrição	Modelo	Saída
String simples	`"instance <instance> is in <state>"`	`"instance i-0123456789 is in RUNNING"`

Descrição	Modelo	Saída
String com aspas de escape	`"instance \"<instance> \" is in <state>"`	`"instance \"i-01234 56789\" is in RUNNING"` Observe que esse é o comportamento no EventBrid ge console. A AWS CLI faz o escape dos caractere s de barra e o resultado é `"instance "i-012345 6789" is in RUNNING"` .
JSON simples	`{ "instance" : <instance>, "state": <state> }`	`{ "instance" : "i-0123456789", "state": "RUNNING" }`
JSON com strings e variáveis	`{ "instance" : <instance >, "state": "<state>", "instanceStatus": "instance \"<instance> \" is in <state>" }`	`{ "instance" : "i-012345 6789", "state": "RUNNING", "instanceStatus": "instance \"i-01234 56789\" is in RUNNING" }`
JSON com uma mistura de variáveis e informações estáticas	`{ "instance" : <instance>, "state": [ 9, <state>, true ], "Transformed" : "Yes" }`	`{ "instance" : "i-0123456789", "state": [ 9, "RUNNING", true ], "Transformed" : "Yes" }`

Descrição	Modelo	Saída
Incluir variáveis reservadas no JSON	`{ "instance" : <instance>, "state": <state>, "ruleArn" : <aws.events.rule-arn>, "ruleName" : <aws.events.rule-name>, "originalEvent" : <aws.events.event.json> }`	`{ "instance" : "i-0123456789", "state": "RUNNING", "ruleArn" : "arn:aws:events:us-east-2:123456789012:rule/example", "ruleName" : "example", "originalEvent" : { ... // commented for brevity } }`
Incluir variáveis reservadas em uma string	`"<aws.events.rule-name> triggered"`	`"example triggered"`
Grupo de CloudWatch registros da Amazon	`{ "timestamp" : <timestamp>, "message": "instance \"<instance>\" is in <state>" }`	`{ "timestamp" : 2015-11-11T21:29:54Z, "message": "instance "i-0123456789" is in RUNNING }`

Transformando a entrada usando a API EventBridge

Para obter informações sobre como usar a EventBridge API para transformar a entrada, consulte Usar o Input Transformer para extrair dados de um evento e inserir esses dados no destino.

Transformando a entrada usando AWS CloudFormation

Para obter informações sobre como usar AWS CloudFormation para transformar a entrada, consulte AWS::Events::Rule InputTransformer.

Problemas comuns com a transformação de entrada

Esses são alguns problemas comuns ao transformar a entrada em: EventBridge

- Para strings, as aspas são necessárias.
- Não há validação ao criar o caminho JSON para o modelo.
- Se especificar uma variável para corresponder a um caminho JSON que não existe no evento, essa variável não será criada e não aparecerá na saída.
- Propriedades JSON, como `aws.events.event.json`, só podem ser usadas como o valor de um campo JSON, não embutidas em outras strings.
- EventBridge não escapa dos valores extraídos pelo caminho de entrada, ao preencher o modelo de entrada de um destino.
- Se um caminho JSON fizer referência a um objeto ou matriz JSON, mas a variável for referenciada em uma string, EventBridge removerá todas as aspas internas para garantir uma string válida. Por exemplo, para uma variável `<detail>` apontada`$.detail`, “Detalhe é**<**detail**>**” resultaria na EventBridge remoção de aspas do objeto.

  Portanto, se quiser gerar um objeto JSON com base em uma única variável de caminho JSON, deverá colocá-lo como uma chave. Neste exemplo, `{"detail": <detail>}`.
- As aspas não são necessárias para variáveis que representam cadeias de caracteres. Elas são permitidas, mas adicionam EventBridge automaticamente aspas aos valores das variáveis de string durante a transformação, para garantir que a saída da transformação seja um JSON válido. EventBridge não adiciona aspas às variáveis que representam objetos ou matrizes JSON. Não adicione aspas para variáveis que representem objetos ou matrizes JSON.

  Por exemplo, o seguinte modelo de entrada inclui variáveis que representam cadeias de caracteres e objetos JSON:

```
{
  "ruleArn" : <aws.events.rule-arn>,
  "ruleName" : <aws.events.rule-name>,
  "originalEvent" : <aws.events.event.json>
}
```

  Resultando em JSON válido com cotação adequada:

```
{
```

```
  "ruleArn" : "arn:aws:events:us-east-2:123456789012:rule/example",
  "ruleName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

- Para saída de texto (não JSON) como cadeias de caracteres de várias linhas, coloque cada linha separada em seu modelo de entrada entre aspas duplas.

  Por exemplo, se você estivesse comparando Amazon Inspector Finding events com o seguinte padrão de evento:

  ```
  {
    "detail": {
      "severity": ["HIGH"],
      "status": ["ACTIVE"]
    },
    "detail-type": ["Inspector2 Finding"],
    "source": ["inspector2"]
  }
  ```

  E usando o seguinte caminho de entrada:

  ```
  {
    "account": "$.detail.awsAccountId",
    "ami": "$.detail.resources[0].details.awsEc2Instance.imageId",
    "arn": "$.detail.findingArn",
    "description": "$.detail.description",
    "instance": "$.detail.resources[0].id",
    "platform": "$.detail.resources[0].details.awsEc2Instance.platform",
    "region": "$.detail.resources[0].region",
    "severity": "$.detail.severity",
    "time": "$.time",
    "title": "$.detail.title",
    "type": "$.detail.type"
  }
  ```

  Você pode usar o modelo de entrada abaixo para gerar uma saída de string de várias linhas:

  ```
  "<severity> severity finding <title>"
  "Description: <description>"
  ```

```
"ARN: \"<arn>\""
"Type: <type>"
"AWS Account: <account>"
"Region: <region>"
"EC2 Instance: <instance>"
"Platform: <platform>"
"AMI: <ami>"
```

Como configurar um transformador de entrada como parte da criação de uma regra

Como parte da criação de uma regra, você pode especificar um transformador de entrada EventBridge para usar no processamento de eventos correspondentes antes de enviar esses eventos para o destino especificado. Você pode configurar transformadores de entrada para destinos que sejam AWS serviços ou destinos de API.

Para criar um transformador de entrada de destino como parte de uma regra

1. Siga as etapas para criar uma regra, conforme detalhado em ???.
2. Na Etapa 3: selecione os destinos, expanda Configurações adicionais.
3. Em Configurar entrada de destino, escolha Transformador de entrada na lista suspensa.

   Clique em Configurar transformador de entrada.

   EventBridge exibe a caixa de diálogo Configurar transformador de entrada.
4. Na seção Evento de amostra, escolha um Tipo de evento de amostra com o qual deseja testar seu padrão de evento. Você pode escolher um AWS evento, um evento de parceiro ou inserir seu próprio evento personalizado.

   AWS events

   Selecione entre os eventos emitidos pelos Serviços da AWS compatíveis.

   1. Selecione Eventos da AWS .
   2. Em Eventos de amostra, escolha o AWS evento desejado. Os eventos são organizados por AWS serviço.

      Quando você seleciona um evento, EventBridge preenche o evento de amostra.

Por exemplo, se você escolher S3 Object Created, EventBridge exibirá um exemplo do evento S3 Object Created.

3. (Opcional) Também é possível selecionar Copiar para copiar o evento de amostra para a área de transferência do seu dispositivo.

Partner events

Selecione entre os eventos emitidos por serviços terceirizados que oferecem suporte EventBridge, como o Salesforce.

1. Selecione eventos de EventBridge parceiros.
2. Em Exemplos de eventos, escolha o evento do parceiro desejado. Os eventos são organizados pelo parceiro.

   Quando você seleciona um evento, EventBridge preenche o evento de amostra.
3. (Opcional) Também é possível selecionar Copiar para copiar o evento de amostra para a área de transferência do seu dispositivo.

Enter your own

Insira o seu próprio evento em texto JSON.

1. Selecione Inserir seu próprio.
2. EventBridge preenche o evento de amostra com um modelo de atributos de evento necessários.
3. Edite e adicione ao evento de amostra conforme desejado. O evento de amostra deve ser JSON válido.
4. (Opcional) Também é possível escolher uma das seguintes opções:
   - Copiar: copie o evento de amostra para a área de transferência do seu dispositivo.
   - Aprimorar: facilita a leitura do texto JSON adicionando quebras de linha, tabulações e espaços.

5. (Opcional) Expanda a seção Exemplos de caminhos de entrada, modelos e saídas para ver exemplos de:

- Como os caminhos JSON são usados para definir variáveis que representam dados de eventos
- Como essas variáveis podem ser usadas em um modelo de transformador de entrada
- A saída resultante que é EventBridge enviada para o destino

Para obter exemplos mais detalhados de transformações de entrada, consulte ???.

6. Na seção Transformador de entrada de destino, defina as variáveis que deseja usar no modelo de entrada.

   Variáveis usam caminho JSON para fazer referência a valores na origem do evento original. Em seguida, você pode referenciar essas variáveis no modelo de entrada para incluir dados do evento de origem original no evento transformado que EventBridge passa para o destino. É possível definir até 100 variáveis. O transformador de entrada deve ser um JSON válido.

   Por exemplo, suponha que você tenha escolhido o AWS evento S3 Object Created como seu evento de amostra para esse transformador de entrada. É possível definir as seguintes variáveis para uso em seu modelo:

   ```
   {
     "requester": "$.detail.requester",
     "key": "$.detail.object.key",
     "bucket": "$.detail.bucket.name"
   }
   ```

   (Opcional) Também é possível escolher Copiar para copiar o transformador de entrada para a área de transferência do seu dispositivo.

7. Na seção Modelo, componha o modelo que você deseja usar para determinar o que EventBridge passa para o alvo.

   É possível usar JSON, strings, informações estáticas, variáveis que você definiu, bem como variáveis reservadas. Para obter exemplos mais detalhados de transformações de entrada, consulte ???.

   Por exemplo, suponha que tenha definido as variáveis no exemplo anterior. O modelo a seguir poderia ser composto, que faz referência a essas variáveis, bem como às variáveis reservadas e às informações estáticas.

```
{
    "message": "<requester> has created the object \"<key>\" in the bucket
 \"<bucket>\"",
    "RuleName": <aws.events.rule-name>,
    "ruleArn" : <aws.events.rule-arn>,
    "Transformed": "Yes"
}
```

(Opcional) Também é possível escolher Copiar para copiar o modelo para a área de transferência do seu dispositivo.

8. Para testar seu modelo, selecione Gerar saída.

   EventBridge processa o evento de amostra com base no modelo de entrada e exibe a saída transformada gerada em Saída. Essas são as informações que EventBridge passarão para o destino no lugar do evento de origem original.

   A saída gerada para o modelo de entrada de exemplo descrito acima seria a seguinte:

```
{
    "message": "123456789012 has created the object "example-key" in the bucket
 "example-bucket"",
    "RuleName": rule-name,
    "ruleArn" : arn:aws:events:us-east-1:123456789012:rule/rule-name,
    "Transformed": "Yes"
}
```

   (Opcional) Também é possível escolher Copiar para copiar a saída gerada para a área de transferência do seu dispositivo.

9. Selecione Confirmar.
10. Siga o restante das etapas para criar uma regra, conforme detalhado em ???.

Testando um transformador de entrada de destino usando o Sandbox EventBridge

Você pode usar transformadores de entrada para personalizar o texto de um evento antes de EventBridge passar as informações para o destino de uma regra.

A configuração de um transformador de entrada normalmente faz parte de um processo maior de especificar um destino ao criar uma nova regra ou editar uma existente. No entanto EventBridge, usando o Sandbox in, você pode configurar rapidamente um transformador de entrada e usar um evento de amostra para confirmar que está obtendo a saída desejada, sem precisar criar ou editar uma regra.

Para obter mais informações sobre a transformação de entrada, consulte ???.

Para testar um transformador de entrada de destino

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. Em Recursos do desenvolvedor, escolha Sandbox e, na página Sandbox, escolha a guia Transformador de entrada de destino.
3. Na seção Evento de amostra, escolha um Tipo de evento de amostra com o qual deseja testar seu padrão de evento. Você pode escolher um AWS evento, um evento de parceiro ou inserir seu próprio evento personalizado.

   AWS events

   Selecione entre os eventos emitidos pelos Serviços da AWS compatíveis.

   1. Selecione Eventos da AWS .
   2. Em Eventos de amostra, escolha o AWS evento desejado. Os eventos são organizados por AWS serviço.

      Quando você seleciona um evento, EventBridge preenche o evento de amostra.

      Por exemplo, se você escolher S3 Object Created, EventBridge exibirá um exemplo do evento S3 Object Created.
   3. (Opcional) Também é possível selecionar Copiar para copiar o evento de amostra para a área de transferência do seu dispositivo.

   Partner events

   Selecione entre os eventos emitidos por serviços terceirizados que oferecem suporte EventBridge, como o Salesforce.

   1. Selecione eventos de EventBridge parceiros.

2. Em Exemplos de eventos, escolha o evento do parceiro desejado. Os eventos são organizados pelo parceiro.

   Quando você seleciona um evento, EventBridge preenche o evento de amostra.

3. (Opcional) Também é possível selecionar Copiar para copiar o evento de amostra para a área de transferência do seu dispositivo.

Enter your own

Insira o seu próprio evento em texto JSON.

1. Selecione Inserir seu próprio.
2. EventBridge preenche o evento de amostra com um modelo de atributos de evento necessários.
3. Edite e adicione ao evento de amostra conforme desejado. O evento de amostra deve ser JSON válido.
4. (Opcional) Também é possível escolher uma das seguintes opções:
   - Copiar: copie o evento de amostra para a área de transferência do seu dispositivo.
   - Aprimorar: facilita a leitura do texto JSON adicionando quebras de linha, tabulações e espaços.

4. (Opcional) Expanda a seção Exemplos de caminhos de entrada, modelos e saídas para ver exemplos de:

   - Como os caminhos JSON são usados para definir variáveis que representam dados de eventos
   - Como essas variáveis podem ser usadas em um modelo de transformador de entrada
   - A saída resultante que é EventBridge enviada para o destino

   Para obter exemplos mais detalhados de transformações de entrada, consulte ???.

5. Na seção Transformador de entrada de destino, defina as variáveis que deseja usar no modelo de entrada.

   Variáveis usam caminho JSON para fazer referência a valores na origem do evento original. Em seguida, você pode referenciar essas variáveis no modelo de entrada para incluir dados

do evento de origem original no evento transformado que EventBridge passa para o destino. É possível definir até 100 variáveis. O transformador de entrada deve ser um JSON válido.

Por exemplo, suponha que você tenha escolhido o AWS evento S3 Object Created como seu evento de amostra para esse transformador de entrada. É possível definir as seguintes variáveis para uso em seu modelo:

```
{
  "requester": "$.detail.requester",
  "key": "$.detail.object.key",
  "bucket": "$.detail.bucket.name"
}
```

(Opcional) Também é possível escolher Copiar para copiar o transformador de entrada para a área de transferência do seu dispositivo.

6. Na seção Modelo, componha o modelo que você deseja usar para determinar o que EventBridge passa para o alvo.

   É possível usar JSON, strings, informações estáticas, variáveis que você definiu, bem como variáveis reservadas. Para obter exemplos mais detalhados de transformações de entrada, consulte ???.

   Por exemplo, suponha que tenha definido as variáveis no exemplo anterior. O modelo a seguir poderia ser composto, que faz referência a essas variáveis, bem como às variáveis reservadas e às informações estáticas.

   ```
   {
       "message": "<requester> has created the object \"<key>\" in the bucket
    \"<bucket>\"",
       "RuleName": <aws.events.rule-name>,
       "ruleArn" : <aws.events.rule-arn>,
       "Transformed": "Yes"
   }
   ```

   (Opcional) Também é possível escolher Copiar para copiar o modelo para a área de transferência do seu dispositivo.

7. Para testar seu modelo, selecione Gerar saída.

EventBridge processa o evento de amostra com base no modelo de entrada e exibe a saída transformada gerada em Saída. Essas são as informações que EventBridge passarão para o destino no lugar do evento de origem original.

A saída gerada para o modelo de entrada de exemplo descrito acima seria a seguinte:

```
{
    "message": "123456789012 has created the object "example-key" in the bucket
 "example-bucket"",
    "RuleName": rule-name,
    "ruleArn" : arn:aws:events:us-east-1:123456789012:rule/rule-name,
    "Transformed": "Yes"
}
```

(Opcional) Também é possível escolher Copiar para copiar a saída gerada para a área de transferência do seu dispositivo.

Arquivamento e reprodução do Amazon EventBridge

No EventBridge, é possível criar um arquivo de eventos para poder reproduzi-los facilmente mais tarde. Por exemplo, você pode querer repetir eventos para se recuperar de erros ou validar uma nova funcionalidade em sua aplicação.

> **Note**
>
> Pode haver um atraso entre a publicação de um evento em um barramento de eventos e a chegada do evento ao arquivo. É recomendado adiar a repetição dos eventos arquivados por 10 minutos para garantir que todos os eventos sejam repetidos.

O seguinte vídeo demonstra o uso do arquivamento e reprodução: Como criar arquivos e reproduções

Tópicos

Arquivamento de eventos da Amazon EventBridge

Ao criar um arquivamento em EventBridge, você pode determinar quais eventos são enviados para o arquivamento especificando um padrão de evento. EventBridge envia eventos que correspondem ao padrão do evento para o arquivo. Também é definido o período de retenção para armazenar eventos no arquivo antes que eles sejam descartados.

Por padrão, EventBridge criptografa dados de eventos em um arquivamento usando o Advanced Encryption Standard (AES-256) de 256 bits sob uma CMK AWS própria, o que ajuda a proteger seus dados contra acesso não autorizado.

> **Note**
>
> Os `SizeBytes` valores `EventCount` e da `DescribeArchive`operação têm um período de reconciliação de 24 horas. Portanto, qualquer evento recém-expirado ou recém-arquivado pode não ser refletido imediatamente nesses valores.

Para criar um arquivamento para todos os eventos

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação à esquerda, escolha Arquivos.
3. Escolha Criar arquivo.
4. Em Detalhes do arquivo, insira um Nome para o arquivo. O nome deve ser exclusivo na sua conta na sua região selecionada.

   Não é possível alterar o nome depois de criar o conjunto de IP.
5. (Opcional) Insira uma Descrição para o arquivo.
6. Em Origem, selecione o barramento de eventos que emite os eventos a serem enviados ao arquivo.
7. Em Período de retenção, siga um destes procedimentos:
   - Escolha Indefinido para reter os eventos no arquivo e nunca excluí-los.
   - Insira o número de dias para reter os eventos. Após o número de dias especificado, EventBridge exclui os eventos do arquivo.
8. Selecione Next (Próximo).
9. Em Padrão de eventos, escolha Sem filtragem de eventos.

10. Escolha Criar arquivo.

Para criar um arquivamento com um padrão de evento

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação à esquerda, escolha Arquivos.
3. Escolha Criar arquivo.
4. Em Detalhes do arquivo, insira um Nome para o arquivo. O nome deve ser exclusivo na sua conta na sua região selecionada.

   Não é possível alterar o nome depois de criar o conjunto de IP.
5. (Opcional) Insira uma Descrição para o arquivo.
6. Em Origem, selecione o barramento de eventos que emite os eventos a serem enviados ao arquivo.
7. Em Período de retenção, siga um destes procedimentos:
   - Escolha Indefinido* para reter os eventos no arquivo e nunca excluí-los.
   - Insira o número de dias para reter os eventos. Após o número de dias especificado, EventBridge exclui os eventos do arquivo.
8. Selecione Next (Próximo).
9. Em Padrão de evento, escolha Filtrar eventos por correspondência de padrões de eventos.
10. Execute um destes procedimentos:
    - Selecione Criador de padrões e Provedor de serviços. Se escolher AWS, selecione também o nome do serviço da AWS e o Tipo de evento a serem usados no padrão.
    - Selecione o editor JSON para criar um padrão manualmente. Também é possível copiar o padrão de uma regra e depois colá-lo no editor JSON.
11. Escolha Criar arquivo.

Para confirmar se os eventos foram enviados com sucesso para o arquivamento, você pode usar a `DescribeArchive`operação da EventBridge API para ver se ela `EventCount` reflete o número de eventos no arquivamento. Se for 0, não há eventos no arquivo.

Como reproduzir eventos arquivados do Amazon EventBridge

Depois de criar um arquivo, é possível reproduzir eventos do arquivo. Por exemplo, se atualizar uma aplicação com funcionalidades adicionais, poderá reproduzir eventos históricos para garantir que os eventos sejam reprocessados para manter a aplicação consistente. Também é possível usar um arquivo para reproduzir eventos para novas funcionalidades. Ao repetir eventos, é possível especificar de qual arquivo reproduzir os eventos, a hora de início e término da repetição do evento, o barramento de eventos ou uma ou mais regras para as quais repetir os eventos.

Os eventos não são necessariamente reproduzidos na mesma ordem em que foram adicionados ao arquivo. Uma repetição processa eventos em repetições com base na hora do evento e os reproduz em intervalos de um minuto. Se especificar uma hora de início e uma hora de término do evento que cubra um intervalo de 20 minutos, os eventos serão reproduzidos primeiro a partir do primeiro minuto desse intervalo de 20 minutos. Em seguida, os eventos do segundo minuto são repetidos. É possível usar a operação `DescribeReplay` da API do EventBridge para determinar o progresso de uma repetição. `EventLastReplayedTime` retorna a data e hora do último evento repetido.

Os eventos são reproduzidos com base no limite de transações por segundo do `PutEvents` da conta da AWS, mas separados dele. É possível solicitar o aumento do limite de PutEvents. Para obter mais informações, consulte Cotas do Amazon EventBridge.

> **Note**
>
> É possível ter um máximo de dez condições de reprodução simultânea por conta e por região da AWS.

Para iniciar a repetição de um evento

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação à esquerda, escolha Reproduções.
3. Escolha Iniciar nova repetição.
4. Informe um Nome para a reprodução e, opcionalmente, uma Descrição.
5. Em Origem, selecione o arquivo do qual reproduzir eventos.
6. Para o destino, é possível reproduzir eventos somente no mesmo barramento de eventos que emitiu os eventos.
7. Em Especificar regras, especifique uma das seguintes:

- Escolha Todas as regras para repetir os eventos de acordo com todas as regras.
- Escolha Especificar regras e selecione a regra ou as regras para as quais reproduzir os eventos.

8. Em Período de tempo de repetição, especifique a data, a hora e o fuso horário para a hora de início e a hora de término. Somente eventos que ocorreram entre a hora de início e a hora de hora de término são reproduzidos.
9. Selecione Iniciar repetição.

Quando os eventos arquivados são repetidos, o status da repetição é Concluído.

Se iniciar uma repetição e depois quiser interrompê-la, poderá cancelá-la, desde que o status seja Iniciando ou Em execução.

Para cancelar uma repetição

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação à esquerda, escolha Reproduções.
3. Escolha a reprodução a ser cancelada.
4. Escolha Cancelar.

Amazon EventBridge Pipes

O Amazon EventBridge Pipes conecta as fontes aos alvos. Os tubos são destinados a point-to-point integrações entre fontes e alvos suportados, com suporte para transformações e enriquecimento avançados. Ele reduz a necessidade de conhecimento especializado e código de integração ao desenvolver arquiteturas orientadas por eventos, promovendo a consistência em todas as aplicações da sua empresa. Para configurar um pipe, a origem é escolhida, adiciona filtragem opcional, define o enriquecimento opcional e escolhe o destino para os dados do evento.

> **Note**
>
> Também é possível rotear eventos usando barramentos de eventos. Os ônibus de eventos são adequados para o many-to-many roteamento de eventos entre serviços orientados a eventos. Para ter mais informações, consulte ???.

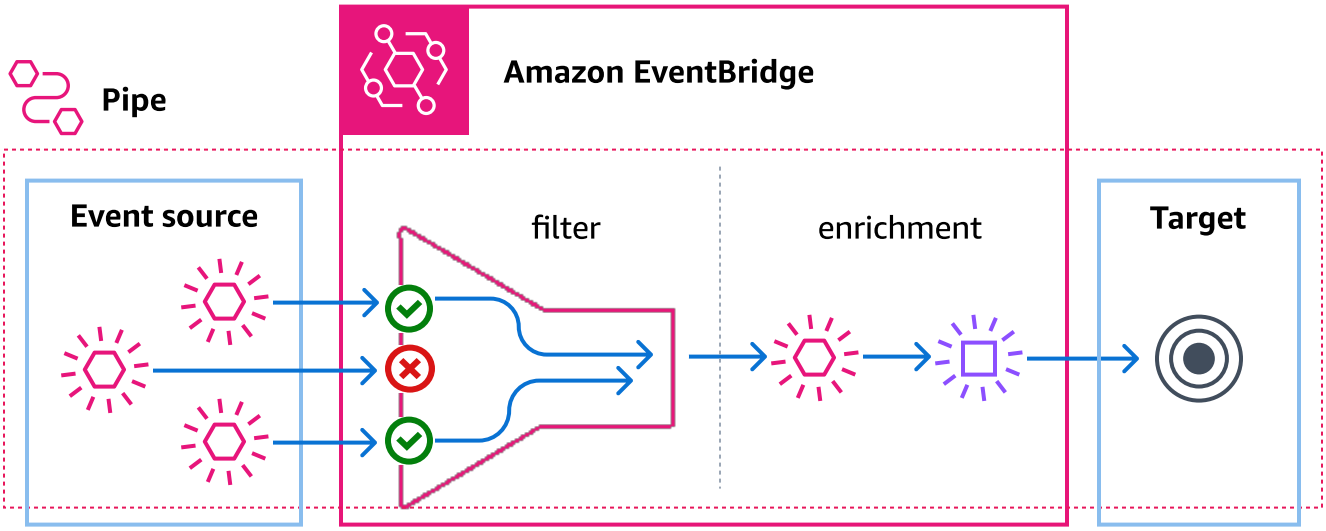
Como funcionam EventBridge os tubos

Em um alto nível, veja como o EventBridge Pipes funciona:

1. Você cria um pipe na sua conta. Isso inclui:
   - Especificar uma das origens de eventos compatíveis da qual deseja que seu pipe receba eventos.
   - Opcionalmente, configurar um filtro para que o pipe processe somente um subconjunto dos eventos que recebe da origem.
   - Opcionalmente, configurar uma etapa de enriquecimento que aprimore os dados do evento antes de enviá-los ao destino.
   - Especificar um dos destinos compatíveis para os quais você deseja que seu pipe envie eventos.
2. A origem do evento começa a enviar eventos para o pipe e o pipe processa o evento antes de enviá-lo ao destino.
   - Se configurou um filtro, o pipe avalia o evento e só o envia para o destino se ele corresponder a esse filtro.

     Somente há cobrança pelos eventos que correspondem ao filtro.
   - Se um enriquecimento foi configurado, o pipe executa esse enriquecimento no evento antes de enviá-lo ao destino.

Se os eventos forem agrupados, o enriquecimento manterá a ordem dos eventos no lote.



Por exemplo, um pipe pode ser usado para criar um sistema de comércio eletrônico. Supõe-se que tenha uma API que contém informações do cliente, como endereços de entrega.

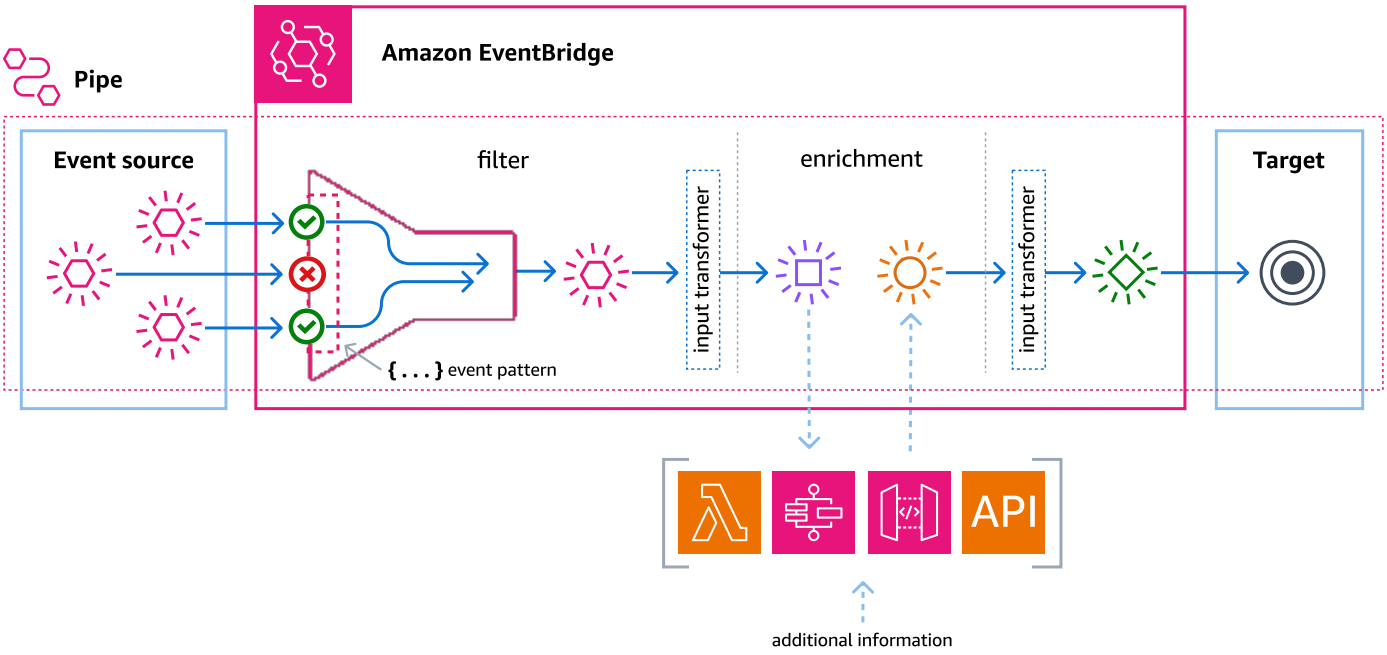
1. É possível pode criar um pipe com o seguinte:
   - Um pedido do Amazon SQS recebeu uma fila de mensagens como origem do evento.
   - Um destino de EventBridge API como enriquecimento
   - Uma máquina de AWS Step Functions estado como alvo
2. Então, quando uma mensagem de pedido recebido do Amazon SQS aparece na fila, ela é enviada para o seu pipe.
3. Em seguida, o canal envia esses dados para o enriquecimento do EventBridge API Destination, que retorna as informações do cliente para esse pedido.
4. Por fim, o tubo envia os dados enriquecidos para a máquina de AWS Step Functions estado, que processa o pedido.

EventBridge Conceitos de tubulações

Aqui está uma análise mais detalhada dos componentes básicos do EventBridge Pipes.

Barra vertical

Um pipe direciona eventos de uma única origem para um único destino. O pipe também inclui a capacidade de filtrar eventos específicos e realizar enriquecimentos nos dados do evento antes de serem enviados ao destino.



Origem

EventBridge O Pipes recebe dados de eventos de várias fontes, aplica filtros opcionais e enriquecimento a esses dados e os envia para um destino. Se uma origem impõe a ordem aos eventos enviados para os pipes, esta ordem é mantida durante todo o processo até o destino.

Para obter mais informações sobre fontes, consulte ???.

Filtros

Um pipe pode filtrar os eventos de uma determinada origem e processar somente um subconjunto desses eventos. Para configurar a filtragem em um pipe, é definido um padrão de evento que o pipe usa para determinar quais eventos enviar para o destino.

Somente há cobrança pelos eventos que correspondem ao filtro.

Para ter mais informações, consulte ???.

Enriquecimento

Com a etapa de enriquecimento do EventBridge Pipes, você pode aprimorar os dados da fonte antes de enviá-los ao destino. Por exemplo, é possível receber eventos Criados pelo tíquete que não incluam os dados completos do tíquete. Ao usar o enriquecimento, é possível fazer com que uma função do Lambda chame a API `get-ticket` para obter os detalhes completos do tíquete. O pipe pode enviar essas informações para um destino.

Para obter mais informações sobre os dados de eventos, consulte ???.

Destino

Depois que os dados do evento forem filtrados e enriquecidos, você poderá especificar o canal para enviá-los para um destino específico, como um stream do Amazon Kinesis ou um grupo de registros da Amazon CloudWatch . Para obter uma lista dos destinos disponíveis, consulte ???.

É possível transformar os dados depois de serem aprimorados e antes de serem enviados pelo pipe para o destino. Para ter mais informações, consulte ???.

Vários pipes, cada um com uma origem diferente, podem enviar eventos ao mesmo destino.

Também é possível usar pipes e barramentos de eventos juntos para enviar eventos para vários destinos. Um caso de uso comum é criar um pipe com um barramento de eventos como destino; o pipe envia eventos para o barramento de eventos, que então envia esses eventos para vários destinos. Por exemplo, é possível criar um pipe com um fluxo do DynamoDB para uma origem e um barramento de eventos como destino. O pipe recebe eventos do fluxo do DynamoDB e os envia para o barramento de eventos, que os envia para vários destinos de acordo com as regras que você especificou no barramento de eventos.

Permissões do Amazon EventBridge Pipes

Ao configurar um pipe, é possível usar um perfil de execução existente ou fazer com que o EventBridge crie um para você com as permissões necessárias. As permissões que o EventBridge Pipes exige variam de acordo com o tipo de origem e estão listadas abaixo. Se estiver configurando seu próprio perfil de execução, deverá adicionar essas permissões.

> **Note**
>
> Se não tiver certeza das permissões exatas e bem definidas necessárias para acessar a origem, use o console do EventBridge Pipes para criar um novo perfil e inspecione as ações listadas na política.

Tópicos

Permissões do perfil de execução do DynamoDB

Em DynamoDB Streams, o EventBridge Pipes exige as seguintes permissões para gerenciar recursos relacionados ao seu fluxo de dados do DynamoDB.

- `dynamodb:DescribeStream`
- `dynamodb:GetRecords`
- `dynamodb:GetShardIterator`
- `dynamodb:ListStreams`

Para enviar registros de lotes com falha para a fila de mensagens não entregues do pipe, sua função de execução do pipe precisa da seguinte permissão:

- `sqs:SendMessage`

Permissões do perfil de execução do Kinesis

No Kinesis, o EventBridge Pipes exige as seguintes permissões para gerenciar recursos relacionados ao seu fluxo de dados do Kinesis.

- kinesis:DescribeStream
- kinesis:DescribeStreamSummary
- kinesis:GetRecords
- kinesis:GetShardIterator
- kinesis:ListShards
- kinesis:ListStreams
- kinesis:SubscribeToShard

Para enviar registros de lotes com falha para a fila de mensagens não entregues do pipe, sua função de execução do pipe precisa da seguinte permissão:

- sqs:SendMessage

Permissões do perfil de execução do Amazon MQ

Para o Amazon MQ, o EventBridge Pipes exige as seguintes permissões para gerenciar recursos relacionados ao seu agente de mensagens do Amazon MQ.

- mq:DescribeBroker
- secretsmanager:GetSecretValue
- ec2:CreateNetworkInterface
- ec2:DeleteNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- logs:CreateLogGroup
- logs:CreateLogStream

- logs:PutLogEvents

Permissões do perfil de execução do Amazon MSK

Para o Amazon MSK, o EventBridge exige as permissões a seguir para gerenciar recursos relacionados ao seu tópico do Amazon MSK.

> **Note**
>
> Se estiver usando a autenticação baseada em perfis do IAM, seu perfil de execução precisará das permissões listadas em ???, além das listadas abaixo.

- kafka:DescribeClusterV2
- kafka:GetBootstrapBrokers
- ec2:CreateNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeVpcs
- ec2:DeleteNetworkInterface
- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- logs:CreateLogGroup
- logs:CreateLogStream
- logs:PutLogEvents

Permissões autogerenciadas do perfil de execução do Apache Kafka

Para o Apache Kafka autogerenciado, o EventBridge exige as seguintes permissões para gerenciar recursos relacionados ao seu fluxo autogerenciado do Apache Kafka.

Permissões obrigatórias

Para criar e armazenar logs em um grupo de logs do Amazon CloudWatch Logs, seu pipe deve ter as seguintes permissões no perfil de execução:

- logs:CreateLogGroup
- logs:CreateLogStream
- logs:PutLogEvents

Permissões opcionais.

Seu pipe também pode precisar dessas permissões para:

- Descreva o segredo do Secrets Manager.
- Acessar a chave gerenciada pelo cliente AWS Key Management Service (AWS KMS)
- Acesse sua Amazon VPC.

Secrets Manager eAWS KMSpermissions

Conforme o tipo de controle de acesso que está sendo configurado para seus agentes do Apache Kafka, seu pipe poderá precisar de permissão para acessar seu segredo do Secrets Manager ou descriptografar sua chave do AWS KMS gerenciada pelo cliente. Para acessar esses recursos, a função de execução da função precisa ter as seguintes permissões:

- secretsmanager:GetSecretValue
- kms:Decrypt

Permissões da VPC

Se somente os usuários dentro de uma VPC puderem acessar seu cluster do Apache Kafka autogerenciado, seu pipe deverá ter permissão para acessar seus recursos da Amazon VPC. Esses recursos incluem sua VPC, sub-redes, security groups e interfaces de rede. Para acessar esses recursos, o perfil de execução do pipe precisa ter as seguintes permissões:

- ec2:CreateNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeVpcs
- ec2:DeleteNetworkInterface
- ec2:DescribeSubnets

- `ec2:DescribeSecurityGroups`

Permissões do perfil de execução do Amazon SQS

Para o Amazon SQS, o EventBridge exige as permissões a seguir para gerenciar recursos relacionados à sua fila do Amazon SQS.

- `sqs:ReceiveMessage`
- `sqs:DeleteMessage`
- `sqs:GetQueueAttributes`

Permissões de enriquecimento e destino

Para fazer chamadas de API para os seus próprios recursos, o EventBridge Pipes precisa da permissão adequada. O EventBridge Pipes usa o perfil do IAM especificado no pipe para enriquecimento e chamadas de destino usando a entidade principal `pipes.amazonaws.com` do IAM.

Criando um EventBridge cachimbo Amazon

EventBridge O Pipes permite criar point-to-point integrações entre fontes e destinos, incluindo enriquecimento e transformações avançadas de eventos. Para criar uma EventBridge tubulação, você executa as seguintes etapas:

1. ???
2. ???
3. ???
4. ???
5. ???

Para obter informações sobre como criar um pipe usando a AWS CLI, consulte create-pipe na Referência de Comandos da AWS CLI.

Como especificar uma origem

Para começar, especifique a origem da qual deseja que o pipe receba eventos.

Para especificar uma origem de pipe usando o console

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Pipes.
3. Escolha Criar pipe.
4. Insira um nome para o pipe.
5. (Opcional) Adicione uma descrição para o pipe.
6. Na guia Criar pipe, em Origem, escolha o tipo de origem que você deseja especificar para esse pipe e configure a origem.

   As propriedades de configuração diferem com base no tipo de origem escolhida:

   Confluent

   Para configurar um stream do Confluent Cloud como fonte, usando o console

   1. Em Source, escolha Confluent Cloud.
   2. Para Servidores bootstrap, insira os endereços de pares `host:port` de seus agentes.
   3. Em Nome do tópico: insira o nome do tópico do qual o pipe lerá.
   4. (Opcional) Em VPC, escolha a VPC que deseja. Em seguida, para sub-redes VPC, escolha as sub-redes desejadas. Em Grupos de segurança, escolha os grupos de segurança.
   5. Para Autenticação - opcional, ative Usar Autenticação e faça o seguinte:
      a. Em Método de autenticação, escolha o tipo de autenticação.
      b. Em Chave secreta, escolha a chave secreta.

      Para obter mais informações, consulte Autenticar nos recursos do Confluent Cloud na documentação do Confluent.
   6. (Opcional) Em Configurações adicionais, faça o seguinte:
      a. Para Posição inicial, escolha uma das seguintes opções:
         - Mais recente: comece a ler o fluxo com o registro mais recente no fragmento.
         - Horizonte de corte: comece a ler o fluxo com o registro mais recente não cortado no fragmento. Este é o registro mais antigo do fragmento.
      b. Em Tamanho do lote: opcional, insira um número máximo de registros para cada lote. O valor padrão é 100.

c. Em Janela de lotes: opcional, insira um número máximo de segundos para coletar registros antes de continuar.

DynamoDB

1. Em Origem, escolha DynamoDB.
2. Para o fluxo do DynamoDB, escolha o fluxo que você deseja usar como origem.
3. Para Posição inicial, escolha uma das seguintes opções:
   - Mais recente: comece a ler o fluxo com o registro mais recente no fragmento.
   - Horizonte de corte: comece a ler o fluxo com o registro mais recente não cortado no fragmento. Este é o registro mais antigo do fragmento.
4. (Opcional) Em Configurações adicionais, faça o seguinte:
   a. Em Tamanho do lote: opcional, insira um número máximo de registros para cada lote. O valor padrão é 100.
   b. Em Janela de lotes: opcional, insira um número máximo de segundos para coletar registros antes de continuar.
   c. Em Lotes simultâneos por fragmento: opcional, insira o número de lotes do mesmo fragmento que podem ser lidos ao mesmo tempo.
   d. Para Em caso de falha parcial do item do lote, escolha o seguinte:
      - AUTOMATIC_BISECT: divida cada lote na metade e faça novas tentativas com cada metade até que todos os registros sejam processados ou que reste apenas uma mensagem com falha no lote.

> **Note**
>
> Se não escolher AUTOMATIC_BISECT, poderá retornar registros específicos com falha e somente aqueles serão repetidos.

Kinesis

Para configurar uma origem do Kinesis usando o console

1. Em Origem, escolha Kinesis.
2. Para o fluxo do Kinesis, escolha o fluxo que você deseja usar como origem.

3. Para Posição inicial, escolha uma das seguintes opções:
   - Mais recente: comece a ler o fluxo com o registro mais recente no fragmento.
   - Horizonte de corte: comece a ler o fluxo com o registro mais recente não cortado no fragmento. Este é o registro mais antigo do fragmento.
   - No timestamp: comece a ler o fluxo a partir de um horário especificado. Em Carimbo de data/hora, insira dados e hora usando os formatos AAAA/MM/DD e hh:mm:ss.
4. (Opcional) Em Configurações adicionais, faça o seguinte:
   a. Em Tamanho do lote: opcional, insira um número máximo de registros para cada lote. O valor padrão é 100.
   b. (Opcional) Para Janela batch: opcional, insira um número máximo de segundos para coletar registros antes de continuar.
   c. Em Lotes simultâneos por fragmento: opcional, insira o número de lotes do mesmo fragmento que podem ser lidos ao mesmo tempo.
   d. Para Em caso de falha parcial do item do lote, escolha o seguinte:
      - AUTOMATIC_BISECT: divida cada lote na metade e faça novas tentativas com cada metade até que todos os registros sejam processados ou que reste apenas uma mensagem com falha no lote.

      > **Note**
      >
      > Se não escolher AUTOMATIC_BISECT, poderá retornar registros específicos com falha e somente aqueles serão repetidos.

Amazon MQ

Para configurar uma origem do Amazon MQ usando o console

1. Em Origem, selecione Amazon MQ.
2. Para o agente do Amazon MQ, escolha o fluxo que deseja usar como origem.
3. Em Nome da fila, insira o nome da fila da qual o pipe lerá.
4. Em Método de autenticação, escolha BASIC_AUTH.
5. Em Chave secreta, escolha a chave secreta.
6. (Opcional) Em Configurações adicionais, faça o seguinte:

a. Em Tamanho do lote: opcional, insira um número máximo de mensagens para cada lote. O valor padrão é 100.

b. Em Janela de lotes: opcional, insira um número máximo de segundos para coletar registros antes de continuar.

Amazon MSK

Para configurar uma origem do Amazon MSK usando o console

1. Em Origem, selecione Amazon MSK.
2. Em cluster do Amazon MSK, escolha o cluster que deseja usar.
3. Em Nome do tópico: insira o nome do tópico do qual o pipe lerá.
4. (Opcional) Em ID do grupo de consumidores: opcional, insira o ID de um grupo de consumidores no qual deseja que o pipe entre.
5. (Opcional) Para Autenticação: opcional, ative Usar Autenticação e faça o seguinte:
   a. Em Método de autenticação, escolha o tipo que deseja.
   b. Em Chave secreta, escolha a chave secreta.
6. (Opcional) Em Configurações adicionais, faça o seguinte:
   a. Em Tamanho do lote: opcional, insira um número máximo de registros para cada lote. O valor padrão é 100.
   b. Em Janela de lotes: opcional, insira um número máximo de segundos para coletar registros antes de continuar.
   c. Para Posição inicial, escolha uma das seguintes opções:
      - Mais recente: comece a ler o tópico com o registro mais recente no fragmento.
      - Horizonte de corte: comece a ler o tópico com o registro mais recente não cortado no fragmento. Este é o registro mais antigo do fragmento.

> **Note**
>
> Horizonte de corte é o mesmo que Mais antigo para Apache Kafka.

Self managed Apache Kafka

Para configurar uma origem autogerenciada do Apache Kafka usando o console

1. Em Origem, escolha Apache Kafka autogerenciado.
2. Para Servidores bootstrap, insira os endereços de pares `host:port` de seus agentes.
3. Em Nome do tópico: insira o nome do tópico do qual o pipe lerá.
4. (Opcional) Em VPC, escolha a VPC que deseja. Em seguida, para sub-redes VPC, escolha as sub-redes desejadas. Em Grupos de segurança, escolha os grupos de segurança.
5. (Opcional) Para Autenticação: opcional, ative Usar Autenticação e faça o seguinte:
   a. Em Método de autenticação, escolha o tipo de autenticação.
   b. Em Chave secreta, escolha a chave secreta.
6. (Opcional) Em Configurações adicionais, faça o seguinte:
   a. Para Posição inicial, escolha uma das seguintes opções:
      - Mais recente: comece a ler o fluxo com o registro mais recente no fragmento.
      - Horizonte de corte: comece a ler o fluxo com o registro mais recente não cortado no fragmento. Este é o registro mais antigo do fragmento.
   b. Em Tamanho do lote: opcional, insira um número máximo de registros para cada lote. O valor padrão é 100.
   c. Em Janela de lotes: opcional, insira um número máximo de segundos para coletar registros antes de continuar.

Amazon SQS

Para configurar uma origem do Amazon SQS usando o console

1. Em Origem, escolha SQS.
2. Na fila SQS, escolha a fila que você deseja usar.
3. (Opcional) Em Configurações adicionais, faça o seguinte:
   a. Em Tamanho do lote: opcional, insira um número máximo de registros para cada lote. O valor padrão é 100.

b. Em Janela de lotes: opcional, insira um número máximo de segundos para coletar registros antes de continuar.

Como configurar a filtragem de eventos (opcional)

É possível adicionar filtragem ao seu pipe para enviar apenas um subconjunto de eventos da sua origem para o destino.

Para configurar a filtragem usando o console

1. Escolha Filtragem.
2. Em Exemplo de evento: opcional, você verá um evento de amostra que pode ser usado para criar seu padrão de evento ou pode inserir seu próprio evento escolhendo Inserir seu próprio evento.
3. Em Padrão do evento, insira o padrão do evento que deseja usar para filtrar os eventos. Para obter mais informações sobre a construção de filtros, consulte???.

   A seguir, um exemplo de padrão de evento que envia somente eventos com o valor Seattle no campo Cidade.

   ```
   {
     "data": {
       "City": ["Seattle"]
     }
   }
   ```

Agora que os eventos estão sendo filtrados, é possível adicionar enriquecimento opcional e um destino para o pipe.

Como definir o enriquecimento do evento (opcional)

Você pode enviar os dados do evento para enriquecimento em uma função Lambda, máquina de estado AWS Step Functions , Amazon API Gateway ou destino de API.

Para selecionar o enriquecimento

1. Escolha Enriquecimento.

2. Em Detalhes, em Serviço, selecione o serviço e as configurações relacionadas que deseja usar para enriquecimento.

Também é possível transformar os dados antes de enviá-los para serem aprimorados.

(Opcional) Para definir o transformador de entrada

1. Escolha o Transformador de entrada de enriquecimento: opcional.
2. Em Amostra de eventos/Carga útil do evento, escolha o tipo de evento de amostra.
3. Para Transformer, insira a sintaxe do transformador, como `"Event happened at <$.detail.field>."` onde `<$.detail.field>` é uma referência a um campo do evento de amostra. Também é possível clicar duas vezes em um campo do evento de amostra para adicioná-lo ao transformador.
4. Em Saída, verifique se a saída tem a aparência desejada.

Agora que os dados foram filtrados e aprimorados, é preciso definir um destino para o qual enviar os dados do evento.

Como configurar um destino

Para configurar um destino

1. Selecione Target.
2. Em Detalhes, para o Serviço de destino, escolha o destino. Os campos que são exibidos variam de acordo com o serviço escolhido. Insira as informações específicas desse tipo de destino conforme necessário.

Também é possível transformar os dados antes de enviá-los ao destino.

(Opcional) Para definir o transformador de entrada

1. Escolha Transformador de entrada de destino: opcional.
2. Em Amostra de eventos/Carga útil do evento, escolha o tipo de evento de amostra.
3. Para Transformer, insira a sintaxe do transformador, como `"Event happened at <$.detail.field>."` onde `<$.detail.field>` é uma referência a um campo do evento

de amostra. Também é possível clicar duas vezes em um campo do evento de amostra para adicioná-lo ao transformador.

4. Em Saída, verifique se a saída tem a aparência desejada.

Agora que o pipe está configurado, verifique se suas configurações estão definidas corretamente.

Como configurar as definições de pipe

Um pipe está ativo por padrão, mas você pode desativá-lo. Também é possível especificar as permissões do pipe, configurar o registro em log de pipes e adicionar tags.

Configurar as definições de pipe

1. Escolha a guia Configurações de pipe.
2. Por padrão, os pipes recém-criados ficam ativos assim que são criados. Se quiser criar um pipe inativo, em Ativação, para Ativar pipe, desative Ativo.
3. Em Permissões, para Perfil de execução, faça o seguinte:

   a. Para EventBridge criar uma nova função de execução para esse canal, escolha Criar uma nova função para esse recurso específico. Em Nome do perfil, você pode, opcionalmente, editar o nome do perfil.

   b. Para um perfil de execução existente, escolha Usar perfil existente. Em Nome do perfil, escolha o perfil.
4. (Opcional) Se você tiver especificado um DynamoDB stream Kinesis or como fonte de canal, poderá configurar uma política de repetição e uma fila de mensagens mortas (DLQ).

   Para Política de repetição e fila de mensagens não entregues: opcional, faça o seguinte:

   Em Revisar política, faça o seguinte:

   a. Se quiser habilitar políticas de novas tentativas, ative Novas tentativas. Por padrão, os pipes recém-criados não têm uma política de repetição ativada.

   b. Em Tempo Máximo do Evento, insira um valor entre um minuto (00:01) e 24 horas (24:00).

   c. Em Tentativas de Repetição, insira um número entre 0 e 185.

   d. Se quiser usar uma fila de mensagens não entregues (DLQ), ative a Fila de mensagens não entregues, escolha o método de sua preferência e escolha a fila ou o tópico que gostaria de usar. Por padrão, os pipes recém-criados não usam uma DLQ.

5. (Opcional) Em Logs: opcional, você pode configurar como o EventBridge Pipes envia informações de registro em log para os serviços compatíveis, incluindo como configurar esses logs.

   Para obter mais informações sobre o registro em log de pipes, consulte ???.

   CloudWatch logs é selecionado como destino de log por padrão, assim como o nível de `ERROR` log. Então, por padrão, o EventBridge Pipes cria um novo CloudWatch grupo de registros para o qual envia registros de log contendo o `ERROR` nível de detalhe.

   Para que o EventBridge Pipes envie registros de log para qualquer um dos destinos de log suportados, faça o seguinte:

   a. Em Registros: opcional, escolha os destinos para os quais deseja que os registros de log sejam entregues.
   b. Em Nível de registro, escolha o nível de informações a EventBridge serem incluídas nos registros de registro. O nível de log `ERROR` é selecionado por padrão.

      Para ter mais informações, consulte ???.
   c. Selecione Incluir dados de execução se EventBridge quiser incluir informações de carga útil do evento e informações de solicitação e resposta de serviço nos registros de log.

      Para ter mais informações, consulte ???.
   d. Configure cada destino de log selecionado:

      Para CloudWatch Logs registros, em CloudWatch registros, faça o seguinte:

      - Para grupo de CloudWatch registros, escolha se deseja EventBridge criar um novo grupo de registros, ou você pode selecionar um grupo de registros existente ou especificar o ARN de um grupo de registros existente.
      - Para novos grupos de logs, edite o nome do grupo de logs conforme desejado.

      CloudWatch os registros são selecionados por padrão.

      Para registros de Firehose transmissão, em Registro de Firehose transmissão, selecione a Firehose transmissão.

      Para Amazon S3 registros, em registros do S3, faça o seguinte:

- Insira o nome do bucket a ser usado como destino do log.
- Insira o ID da AWS conta do proprietário do bucket.
- Insira qualquer texto de prefixo que você queira usar quando o EventBridge criar objetos do S3.

  Para obter informações, consulte Como organizar objetos usando prefixos no Guia do usuário do Amazon Simple Storage Service .
- Escolha como você deseja EventBridge formatar os registros de log do S3:
  - `json`: JSON
  - `plain`: texto sem formatação
  - `w3c`: Formato de arquivo de registro em log estendido do W3C

6. (Opcional) Em Tags: opcional, escolha Adicionar nova tag e insira uma ou mais tags para a regra. Para ter mais informações, consulte ???.
7. Escolha Criar pipe.

Como validar os parâmetros de configuração

Depois que um pipe é criado, EventBridge valida os seguintes parâmetros de configuração:

- Função do IAM — Como a origem de um canal não pode ser alterada após a criação do canal, EventBridge verifica se a função do IAM fornecida pode acessar a fonte.

  > **Note**
  >
  > EventBridge não executa a mesma validação para enriquecimentos ou alvos porque eles podem ser atualizados após a criação do canal.

- Lotes — EventBridge valida se o tamanho do lote da origem não excede o tamanho máximo do lote de destino. Se isso acontecer, será EventBridge necessário um tamanho de lote menor. Além disso, se um destino não suportar o agrupamento em lotes, você não poderá configurar o envio em lotes EventBridge para a origem.
- Enriquecimentos — EventBridge valida que o tamanho do lote para enriquecimentos do API Gateway e do destino da API é 1 porque somente tamanhos de lote de 1 são suportados.

Como iniciar e interromper um pipe

Por padrão, um pipe está `Running` e processa eventos quando é criado.

Se criar um canal com fontes do Amazon SQS, Kinesis ou DynamoDB, a criação do pipe normalmente pode levar um ou dois minutos.

Se criar um pipe com origens do Amazon MSK, do Apache Kafka autogerenciado ou do Amazon MQ, a criação de pipes pode levar até dez minutos.

Para criar um pipe sem processar eventos usando o console

- Desative a configuração Ativar pipe.

Para criar um pipe sem processar eventos programaticamente

- Em sua chamada de API, defina `DesiredState` como `Stopped`.

Para iniciar ou interromper um pipe existente usando o console

- Na guia Configurações de pipes, em Ativação, para Ativar pipe, ative ou desative Ativo.

Para iniciar ou interromper um pipe existente programaticamente

- Em sua chamada de API, defina o parâmetro `DesiredState` como `RUNNING` ou `STOPPED`.

Pode haver um atraso entre o momento em que um pipe está `STOPPED` e o momento em que ele não processa mais os eventos:

- No Amazon SQS e nas origens de fluxo, este atraso geralmente é inferior a dois minutos.
- Para origens do Amazon MQ e do Apache Kafka, esse atraso pode ser de até quinze minutos.

Fontes da Amazon EventBridge Pipes

EventBridge O Pipes recebe dados de eventos de várias fontes, aplica filtros e enriquecimentos opcionais a esses dados e os envia para um destino.

Se uma fonte impõe a ordem aos eventos enviados ao EventBridge Pipes, essa ordem é mantida durante todo o processo até o destino.

Os seguintes AWS serviços podem ser especificados como fontes para EventBridge Pipes:

- Fluxo do Amazon DynamoDB
- Fluxo do Amazon Kinesis
- Agente do Amazon MQ
- Fluxo do Amazon MSK
- Fila do Amazon SQS
- Stream do Apache Kafka

  Ao especificar um stream do Apache Kafka como fonte de pipe, você pode especificar um stream do Apache Kafka que você mesmo gerencia ou gerenciado por um provedor terceirizado, como:

  - Confluent Cloud
  - CloudKarafka
  - Redpanda

Fluxo do Amazon DynamoDB como origem

É possível usar o EventBridge Pipes para receber registros em um fluxo do DynamoDB. Opcionalmente, é possível filtrar ou aprimorar esses registros antes de enviá-los para um destino para processamento. Há configurações específicas para o Amazon DynamoDB Streams que podem ser escolhidas ao configurar o pipe. O EventBridge Pipes mantém a ordem dos registros do fluxo de dados ao enviar esses dados para o destino.

> **Important**
>
> A desativação de um fluxo do DynamoDB que é a origem de um pipe faz com que este pipe se torne inutilizável, mesmo que você habilite o fluxo novamente. Isto pode acontecer porque:
>
> - Não é possível interromper, iniciar ou atualizar um pipe cuja origem está desativada.
> - Não é possível atualizar um pipe com uma nova origem após a criação. Ao reativar um fluxo do DynamoDB, esse fluxo recebe um novo nome do recurso da Amazon (ARN) e não está mais associado ao seu pipe.

Se reativar o fluxo do DynamoDB, precisará criar um novo pipe usando o novo ARN do fluxo.

Evento de exemplo

O exemplo de evento a seguir mostra as informações recebidas pelo pipe. É possível usar esse evento para criar e filtrar seus padrões de eventos ou para definir a transformação de entrada. Nem todos os campos podem ser filtrados. Para mais informações sobre quais campos podem ser filtrados, consulte ???.


```
[
  {
    "eventID": "1",
    "eventVersion": "1.0",
    "dynamodb": {
      "Keys": {
        "Id": {
          "N": "101"
        }
      },
      "NewImage": {
        "Message": {
          "S": "New item!"
        },
        "Id": {
          "N": "101"
        }
      },
      "StreamViewType": "NEW_AND_OLD_IMAGES",
      "SequenceNumber": "111",
      "SizeBytes": 26
    },
    "awsRegion": "us-west-2",
    "eventName": "INSERT",
    "eventSourceARN": "arn:aws:dynamodb:us-east-1:111122223333:table/EventSourceTable",
    "eventSource": "aws:dynamodb"
  },
  {
    "eventID": "2",
    "eventVersion": "1.0",
    "dynamodb": {
```

```
      "OldImage": {
        "Message": {
          "S": "New item!"
        },
        "Id": {
          "N": "101"
        }
      },
      "SequenceNumber": "222",
      "Keys": {
        "Id": {
          "N": "101"
        }
      },
      "SizeBytes": 59,
      "NewImage": {
        "Message": {
          "S": "This item has changed"
        },
        "Id": {
          "N": "101"
        }
      },
      "StreamViewType": "NEW_AND_OLD_IMAGES"
    },
    "awsRegion": "us-west-2",
    "eventName": "MODIFY",
    "eventSourceARN": "arn:aws:dynamodb:us-east-1:111122223333:table/EventSourceTable",
    "eventSource": "aws:dynamodb"
  }
]
```

Fluxos de sondagem e agrupamento em lotes

O EventBridge sonda os fragmentos em seu fluxo do DynamoDB em busca de registros a uma taxa básica de quatro vezes por segundo. Quando os registros estão disponíveis, o EventBridge processa o evento e aguarda o resultado. Se o processamento tiver êxito, o EventBridge continua a sondagem até que ela receba mais registros.

Por padrão, o EventBridge invoca seu pipe assim que os registros estão disponíveis. Se o lote que o EventBridge lê da origem tiver apenas um registro nele, apenas um evento será processado. Para evitar invocar a função com um número pequeno de registros, você pode instruir à origem

dos eventos para fazer o buffer dos registros por até cinco minutos, configurando uma janela de processamento de lotes. Antes de processar os eventos, o EventBridge continua a ler registros da origem de eventos até coletar um lote inteiro, até que a janela de processamento de lotes expire ou até que o lote atinja o limite de carga útil de 6 MB.

Você também pode aumentar a simultaneidade processando vários lotes de cada fragmento em paralelo. O EventBridge pode processar até 10 lotes em cada fragmento simultaneamente. Se aumentar o número de lotes simultâneos por fragmento, o EventBridge ainda garantirá o processamento por ordem no nível da chave de partição.

Configure a configuração `ParallelizationFactor` para processar um fragmento de um fluxo de dados do Kinesis ou do DynamoDB com mais de uma execução do pipe simultaneamente. É possível especificar o número de lotes simultâneos que o Lambda pesquisa de um fragmento por meio de um fator de paralelização de 1 (padrão) a 10. Por exemplo, ao definir `ParallelizationFactor` como 2, você poderá ter no máximo 200 execuções simultâneas do EventBridge Pipe para processar 100 fragmentos de dados do Kinesis. Isso ajuda a aumentar o throughput de processamento quando o volume de dados é volátil e o valor de `IteratorAge` é alto. Observe que o fator de paralelização não funcionará se você estiver usando a agregação do Kinesis.

Posição inicial de sondagem e fluxo

Esteja ciente de que a origem de fluxo durante a criação e as atualizações de pipes é, finamente, consistente.

- Durante a criação do pipe, pode levar alguns minutos para a sondagem de eventos do fluxo iniciar.
- Durante as atualizações de pipe para a configuração de sondagem de origem, pode levar alguns minutos para interromper e reiniciar a sondagem de eventos do fluxo.

Este comportamento significa que, se especificar `LATEST` como posição inicial do fluxo, o mapeamento da origem do evento poderá perder eventos durante a criação ou as atualizações. Para garantir que nenhum evento seja perdido, especifique a posição inicial do fluxo como `TRIM_HORIZON`.

Gerar relatórios de falhas de itens de lote

Quando o EventBridge consome e processa dados de streaming de uma origem, ele definirá checkpoints por padrão no número mais elevado na sequência de um lote, mas somente quando o lote tiver êxito total. Para evitar o reprocessamento de todas as mensagens processadas com êxito

em um lote com falha, é possível configurar o enriquecimento ou o destino para retornar um objeto indicando quais mensagens tiveram êxito e quais falharam. Isso se chama resposta parcial em lote.

Para obter mais informações, consulte ???.

Condições de sucesso e falha

Se retornar qualquer um dos seguintes, o EventBridge tratará um lote como um êxito total:

- Uma lista de `batchItemFailure` vazia
- Uma lista de `batchItemFailure` nula
- Uma `EventResponse` vazia
- Uma `EventResponse` nula

Se retornar qualquer um dos seguintes, o EventBridge tratará um lote como uma falha total:

- Uma string `itemIdentifier` vazia
- Uma `itemIdentifier` nula
- Um `itemIdentifier` com um nome de chave inválido

O EventBridge faz novas tentativas após falhas com base na sua estratégia de repetição.

Fluxo do Amazon Kinesis como uma origem

É possível usar o EventBridge Pipes para receber registros em um fluxo de dados do Kinesis. Opcionalmente, é possível filtrar ou aprimorar esses registros antes de enviá-los para um dos destinos disponíveis para processamento. Há configurações específicas do Kinesis que podem ser escolhidas ao configurar um pipe. O EventBridge Pipes mantém a ordem dos registros do fluxo de dados ao enviar esses dados para o destino.

Uma transmissão de dados do Kinesis é um conjunto de fragmentos. Cada fragmento contém uma sequência de registros de dados. Um consumidor é um aplicativo que processa os dados de um stream de dados do Kinesis. É possível mapear uma EventBridge Pipe para um consumidor de throughput compartilhado (iterador padrão) ou para um consumidor de throughput dedicado com distribuição avançada.

Para iteradores padrão, o EventBridge usa o protocolo HTTP para sondar cada fragmento no fluxo do Kinesis para registros. O pipe compartilha o throughput de leitura com outros consumidores do fragmento.

Para minimizar a latência e maximizar o throughput de leitura, é possível criar um consumidor de fluxo de dados com distribuição avançada. Os consumidores de fluxo obtêm uma conexão dedicada para cada estilhaço que não afeta outros aplicativos que fazem leitura do fluxo. O throughput dedicado pode ajudar se você tiver muitos aplicativos lendo os mesmos dados, ou se você estiver reprocessando um fluxo com grandes registros. O Kinesis envia registros ao EventBridge por meio de HTTP/2. Para obter detalhes sobre fluxos de dados do Kinesis, consulte Como ler dados do Amazon Kinesis Data Streams.

Evento de exemplo

O exemplo de evento a seguir mostra as informações recebidas pelo pipe. É possível usar esse evento para criar e filtrar seus padrões de eventos ou para definir a transformação de entrada. Nem todos os campos podem ser filtrados. Para mais informações sobre quais campos podem ser filtrados, consulte ???.


```
[
  {
    "kinesisSchemaVersion": "1.0",
    "partitionKey": "1",
    "sequenceNumber": "49590338271490256608559692538361571095921575989136588898",
    "data": "SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
    "approximateArrivalTimestamp": 1545084650.987
    "eventSource": "aws:kinesis",
    "eventVersion": "1.0",
    "eventID":
 "shardId-000000000006:49590338271490256608559692538361571095921575989136588898",
    "eventName": "aws:kinesis:record",
    "invokeIdentityArn": "arn:aws:iam::123456789012:role/lambda-role",
    "awsRegion": "us-east-2",
    "eventSourceARN": "arn:aws:kinesis:us-east-2:123456789012:stream/lambda-stream"
  },
  {
    "kinesisSchemaVersion": "1.0",
    "partitionKey": "1",
    "sequenceNumber": "49590338271490256608559692540925702759324208523137515618",
    "data": "VGhpcyBpcyBvbmx5IGEgdGVzdC4=",
    "approximateArrivalTimestamp": 1545084711.166
    "eventSource": "aws:kinesis",
```

```
    "eventVersion": "1.0",
    "eventID":
 "shardId-000000000006:49590338271490256608559692540925702759324208523137515618",
    "eventName": "aws:kinesis:record",
    "invokeIdentityArn": "arn:aws:iam::123456789012:role/lambda-role",
    "awsRegion": "us-east-2",
    "eventSourceARN": "arn:aws:kinesis:us-east-2:123456789012:stream/lambda-stream"
  }
]
```

Fluxos de sondagem e agrupamento em lotes

O EventBridge sonda os fragmentos em seu fluxo do Kinesis em busca de registros a uma taxa básica de quatro vezes por segundo. Quando os registros estão disponíveis, o EventBridge processa o evento e aguarda o resultado. Se o processamento tiver êxito, o EventBridge continua a sondagem até que ela receba mais registros.

Por padrão, o EventBridge invoca seu pipe assim que os registros estão disponíveis. Se o lote que o EventBridge lê da origem tiver apenas um registro nele, apenas um evento será processado. Para evitar invocar a função com um número pequeno de registros, você pode instruir à origem dos eventos para fazer o buffer dos registros por até cinco minutos, configurando uma janela de processamento de lotes. Antes de processar os eventos, o EventBridge continua a ler registros da origem de eventos até coletar um lote inteiro, até que a janela de processamento de lotes expire ou até que o lote atinja o limite de carga útil de 6 MB.

Você também pode aumentar a simultaneidade processando vários lotes de cada fragmento em paralelo. O EventBridge pode processar até 10 lotes em cada fragmento simultaneamente. Se aumentar o número de lotes simultâneos por fragmento, o EventBridge ainda garantirá o processamento por ordem no nível da chave de partição.

Configure a configuração `ParallelizationFactor` para processar um fragmento de um fluxo de dados do Kinesis ou do DynamoDB com mais de uma execução do pipe simultaneamente. É possível especificar o número de lotes simultâneos que o Lambda pesquisa de um fragmento por meio de um fator de paralelização de 1 (padrão) a 10. Por exemplo, ao definir `ParallelizationFactor` como 2, você poderá ter no máximo 200 execuções simultâneas do EventBridge Pipe para processar 100 fragmentos de dados do Kinesis. Isso ajuda a aumentar o throughput de processamento quando o volume de dados é volátil e o valor de `IteratorAge` é alto. Observe que o fator de paralelização não funcionará se você estiver usando a agregação do Kinesis.

Posição inicial de sondagem e fluxo

Esteja ciente de que a origem de fluxo durante a criação e as atualizações de pipes é, finamente, consistente.

- Durante a criação do pipe, pode levar alguns minutos para a sondagem de eventos do fluxo iniciar.
- Durante as atualizações de pipe para a configuração de sondagem de origem, pode levar alguns minutos para interromper e reiniciar a sondagem de eventos do fluxo.

Este comportamento significa que, se especificar `LATEST` como posição inicial do fluxo, o mapeamento da origem do evento poderá perder eventos durante a criação ou as atualizações. Para garantir que nenhum evento seja perdido, especifique a posição inicial do fluxo como `TRIM_HORIZON` ou `AT_TIMESTAMP`.

Gerar relatórios de falhas de itens de lote

Quando o EventBridge consome e processa dados de streaming de uma origem, ele definirá checkpoints por padrão no número mais elevado na sequência de um lote, mas somente quando o lote tiver êxito total. Para evitar o reprocessamento de todas as mensagens processadas com êxito em um lote com falha, é possível configurar o enriquecimento ou o destino para retornar um objeto indicando quais mensagens tiveram êxito e quais falharam. Isso se chama resposta parcial em lote.

Para obter mais informações, consulte ???.

Condições de sucesso e falha

Se retornar qualquer um dos seguintes, o EventBridge tratará um lote como um êxito total:

- Uma lista de `batchItemFailure` vazia
- Uma lista de `batchItemFailure` nula
- Uma `EventResponse` vazia
- Uma `EventResponse` nula

Se retornar qualquer um dos seguintes, o EventBridge tratará um lote como uma falha total:

- Uma string `itemIdentifier` vazia
- Uma `itemIdentifier` nula
- Um `itemIdentifier` com um nome de chave inválido

O EventBridge faz novas tentativas após falhas com base na sua estratégia de repetição.

Agente de mensagens Amazon MQ como uma origem

Você pode usar o EventBridge Pipes para receber registros de um agente de mensagens do Amazon MQ. Opcionalmente, é possível filtrar ou aprimorar esses registros antes de enviá-los para um dos destinos disponíveis para processamento. Há configurações específicas para o Amazon MQ que você pode escolher ao configurar um canal. EventBridge O Pipes mantém a ordem dos registros do agente de mensagens ao enviar esses dados para o destino.

O Amazon MQ é um serviço gerenciado de agente de mensagem para o Apache ActiveMQ e o RabbitMQ. Um agente de mensagens habilita aplicações de software e componentes para se comunicarem usando diferentes linguagens de programação, sistemas operacionais e protocolos de mensagens formais com destinos de eventos de tópicos ou filas.

O Amazon MQ também pode gerenciar instâncias do Amazon Elastic Compute Cloud (Amazon EC2) em seu nome instalando agentes do ActiveMQ ou RabbitMQ. Depois que um agente é instalado, ele fornece diferentes topologias de rede e outras necessidades de infraestrutura para suas instâncias.

A origem do Amazon MQ tem as seguintes restrições de configuração:

- Conta cruzada — EventBridge não suporta processamento entre contas. Você não pode usar EventBridge para processar registros de um agente de mensagens do Amazon MQ que esteja em uma conta diferente AWS .
- Autenticação — Para o ActiveMQ, somente o ActiveMQ é suportado. SimpleAuthenticationPlugin Para RabbitMQ, somente oPLAINMecanismo de autenticação é compatível. Para gerenciar credenciais, use AWS Secrets Manager. Para obter mais informações sobre a autenticação do ActiveMQ, consulte Como integrar agentes do ActiveMQ com o LDAP no Guia do desenvolvedor do Amazon MQ.
- Cota de conexão: os agentes têm um número máximo de conexões permitidas por cada protocolo de nível de transmissão de dados. Essa cota é baseada no tipo de instância do agente. Para obter mais informações, consulte a seção Agentes do *Cotas no Amazon MQ* no Guia do desenvolvedor do Amazon MQ.
- Conectividade: é possível criar agentes em uma nuvem privada virtual (VPC) pública ou privada. Para VPCs privadas, seu pipe precisa de acesso à VPC para receber mensagens.
- Destinos de eventos: somente destinos de fila são compatíveis. No entanto, é possível usar um tópico virtual, que se comporta como um tópico internamente e como uma fila externamente

enquanto interage com seus pipes. Para obter mais informações, consulte Destinos virtuais no site do Apache ActiveMQ e Hosts virtuais nio site do the RabbitMQ.

- Topologia de rede: para o ActiveMQ, somente um agente de instância única ou em espera é aceito por pipe. No RabbitMQ, apenas um agente de instância única ou implantação de cluster é aceito por cada pipe. Os agentes de instância única requerem um endpoint de failover. Para obter mais informações sobre esses modos de implantação do agente, consulte Arquitetura do agente ativo do MQ e Arquitetura do agente de MQ do Rabbit noGuia do desenvolvedor do Amazon MQ.
- Protocolos: os protocolos compatíveis dependem do tipo de integração do Amazon MQ.
  - Para integrações com o ActiveMQ EventBridge , usa OpenWire o protocolo /Java Message Service (JMS) para consumir mensagens. O consumo de mensagens não é compatível com nenhum outro protocolo. EventBridge suporta somente as BytesMessageoperações TextMessagee dentro do protocolo JMS. Para obter mais informações sobre o OpenWire protocolo, consulte o OpenWiresite do Apache ActiveMQ.
  - Para integrações com o RabbitMQ, EventBridge usa o protocolo AMQP 0-9-1 para consumir mensagens. Nenhum outro protocolo é compatível para o consumo de mensagens. Para obter mais informações sobre a implementação do protocolo AMQP 0-9-1 pelo RabbitMQ, consulteAMQP 0-9-1 Guia de referência completono site do RabbitMQ.

EventBridge suporta automaticamente as versões mais recentes do ActiveMQ e do RabbitMQ suportadas pelo Amazon MQ. Para obter as versões compatíveis mais recentes, consulte Notas de versão do Amazon MQ no Guia do desenvolvedor do Amazon MQ.

> **Note**
>
> Por padrão, o Amazon MQ tem uma janela de manutenção semanal para agentes. Durante essa janela de tempo, os agentes não estão disponíveis. Para corretores sem espera, não EventBridge processará mensagens até que a janela termine.

Eventos de exemplo

O exemplo de evento a seguir mostra as informações recebidas pelo pipe. É possível usar esse evento para criar e filtrar seus padrões de eventos ou para definir a transformação de entrada. Nem todos os campos podem ser filtrados. Para mais informações sobre quais campos podem ser filtrados, consulte ???.

ActiveMQ

```
[
  {
    "eventSource": "aws:amq",
    "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
    "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
    "messageType": "jms/text-message",
    "data": "QUJDOkFBQUE=",
    "connectionId": "myJMSCoID",
    "redelivered": false,
    "destination": {
      "physicalname": "testQueue"
    },
    "timestamp": 1598827811958,
    "brokerInTime": 1598827811958,
    "brokerOutTime": 1598827811959
  },
  {
    "eventSource": "aws:amq",
    "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
    "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
    "messageType": "jms/bytes-message",
    "data": "3DTOOW7crj51prgVLQaGQ82S48k=",
    "connectionId": "myJMSCoID1",
    "persistent": false,
    "destination": {
      "physicalname": "testQueue"
    },
    "timestamp": 1598827811958,
    "brokerInTime": 1598827811958,
    "brokerOutTime": 1598827811959
  }
]
```

RabbitMQ

```
[
  {
    "eventSource": "aws:rmq",
```

```
    "eventSourceArn": "arn:aws:mq:us-
west-2:111122223333:broker:pizzaBroker:b-9bcfa592-423a-4942-879d-eb284b418fc8",
    "eventSourceKey": "pizzaQueue::/",
    "basicProperties": {
      "contentType": "text/plain",
      "contentEncoding": null,
      "headers": {
        "header1": {
          "bytes": [
            118,
            97,
            108,
            117,
            101,
            49
          ]
        },
        "header2": {
          "bytes": [
            118,
            97,
            108,
            117,
            101,
            50
          ]
        },
        "numberInHeader": 10
      },
      "deliveryMode": 1,
      "priority": 34,
      "correlationId": null,
      "replyTo": null,
      "expiration": "60000",
      "messageId": null,
      "timestamp": "Jan 1, 1970, 12:33:41 AM",
      "type": null,
      "userId": "AIDACKCEVSQ6C2EXAMPLE",
      "appId": null,
      "clusterId": null,
      "bodySize": 80
    },
    "redelivered": false,
    "data": "eyJ0aW1lb3V0IjowLCJkYXRhIjoiQ1pybWYwR3c4T3Y0YnFMUXhENEUifQ=="
```

```
  }
]
```

Grupo de consumidores

Para interagir com o Amazon MQ, EventBridge crie um grupo de consumidores que pode ler seus corretores do Amazon MQ. O grupo de consumidores é criado com o mesmo ID que um UUID de pipes.

Para fontes do Amazon MQ, agrupa registros em EventBridge lotes e os envia para sua função em uma única carga útil. Para controlar o comportamento, é necessário configurar a janela de lotes e o tamanho do lote. EventBridge extrai mensagens até que ocorra uma das seguintes situações:

- Os registros processados atingem o tamanho máximo da carga útil de 6 MB.
- A janela de processamento de lotes expira.
- O número de registros atinge o tamanho total do lote.

EventBridge converte seu lote em uma única carga útil e, em seguida, invoca sua função. As mensagens não são persistentes nem desserializadas. Em vez disso, o grupo de consumidores os recupera como um BLOB de bytes. Em seguida, o base64 os codifica em uma carga útil JSON. Se o pipe retornar um erro para qualquer uma das mensagens em um lote, EventBridge tente novamente o lote inteiro de mensagens até que o processamento seja bem-sucedido ou as mensagens expirem.

Configuração de rede

Por padrão, os agentes do Amazon MQ são criados com o a sinalização `PubliclyAccessible` definida como falsa. Somente quando `PubliclyAccessible` é definido como verdadeiro que o agente recebe um endereço IP público. Para ter acesso total ao seu pipe, seu agente deve usar um endpoint público ou fornecer acesso à VPC.

Se seu agente do Amazon MQ não estiver acessível ao público, EventBridge deverá ter acesso aos recursos da Amazon Virtual Private Cloud (Amazon VPC) associados ao seu corretor.

- Para acessar a VPC de seus corretores do Amazon MQ EventBridge , você pode usar o acesso de saída à Internet para as sub-redes de sua fonte. Para sub-redes públicas, esse deve ser um gateway NAT gerenciado. Para sub-redes privadas, pode ser um gateway NAT ou o seu próprio NAT. Certifique-se de que o NAT tenha um endereço IP público e possa se conectar à Internet.
- EventBridge O Pipes também suporta a entrega de eventos AWS PrivateLink, permitindo que você envie eventos de uma fonte de eventos localizada em um Amazon Virtual Private Cloud (Amazon

VPC) para um destino do Pipes sem atravessar a Internet pública. Você pode usar o Pipes para pesquisar Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogerenciado e Amazon MQ fontes residentes em uma sub-rede privada sem a necessidade de implantar um gateway de internet, configurar regras de firewall ou configurar servidores proxy.

Para configurar um VPC endpoint, consulte Criar um VPC endpoint no Guia do usuário.AWS PrivateLink Para nome do serviço, selecione`com.amazonaws.region.pipes-data`.

Configure os grupos de segurança da Amazon VPC com as seguintes regras (no mínimo):

- Regras de entrada — Permita todo o tráfego na porta do agente Amazon MQ para os grupos de segurança especificados para sua origem.
- Regras de saída: permitir todo o tráfego na porta 443 para todos os destinos. Permita todo o tráfego na porta do agente Amazon MQ para os grupos de segurança especificados para sua origem.

  As portas do broker incluem:

  - 9092 para texto sem formatação
  - 9094 para TLS
  - 9096 para SASL
  - 9098 para IAM

> **Note**
>
> Sua configuração da Amazon VPC pode ser detectada pela API do Amazon MQ. Não é necessário defini-la durante a configuração.

Tópico do Amazon Managed Streaming for Apache Kafka como uma origem

Você pode usar o EventBridge Pipes para receber registros de um tópico do Amazon Managed Streaming for Apache Kafka (Amazon MSK). Opcionalmente, é possível filtrar ou aprimorar esses registros antes de enviá-los para um dos destinos disponíveis para processamento. Há configurações específicas para o Amazon MSK que você pode escolher ao configurar um canal.

EventBridge O Pipes mantém a ordem dos registros do agente de mensagens ao enviar esses dados para o destino.

O Amazon MSK é um serviço totalmente gerenciado que pode ser usado na criação e a execução de aplicações que usam o Apache Kafka para processar dados de streaming. O Amazon MSK simplifica a configuração, a escalabilidade e o gerenciamento de clusters que executam o Apache Kafka. Com o Amazon MSK, você pode configurar seu aplicativo para várias zonas de disponibilidade e para segurança com AWS Identity and Access Management (IAM). O Amazon MSK é compatível com várias versões de código aberto do Kafka.

O Amazon MSK como fonte opera de forma semelhante ao uso do Amazon Simple Queue Service (Amazon SQS) ou do Amazon Kinesis. EventBridgepesquisa internamente novas mensagens da origem e, em seguida, invoca o alvo de forma síncrona. EventBridge lê as mensagens em lotes e as fornece à sua função como uma carga útil de eventos. O tamanho máximo do lote é configurável. (O valor padrão é de 100 mensagens.)

Para fontes baseadas no Apache Kafka, EventBridge suporta parâmetros de controle de processamento, como janelas de lote e tamanho do lote.

EventBridge lê as mensagens sequencialmente para cada partição. Depois de EventBridge processar cada lote, ele confirma os deslocamentos das mensagens nesse lote. Se o destino do canal retornar um erro para qualquer uma das mensagens em um lote, EventBridge tente novamente o lote inteiro de mensagens até que o processamento seja bem-sucedido ou as mensagens expirem.

EventBridge envia o lote de mensagens no evento quando invoca o destino. O payload do evento contém uma matriz de mensagens. Cada item de array contém detalhes do tópico do Amazon MSK e do identificador de partição, juntamente com um carimbo de data/hora e uma mensagem codificada em base64.

Eventos de exemplo

O exemplo de evento a seguir mostra as informações recebidas pelo pipe. É possível usar esse evento para criar e filtrar seus padrões de eventos ou para definir a transformação de entrada. Nem todos os campos podem ser filtrados. Para mais informações sobre quais campos podem ser filtrados, consulte ???.

```
[
  {
    "eventSource": "aws:kafka",
    "eventSourceArn": "arn:aws:kafka:sa-east-1:123456789012:cluster/
vpc-2priv-2pub/751d2973-a626-431c-9d4e-d7975eb44dd7-2",
```

```
    "eventSourceKey": "mytopic-0",
    "topic": "mytopic",
    "partition": "0",
    "offset": 15,
    "timestamp": 1545084650987,
    "timestampType": "CREATE_TIME",
    "key":"abcDEFghiJKLmnoPQRstuVWXyz1234==",
    "value":"SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
    "headers": [
      {
        "headerKey": [
          104,
          101,
          97,
          100,
          101,
          114,
          86,
          97,
          108,
          117,
          101
        ]
      }
    ]
  }
]
```

Posição inicial de sondagem e fluxo

Esteja ciente de que a origem de fluxo durante a criação e as atualizações de pipes é, finamente, consistente.

- Durante a criação do pipe, pode levar alguns minutos para a sondagem de eventos do fluxo iniciar.
- Durante as atualizações de pipe para a configuração de sondagem de origem, pode levar alguns minutos para interromper e reiniciar a sondagem de eventos do fluxo.

Este comportamento significa que, se especificar `LATEST` como posição inicial do fluxo, o mapeamento da origem do evento poderá perder eventos durante a criação ou as atualizações. Para garantir que nenhum evento seja perdido, especifique a posição inicial do fluxo como `TRIM_HORIZON`.

Autenticação de cluster do MSK

EventBridge precisa de permissão para acessar o cluster Amazon MSK, recuperar registros e realizar outras tarefas. O Amazon MSK oferece suporte a várias opções para controlar o acesso do cliente ao cluster do MSK. Para obter mais informações quando cada método de autenticação é usado, consulte ???.

Opções de acesso ao cluster

- Acesso não autenticado
- Autenticação SASL/SCRAM
- Autenticação baseada em função do IAM
- Autenticação TLS mútua
- Configurar o segredo de mTLS
- Como EventBridge escolhe um corretor de bootstrap

Acesso não autenticado

É recomendado usar somente o acesso não autenticado para desenvolvimento. O acesso não autenticado só funcionará se a autenticação baseada em perfis do IAM estiver desativada para o cluster.

Autenticação SASL/SCRAM

O Amazon MSK é compatível com autenticação Simple Authentication e Security Layer/Salted Challenge Response Authentication Mechanism (SASL/SCRAM) com criptografia Transport Layer Security (TLS). EventBridge Para se conectar ao cluster, você armazena as credenciais de autenticação (credenciais de login) em um segredo. AWS Secrets Manager

Para obter mais informações sobre o uso do Secrets Manager, consulte Autenticação de nome de usuário e senha com o AWS Secrets Manager, no Guia do Desenvolvedor do Amazon Managed Streaming para Apache Kafka.

O Amazon MSK não oferece suporte a autenticação SASL/PLAIN.

Autenticação baseada em função do IAM

Use o IAM para autenticar a identidade dos clientes que se conectam ao cluster do MSK. Se a autenticação do IAM estiver ativa no seu cluster MSK e você não fornecer um segredo para a

autenticação, EventBridge automaticamente usará a autenticação do IAM como padrão. Para criar e implantar políticas baseadas em funções ou usuários do IAM, use o console ou a API do IAM. Para obter mais informações, consulte IAM access control (Controle de acesso do IAM) no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

EventBridge Para permitir a conexão com o cluster MSK, a leitura de registros e a execução de outras ações necessárias, adicione as seguintes permissões à função de execução de seus pipes.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:DescribeGroup",
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeTopic",
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeClusterDynamicConfiguration"
            ],
            "Resource": [
                "arn:aws:kafka:region:account-id:cluster/cluster-name/cluster-uuid",
                "arn:aws:kafka:region:account-id:topic/cluster-name/cluster-uuid/topic-
name",
                "arn:aws:kafka:region:account-id:group/cluster-name/cluster-
uuid/consumer-group-id"
            ]
        }
    ]
}
```

É possível definir o escopo dessas permissões para abranger clusters, tópicos e grupos específicos. Para obter mais informações, consulte Amazon MSK Kafka actions (Ações do Amazon MSK Kafka) no Guia do desenvolvedor Amazon Managed Streaming for Apache Kafka.

Autenticação TLS mútua

O TLS mútuo (mTLS) fornece autenticação bidirecional entre o cliente e o servidor. O cliente envia um certificado ao servidor para que o servidor verifique o cliente, e o servidor envia um certificado ao cliente para que o cliente verifique o servidor.

Para o Amazon MSK, EventBridge atua como cliente. Você configura um certificado de cliente (como um segredo no Secrets Manager) para se autenticar EventBridge com os corretores em seu cluster MSK. O certificado do servidor deve ser assinado por uma autoridade de certificação (CA) no armazenamento de confiança da . O cluster MSK envia um certificado de servidor para EventBridge autenticar os corretores com. EventBridge O certificado do servidor deve ser assinado por uma CA que esteja no armazenamento AWS confiável.

O Amazon MSK não oferece suporte a certificados de servidor autoassinados, porque todos os corretores do Amazon MSK usam certificados públicos assinados pelas CAs do Amazon Trust Services, que EventBridge são confiáveis por padrão.

Para obter mais informações sobre o mTLS para o Amazon MSK, consulte Mutual TLS Authentication (Autenticação TLS mútua) no Guia do desenvolvedor Amazon Managed Streaming for Apache Kafka.

Configurar o segredo de mTLS

O segredo CLIENT_CERTIFICATE_TLS_AUTH requer um campo de certificado e um campo de chave privada. Para uma chave privada criptografada, o segredo requer uma senha de chave privada. Tanto o certificado como a chave privada devem estar no formato PEM.

> **Note**
>
> EventBridge suporta os algoritmos de criptografia de chave privada PBES1 (mas não PBES2).

O campo certificate (certificado) deve conter uma lista de certificados, começando pelo certificado do cliente, seguido por quaisquer certificados intermediários e terminando com o certificado raiz. Cada certificado deve iniciar em uma nova linha com a seguinte estrutura:

```
-----BEGIN CERTIFICATE-----
        <certificate contents>
-----END CERTIFICATE-----
```

O Secrets Manager oferece suporte a segredos de até 65.536 bytes, que é espaço suficiente para cadeias de certificados longas.

A chave privada deve estar no formato PKCS #8, com a seguinte estrutura:

```
-----BEGIN PRIVATE KEY-----
         <private key contents>
-----END PRIVATE KEY-----
```

Para uma chave privada criptografada, use a seguinte estrutura:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
          <private key contents>
-----END ENCRYPTED PRIVATE KEY-----
```

O exemplo a seguir exibe o conteúdo de um segredo para autenticação mTLS usando uma chave privada criptografada. Para uma chave privada criptografada, inclua a senha da chave privada no segredo.

```
{
 "privateKeyPassword": "testpassword",
 "certificate": "-----BEGIN CERTIFICATE-----
MIIE5DCCAsygAwIBAgIRAPJdwaFaNRrytHBto0j5BA0wDQYJKoZIhvcNAQELBQAw
...
j0Lh4/+1HfgyE2KlmII36dg4IMzNjAFEBZiCRoPimO40s1cRqtFHXoal0QQbIlxk
cmUuiAii9R0=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFgjCCA2qgAwIBAgIQdjNZd6uFf9hbNC5RdfmHrzANBgkqhkiG9w0BAQsFADBb
...
rQoiowbbk5wXCheYSANQIfTZ6weQTgiCHCCbuuMKNVS95FkXm0vqVD/YpXKwA/no
c8PH3PSoAaRwMMgOSA2ALJvbRz8mpg==
-----END CERTIFICATE-----",
 "privateKey": "-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFKzBVBgkqhkiG9w0BBQ0wSDAnBgkqhkiG9w0BBQwwGgQUiAFcK5hT/X7Kjmgp
...
QrSekqF+kWzmB6nAfSzgO9IaoAaytLvNgGTckWeUkWn/V0Ck+LdGUXzAC4RxZnoQ
zp2mwJn2NYB7AZ7+imp0azDZb+8YG2aUCiyqb6PnnA==
-----END ENCRYPTED PRIVATE KEY-----"
}
```

Como EventBridge escolhe um corretor de bootstrap

EventBridge escolhe um [agente de bootstrap] com base nos métodos de autenticação disponíveis em seu cluster e se você fornece um segredo para autenticação. Se você fornecer um segredo para mTLS ou SASL/SCRAM, escolhe EventBridge automaticamente esse método de autenticação. Se

você não fornecer um segredo, EventBridge escolha o método de autenticação mais forte que esteja ativo no seu cluster. A seguir está a ordem de prioridade na qual EventBridge seleciona um corretor, da autenticação mais forte para a mais fraca:

- mTLs (segredo fornecido para mTLs)
- SASL/SCRAM (segredo fornecido para SASL/SCRAM)
- SASL IAM (nenhum segredo fornecido, e a autenticação do IAM está ativa)
- TLS não autenticado (nenhum segredo fornecido, e a autenticação do IAM não está ativa)
- Texto simples (nenhum segredo fornecido, e a autenticação do IAM e o TLS não autenticado não estão ativos)

> **Note**
>
> Se não EventBridge conseguir se conectar ao tipo de corretor mais seguro, ele não tentará se conectar a um tipo de corretor diferente (mais fraco). Se você quiser EventBridge escolher um tipo de agente mais fraco, desative todos os métodos de autenticação mais fortes em seu cluster.

Configuração de rede

EventBridge deve ter acesso aos recursos da Amazon Virtual Private Cloud (Amazon VPC) associados ao seu cluster Amazon MSK.

- Para acessar a VPC do seu cluster Amazon MSK, EventBridge você pode usar o acesso de saída à Internet para as sub-redes da sua fonte. Para sub-redes públicas, esse deve ser um gateway NAT gerenciado. Para sub-redes privadas, pode ser um gateway NAT ou o seu próprio NAT. Certifique-se de que o NAT tenha um endereço IP público e possa se conectar à Internet.
- EventBridge O Pipes também suporta a entrega de eventos AWS PrivateLink, permitindo que você envie eventos de uma fonte de eventos localizada em um Amazon Virtual Private Cloud (Amazon VPC) para um destino do Pipes sem atravessar a Internet pública. Você pode usar o Pipes para pesquisar Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogerenciado e Amazon MQ fontes residentes em uma sub-rede privada sem a necessidade de implantar um gateway de internet, configurar regras de firewall ou configurar servidores proxy.

  Para configurar um VPC endpoint, consulte Criar um VPC endpoint no Guia do usuário.AWS PrivateLink Para nome do serviço, selecione`com.amazonaws.region.pipes-data`.

Configure os grupos de segurança da Amazon VPC com as seguintes regras (no mínimo):

- Regras de entrada — Permita todo o tráfego na porta do agente Amazon MSK para os grupos de segurança especificados para sua origem.
- Regras de saída: permitir todo o tráfego na porta 443 para todos os destinos. Permita todo o tráfego na porta do agente Amazon MSK para os grupos de segurança especificados para sua origem.

  As portas do broker incluem:

  - 9092 para texto simples
  - 9094 para TLS
  - 9096 para SASL
  - 9098 para IAM

> **Note**
>
> Sua configuração da Amazon VPC pode ser detectada pela API do Amazon MSK. Não é necessário defini-la durante a configuração.

ID de grupo de consumidores personalizável

Ao configurar o Apache Kafka como uma origem de eventos, é possível especificar um ID de grupo de consumidores. Este ID de grupo de consumidores é um identificador existente para o grupo de consumidores do Apache Kafka no qual deseja que a função do Lambda ingresse. Você pode usar esse recurso para migrar qualquer configuração contínua de processamento de registros do Apache Kafka de outros consumidores para o. EventBridge

Se você especificar um ID de grupo de consumidores e houver outros pesquisadores ativos dentro desse grupo de consumidores, o Apache Kafka distribuirá mensagens entre todos os consumidores. Em outras palavras, EventBridge não recebe todas as mensagens do tópico Apache Kafka. Se você quiser EventBridge lidar com todas as mensagens do tópico, desative qualquer outra pesquisa nesse grupo de consumidores.

Além disso, se você especificar um ID de grupo de consumidores e o Apache Kafka encontrar um grupo de consumidores válido existente com o mesmo ID, EventBridge ignorará o `StartingPosition` parâmetro do seu canal. Em vez disso, EventBridge começa a processar

os registros de acordo com a compensação comprometida do grupo de consumidores. Se você especificar um ID de grupo de consumidores e o Apache Kafka não conseguir encontrar um grupo de consumidores existente, então EventBridge configura sua fonte com o especificado. `StartingPosition`

O ID do grupo de consumidores que você especificar deverá ser exclusivo entre todas as origens de eventos do Apache Kafka. Após criar um pipe com o ID do grupo de consumidores especificado, não poderá atualizar este valor.

Ajuste de escala automático da origem do Amazon MSK

Quando você cria inicialmente uma fonte do Amazon MSK, EventBridge aloca um consumidor para processar todas as partições no tópico Apache Kafka. Cada consumidor conta com vários processadores em execução em paralelo para lidar com um aumento de workloads. Além disso, aumenta ou diminui EventBridge automaticamente o número de consumidores, com base na carga de trabalho. Para preservar a ordenação de mensagens em cada partição, o número máximo de consumidores é um consumidor por partição no tópico.

Em intervalos de um minuto, EventBridge avalia o atraso de compensação do consumidor de todas as partições no tópico. Se o atraso for muito alto, a partição está recebendo mensagens mais rápido do que EventBridge pode processá-las. Se necessário, EventBridge adiciona ou remove consumidores do tópico. O processo de escalabilidade de adicionar ou remover consumidores ocorre em até três minutos após a avaliação.

Se sua meta estiver sobrecarregada, EventBridge reduz o número de consumidores. Essa ação reduz a workload no pipe, reduzindo o número de mensagens que os consumidores podem recuperar e enviar para o pipe.

Streams do Apache Kafka como fonte

O Apache Kafka é uma plataforma de streaming de eventos de código aberto que é compatível com workloads, como pipelines de dados e análises de streaming. Você pode usar o Amazon Managed Streaming for Apache Kafka (Amazon MSK) ou um cluster autogerenciado do Apache Kafka. Na AWS terminologia, um cluster autogerenciado se refere a qualquer cluster Apache Kafka não hospedado pelo. AWS Isso inclui tanto os clusters que você mesmo gerencia quanto os hospedados por um provedor terceirizado Confluent Cloud, como CloudKarafka, ou Redpanda.

Para obter mais informações sobre outras opções de AWS hospedagem para seu cluster, consulte Melhores práticas para executar o Apache Kafka AWS no blog de AWS Big Data.

O Apache Kafka, como fonte, opera de forma semelhante ao uso do Amazon Simple Queue Service (Amazon SQS) ou do Amazon Kinesis. EventBridgepesquisa internamente novas mensagens da origem e, em seguida, invoca o alvo de forma síncrona. EventBridge lê as mensagens em lotes e as fornece à sua função como uma carga útil de eventos. O tamanho máximo do lote é configurável. (O valor padrão é de 100 mensagens.)

Para fontes baseadas no Apache Kafka, EventBridge suporta parâmetros de controle de processamento, como janelas de lotes e tamanho do lote.

EventBridge envia o lote de mensagens no parâmetro event quando invoca seu canal. O payload do evento contém uma matriz de mensagens. Cada item da matriz contém detalhes do tópico do Apache Kafka e do identificador de partição do Apache Kafka, juntamente com um carimbo de data/hora e uma mensagem codificada em base64.

Eventos de exemplo

O exemplo de evento a seguir mostra as informações recebidas pelo pipe. É possível usar esse evento para criar e filtrar seus padrões de eventos ou para definir a transformação de entrada. Nem todos os campos podem ser filtrados. Para mais informações sobre quais campos podem ser filtrados, consulte ???.

```
[
  {
    "eventSource": "SelfManagedKafka",
    "bootstrapServers": "b-2.demo-cluster-1.a1bcde.c1.kafka.us-
east-1.amazonaws.com:9092,b-1.demo-cluster-1.a1bcde.c1.kafka.us-
east-1.amazonaws.com:9092",
    "eventSourceKey": "mytopic-0",
    "topic": "mytopic",
    "partition": 0,
    "offset": 15,
    "timestamp": 1545084650987,
    "timestampType": "CREATE_TIME",
    "key":"abcDEFghiJKLmnoPQRstuVWXyz1234==",
    "value":"SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
    "headers": [
      {
        "headerKey": [
          104,
          101,
          97,
```

```
          100,
          101,
          114,
          86,
          97,
          108,
          117,
          101
        ]
      }
    ]
  }
]
```

Autenticação de clusters do Apache Kafka

EventBridge O Pipes suporta vários métodos de autenticação com seu cluster Apache Kafka autogerenciado. Configure o cluster Apache Kafka para utilizar um dos métodos de autenticação compatíveis. Para obter mais informações sobre a segurança do Apache Kafka, consulte a seção Segurança da documentação do Apache Kafka.

Acesso por VPC

Se você estiver usando um ambiente autogerenciado do Apache Kafka em que somente os usuários do Apache Kafka na sua VPC têm acesso aos seus corretores do Apache Kafka, você deve configurar a Amazon Virtual Private Cloud (Amazon VPC) na fonte do Apache Kafka.

Autenticação SASL/SCRAM

EventBridge O Pipes oferece suporte à autenticação Simple Authentication and Security Layer/Salted Challenge Response Authentication Mechanism (SASL/SCRAM) com criptografia Transport Layer Security (TLS). EventBridge O Pipes envia as credenciais criptografadas para autenticação no cluster. Para obter mais informações sobre a autenticação do SASL/SCRAM, consulte RFC 5802.

EventBridge O Pipes suporta autenticação SASL/PLAIN com criptografia TLS. Com a autenticação SASL/PLAIN, o EventBridge Pipes envia credenciais como texto não criptografado para o servidor.

Para a autenticação SASL, armazene as credenciais de login como um segredo no AWS Secrets Manager.

Autenticação TLS mútua

O TLS mútuo (mTLS) fornece autenticação bidirecional entre o cliente e o servidor. O cliente envia um certificado ao servidor para que o servidor verifique o cliente, e o servidor envia um certificado ao cliente para que o cliente verifique o servidor.

No Apache Kafka autogerenciado, EventBridge Pipes atua como cliente. Você configura um certificado de cliente (como um segredo no Secrets Manager) para autenticar EventBridge Pipes com seus corretores Apache Kafka. O certificado do servidor deve ser assinado por uma autoridade de certificação (CA) no armazenamento de confiança da .

O cluster Apache Kafka envia um certificado de servidor ao EventBridge Pipes para autenticar os corretores do Apache Kafka com o Pipes. EventBridge O certificado do servidor pode ser um certificado CA público ou um certificado de CA privado/autoassinado. O certificado público de CA deve ser assinado por uma CA que esteja no repositório confiável do EventBridge Pipes. Para uma CA privada/certificado autoassinado, você configura o certificado CA raiz do servidor (como um segredo no Secrets Manager). EventBridge O Pipes usa o certificado raiz para verificar os corretores Apache Kafka.

Para obter mais informações sobre o mTLS, consulte Introdução à autenticação TLS mútua para o Amazon MSK como origem.

Configurar o segredo do certificado do cliente

O segredo CLIENT_CERTIFICATE_TLS_AUTH requer um campo de certificado e um campo de chave privada. Para uma chave privada criptografada, o segredo requer uma senha de chave privada. Tanto o certificado como a chave privada devem estar no formato PEM.

> **Note**
>
> EventBridge O Pipes suporta os algoritmos de criptografia de chave privada PBES1 (mas não PBES2).

O campo certificate (certificado) deve conter uma lista de certificados, começando pelo certificado do cliente, seguido por quaisquer certificados intermediários e terminando com o certificado raiz. Cada certificado deve iniciar em uma nova linha com a seguinte estrutura:

```
-----BEGIN CERTIFICATE-----
```

```
        <certificate contents>
-----END CERTIFICATE-----
```

O Secrets Manager oferece suporte a segredos de até 65.536 bytes, que é espaço suficiente para cadeias de certificados longas.

A chave privada deve estar no formato PKCS #8, com a seguinte estrutura:

```
-----BEGIN PRIVATE KEY-----
         <private key contents>
-----END PRIVATE KEY-----
```

Para uma chave privada criptografada, use a seguinte estrutura:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
          <private key contents>
-----END ENCRYPTED PRIVATE KEY-----
```

O exemplo a seguir exibe o conteúdo de um segredo para autenticação mTLS usando uma chave privada criptografada. Para uma chave privada criptografada, inclua a senha de chave privada no segredo.

```
{
 "privateKeyPassword": "testpassword",
 "certificate": "-----BEGIN CERTIFICATE-----
MIIE5DCCAsygAwIBAgIRAPJdwaFaNRrytHBto0j5BA0wDQYJKoZIhvcNAQELBQAw
...
j0Lh4/+1HfgyE2KlmII36dg4IMzNjAFEBZiCRoPimO40s1cRqtFHXoal0QQbIlxk
cmUuiAii9R0=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFgjCCA2qgAwIBAgIQdjNZd6uFf9hbNC5RdfmHrzANBgkqhkiG9w0BAQsFADBb
...
rQoiowbbk5wXCheYSANQIfTZ6weQTgiCHCCbuuMKNVS95FkXm0vqVD/YpXKwA/no
c8PH3PSoAaRwMMgOSA2ALJvbRz8mpg==
-----END CERTIFICATE-----",
 "privateKey": "-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFKzBVBgkqhkiG9w0BBQ0wSDAnBgkqhkiG9w0BBQwwGgQUiAFcK5hT/X7Kjmgp
...
QrSekqF+kWzmB6nAfSzgO9IaoAaytLvNgGTckWeUkWn/V0Ck+LdGUXzAC4RxZnoQ
zp2mwJn2NYB7AZ7+imp0azDZb+8YG2aUCiyqb6PnnA==
```

```
-----END ENCRYPTED PRIVATE KEY-----"
}
```

Configurar o segredo do certificado CA raiz do servidor

Este segredo é criado se seus agentes do Apache Kafka usarem criptografia TLS com certificados assinados por uma CA privada. É possível usar criptografia TLS para autenticação de VPC, SASL/SCRAM, SASL/PLAIN ou mTLS.

O segredo do certificado CA raiz do servidor requer um campo que contenha o certificado CA raiz do agente do Apache Kafka no formato PEM. O exemplo a seguir exibe a estrutura do segredo.

```
{
     "certificate": "-----BEGIN CERTIFICATE-----
  MIID7zCCAtegAwIBAgIBADANBgkqhkiG9w0BAQsFADCBmDELMAkGA1UEBhMCVVMx
  EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxJTAjBgNVBAoT
  HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xOzA5BgNVBAMTMlN0YXJmaWVs
  ZCBTZXJ2aWNlcyBSb290IENlcnRpZmljYXRlIEF1dG...
  -----END CERTIFICATE-----"
```

Configuração de rede

Se você estiver usando um ambiente autogerenciado do Apache Kafka que usa conectividade VPC privada, EventBridge deve ter acesso aos recursos da Amazon Virtual Private Cloud (Amazon VPC) associados aos seus corretores do Apache Kafka.

- Para acessar a VPC do seu cluster Apache Kafka, você EventBridge pode usar o acesso de saída à Internet para as sub-redes da sua fonte. Para sub-redes públicas, esse deve ser um gateway NAT gerenciado. Para sub-redes privadas, pode ser um gateway NAT ou o seu próprio NAT. Certifique-se de que o NAT tenha um endereço IP público e possa se conectar à Internet.
- EventBridge O Pipes também suporta a entrega de eventos AWS PrivateLink, permitindo que você envie eventos de uma fonte de eventos localizada em um Amazon Virtual Private Cloud (Amazon VPC) para um destino do Pipes sem atravessar a Internet pública. Você pode usar o Pipes para pesquisar Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogerenciado e Amazon MQ fontes residentes em uma sub-rede privada sem a necessidade de implantar um gateway de internet, configurar regras de firewall ou configurar servidores proxy.

  Para configurar um VPC endpoint, consulte Criar um VPC endpoint no Guia do usuário.AWS PrivateLink Para nome do serviço, selecione`com.amazonaws.`*`region`*`.pipes-data`.

Configure os grupos de segurança da Amazon VPC com as seguintes regras (no mínimo):

- Regras de entrada — Permita todo o tráfego na porta do broker Apache Kafka para os grupos de segurança especificados para sua fonte.
- Regras de saída: permitir todo o tráfego na porta 443 para todos os destinos. Permita todo o tráfego na porta do agente Apache Kafka para os grupos de segurança especificados para sua fonte.

  As portas do broker incluem:

  - 9092 para texto sem formatação
  - 9094 para TLS
  - 9096 para SASL
  - 9098 para IAM

Dimensionamento automático do consumidor com fontes do Apache Kafka

Quando você cria inicialmente uma fonte do Apache Kafka, EventBridge aloca um consumidor para processar todas as partições no tópico do Kafka. Cada consumidor conta com vários processadores em execução em paralelo para lidar com um aumento de workloads. Além disso, aumenta ou diminui EventBridge automaticamente o número de consumidores, com base na carga de trabalho. Para preservar a ordenação de mensagens em cada partição, o número máximo de consumidores é um consumidor por partição no tópico.

Em intervalos de um minuto, EventBridge avalia o atraso de compensação do consumidor de todas as partições no tópico. Se o atraso for muito alto, a partição está recebendo mensagens mais rápido do que EventBridge pode processá-las. Se necessário, EventBridge adiciona ou remove consumidores do tópico. O processo de escalabilidade de adicionar ou remover consumidores ocorre em até três minutos após a avaliação.

Se sua meta estiver sobrecarregada, EventBridge reduz o número de consumidores. Essa ação reduz a workload na função, reduzindo o número de mensagens que os consumidores podem recuperar e enviar para a função.

Amazon Simple Queue Service como uma origem

Você pode usar o EventBridge Pipes para receber registros de uma fila do Amazon SQS. Opcionalmente, é possível filtrar ou aprimorar esses registros antes de enviá-los para um destino disponível para processamento.

Você pode usar um canal para processar mensagens em uma fila do Amazon Simple Queue Service (Amazon SQS). EventBridge Os Pipes suportam filas padrão e filas de primeiro a entrar, primeiro a sair (FIFO). Com o Amazon SQS, você pode descarregar tarefas de um componente do aplicativo enviando-as a uma fila e processando-as de forma assíncrona.

EventBridge pesquisa a fila e invoca seu canal de forma síncrona com um evento que contém mensagens da fila. EventBridge lê mensagens em lotes e invoca seu canal uma vez para cada lote. Quando seu pipe processa com sucesso um lote, EventBridge exclui suas mensagens da fila.

Por padrão, EventBridge pesquisa até 10 mensagens em sua fila simultaneamente e envia esse lote para seu canal. Para evitar invocar a função com um número pequeno de registros, você pode instruir à origem dos eventos para fazer o buffer dos registros por até cinco minutos, configurando uma janela de lote. Antes de invocar o pipe, EventBridge continue a pesquisar mensagens da fila padrão do Amazon SQS até que uma dessas coisas ocorra:

- A janela de lotes expira.
- A cota de tamanho da carga útil de invocação foi atingida.
- O tamanho máximo do lote configurado foi atingido.

> **Note**
>
> Se você estiver usando uma janela de lote e sua fila do Amazon SQS contiver pouco tráfego, EventBridge espere até 20 segundos antes de invocar seu canal. Isto será válido mesmo se definir uma janela de lote inferior a 20 segundos. Para as filas FIFO, os registros contêm atributos adicionais relacionados a desduplicação e sequenciamento.

Ao EventBridge ler um lote, as mensagens permanecem na fila, mas ficam ocultas durante o tempo limite de visibilidade da fila. Se seu pipe processar o lote com sucesso, EventBridge excluirá as mensagens da fila. Por padrão, se seu pipe se deparar com um erro durante o processamento de um lote, todas as mensagens naquele lote ficarão visíveis na fila novamente. Por conta disso, seu pipe deve ter a capacidade de processar a mesma mensagem várias vezes sem causar efeitos colaterais não intencionais. É possível modificar esse comportamento de reprocessamento incluindo falhas de item de lote na resposta do pipe. O exemplo a seguir mostra um evento para um lote de duas mensagens.

Eventos de exemplo

O exemplo de evento a seguir mostra as informações recebidas pelo pipe. É possível usar esse evento para criar e filtrar seus padrões de eventos ou para definir a transformação de entrada. Nem todos os campos podem ser filtrados. Para mais informações sobre quais campos podem ser filtrados, consulte ???.

Fila padrão

```
[
  {
    "messageId": "059f36b4-87a3-44ab-83d2-661975830a7d",
    "receiptHandle": "AQEBwJnKyrHigUMZj6rYigCgxlaS3SLy0a...",
    "body": "Test message.",
    "attributes": {
      "ApproximateReceiveCount": "1",
      "SentTimestamp": "1545082649183",
      "SenderId": "AIDAIENQZJOLO23YVJ4VO",
      "ApproximateFirstReceiveTimestamp": "1545082649185"
    },
    "messageAttributes": {},
    "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
    "eventSource": "aws:sqs",
    "eventSourceARN": "arn:aws:sqs:us-east-2:123456789012:my-queue",
    "awsRegion": "us-east-2"
  },
  {
    "messageId": "2e1424d4-f796-459a-8184-9c92662be6da",
    "receiptHandle": "AQEBzWwaftRI0KuVm4tP+/7q1rGgNqicHq...",
    "body": "Test message.",
    "attributes": {
      "ApproximateReceiveCount": "1",
      "SentTimestamp": "1545082650636",
      "SenderId": "AIDAIENQZJOLO23YVJ4VO",
      "ApproximateFirstReceiveTimestamp": "1545082650649"
    },
    "messageAttributes": {},
    "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
    "eventSource": "aws:sqs",
    "eventSourceARN": "arn:aws:sqs:us-east-2:123456789012:my-queue",
    "awsRegion": "us-east-2"
  }
]
```

Fila FIFO

```
[
  {
    "messageId": "11d6ee51-4cc7-4302-9e22-7cd8afdaadf5",
    "receiptHandle": "AQEBBX8nesZEXmkhsmZeyIE8iQAMig7qw...",
    "body": "Test message.",
    "attributes": {
      "ApproximateReceiveCount": "1",
      "SentTimestamp": "1573251510774",
      "SequenceNumber": "18849496460467696128",
      "MessageGroupId": "1",
      "SenderId": "AIDAIO23YVJENQZJOL4VO",
      "MessageDeduplicationId": "1",
      "ApproximateFirstReceiveTimestamp": "1573251510774"
    },
    "messageAttributes": {},
    "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
    "eventSource": "aws:sqs",
    "eventSourceARN": "arn:aws:sqs:us-east-2:123456789012:fifo.fifo",
    "awsRegion": "us-east-2"
  }
]
```


Escalabilidade e processamento

Para filas padrão, EventBridge usa sondagem longa para sondar uma fila até que ela se torne ativa. Quando as mensagens estão disponíveis, EventBridge lê até cinco lotes e as envia para o seu canal. Se as mensagens ainda estiverem disponíveis, EventBridge aumenta o número de processos que estão lendo lotes em até 300 instâncias a mais por minuto. O número máximo de lotes que um pipe pode processar simultaneamente é mil.

Para filas FIFO, EventBridge envia mensagens para seu canal na ordem em que ele as recebe. Ao enviar uma mensagem para uma fila do FIFO, você especifica umID do grupo de mensagens. O Amazon SQS facilita a entrega de mensagens no mesmo grupo para EventBridge, em ordem. EventBridge classifica as mensagens recebidas em grupos e envia somente um lote por vez para um grupo. Se o canal retornar um erro, ele tentará todas as novas tentativas nas mensagens afetadas antes de EventBridge receber mensagens adicionais do mesmo grupo.

Configurando uma fila para usar com Pipes EventBridge

Crie uma fila do Amazon SQS para servir como uma origem para seu pipe. Em seguida, configure a fila para permitir que seu canal processe cada lote de eventos e tente novamente em resposta EventBridge aos erros de limitação à medida que ele aumenta.

Para permitir que o pipe tenha tempo para processar cada lote de registros, defina o tempo limite de visibilidade da fila de origem para pelo menos seis vezes o runtime combinado do enriquecimento do pipe e dos componentes de destino. O tempo extra permite que você EventBridge tente novamente se o tubo for acelerado durante o processamento de um lote anterior.

Se a sua função não conseguir processar uma mensagem várias vezes, o Amazon SQS poderá enviá-la para uma fila de mensagens não entregues. Quando seu pipe retorna um erro, ele o EventBridge mantém na fila. Após o tempo limite de visibilidade, o EventBridge recebe a mensagem novamente. Para enviar mensagens para uma segunda fila após vários recebimentos, configure uma fila de mensagens mortas na fila de origem.

> **Note**
>
> Certifique-se de configurar a fila de mensagens não entregues na fila de origem, e não no pipe. A fila de mensagens não entregues configurada em uma função é usada para a fila de invocação assíncrona da função, e não para filas de origem de evento.

Se seu pipe retornar um erro ou não puder ser invocada porque está na simultaneidade máxima, o processamento poderá ter êxito com tentativas adicionais. Para que as mensagens tenham mais chances de serem processadas antes de serem enviadas para a fila de mensagens não entregues, defina a `maxReceiveCount` na política de redirecionamento da fila de origem como pelo menos 5.

Gerar relatórios de falhas de itens de lote

Quando EventBridge consome e processa dados de streaming de uma fonte, por padrão, ele aponta para o número de sequência mais alto de um lote, mas somente quando o lote é totalmente bem-sucedido. Para evitar o reprocessamento de todas as mensagens processadas com êxito em um lote com falha, é possível configurar o enriquecimento ou o destino para retornar um objeto indicando quais mensagens tiveram êxito e quais falharam. Isso se chama resposta parcial em lote.

Para ter mais informações, consulte ???.

Condições de sucesso e falha

Se você retornar qualquer um dos itens a seguir, EventBridge tratará um lote como um sucesso total:

- Uma lista de `batchItemFailure` vazia
- Uma lista de `batchItemFailure` nula
- Uma `EventResponse` vazia
- Uma `EventResponse` nula

Se você retornar qualquer um dos itens a seguir, EventBridge tratará um lote como uma falha completa:

- Uma string `itemIdentifier` vazia
- Uma `itemIdentifier` nula
- Um `itemIdentifier` com um nome de chave inválido

EventBridge repete as falhas com base em sua estratégia de repetição.

Filtragem Amazon EventBridge Pipes

Com o EventBridge Pipes, você pode filtrar os eventos de uma determinada fonte e processar apenas um subconjunto deles. Essa filtragem funciona da mesma forma que a filtragem em um barramento de EventBridge eventos ou mapeamento de origem de eventos Lambda, usando padrões de eventos. Para obter mais informações sobre padrões de eventos, consulte ???.

Um objeto `FilterCriteria` de critérios de filtro é uma estrutura que consiste em uma lista de filtros (`Filters`). Cada filtro é uma estrutura que define um padrão de filtragem (`Pattern`). Um `Pattern` é uma representação em string de uma regra de filtro JSON. Um objeto de `FilterCriteria` é como este exemplo:

```
{
  "Filters": [
    {"Pattern": "{ \"Metadata1\": [ rule1 ], \"data\": { \"Data1\": [ rule2 ] }}"
    }
  ]
}
```

Explicando melhor, aqui está o valor de `Pattern` do filtro expandido em JSON simples:

```
{
  "Metadata1": [ pattern1 ],
  "data": {"Data1": [ pattern2 ]}
}
```

As partes principais para um objeto `FilterCriteria` são as propriedades de metadados e propriedades de dados.

- Metadata properties (Propriedades dos metadados) são os campos do objeto do evento. No exemplo, `FilterCriteria.Metadata1` se refere a uma propriedade de metadados.
- Data properties (Propriedades de dados) são os campos do corpo do evento. No exemplo, `FilterCriteria.Data1` se refere a uma propriedade de dados.

Por exemplo, suponha que seu fluxo do Kinesis contenha um evento como este:

```
{
  "kinesisSchemaVersion": "1.0",
  "partitionKey": "1",
  "sequenceNumber": "49590338271490256608559692538361571095921575989136588898",
  "data": {"City": "Seattle",
    "State": "WA",
    "Temperature": "46",
    "Month": "December"
  },
  "approximateArrivalTimestamp": 1545084650.987
}
```

Quando o evento fluir pelo seu pipe, ele terá a seguinte aparência com o campo `data` codificado em base64:

```
{
  "kinesisSchemaVersion": "1.0",
  "partitionKey": "1",
  "sequenceNumber": "49590338271490256608559692538361571095921575989136588898",
  "data": "SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
  "approximateArrivalTimestamp": 1545084650.987,
  "eventSource": "aws:kinesis",
  "eventVersion": "1.0",
  "eventID":
 "shardId-000000000006:49590338271490256608559692538361571095921575989136588898",
```

```
  "eventName": "aws:kinesis:record",
  "invokeIdentityArn": "arn:aws:iam::123456789012:role/lambda-role",
  "awsRegion": "us-east-2",
  "eventSourceARN": "arn:aws:kinesis:us-east-2:123456789012:stream/lambda-stream"
}
```

As propriedades de metadados do evento do Kinesis são os campos dentro do objeto `data`, como `partitionKey` ou `sequenceNumber`.

As propriedades de dados do evento do Kinesis são os campos dentro do objeto `data`, como `City` ou `Temperature`.

Ao filtrar para corresponder a esse evento, é possível usar filtros nos campos decodificados. Por exemplo, para filtrar `partitionKey` e `City`, o seguinte filtro deveria ser usado:

```
{
  "partitionKey": [
    "1"
  ],
  "data": {
    "City": [
      "Seattle"
    ]
  }
}
```

Quando você está criando filtros de eventos, o EventBridge Pipes pode acessar o conteúdo do evento. Este conteúdo é um escape por JSON, como o campo `body` do Amazon SQS, ou codificado em base64, como o campo `data` do Kinesis. Se seus dados forem JSON válidos, seus modelos de entrada ou caminhos JSON para os parâmetros de destino poderão referenciar o conteúdo diretamente. Por exemplo, se uma origem de eventos do Kinesis for um JSON válido, será possível fazer referência a uma variável usando `<$.data.someKey>`.

Ao criar padrões de eventos, é possível filtrar com base nos campos enviados pela API de origem e não nos campos adicionados pela operação de pesquisa. Os seguintes campos não podem ser usados em padrões de eventos:

- `awsRegion`
- `eventSource`
- `eventSourceARN`

- `eventVersion`
- `eventID`
- `eventName`
- `invokeIdentityArn`
- `eventSourceKey`

Mensagem e campos de dados

Cada fonte do EventBridge Pipe contém um campo que contém a mensagem ou os dados principais. Eles são referidos como campos de mensagem ou campos de dados. Estes campos são especiais porque podem ter escape em JSON ou serem codificados em base64, mas quando são JSON válidos, podem ser filtrados com padrões JSON como se o corpo não tivesse passado por um escape. O conteúdo desses campos também pode ser usado perfeitamente em transformadores de entrada.

Filtragem correta das mensagens do Amazon SQS

Se uma mensagem do Amazon SQS não atender aos seus critérios de filtro, ela EventBridge será removida automaticamente da fila. Não é necessário excluir manualmente essas mensagens no Amazon SQS.

No Amazon SQS, o `body` da mensagem pode ser qualquer string. Porém, isso pode ser problemático se os `FilterCriteria` esperarem que o `body` esteja em um formato JSON válido. O cenário oposto também é verdadeiro: se o `body` da mensagem recebida estiver em um formato JSON válido, mas os critérios de filtro esperarem que o `body` seja uma string de texto simples, isso poderá levar a um comportamento não pretendido.

Para evitar este problema, certifique-se de que o formato do `body` nos `FilterCriteria` corresponde ao formato esperado de `body` nas mensagens que recebidas da fila. Antes de filtrar suas mensagens, avalia EventBridge automaticamente o formato da mensagem recebida `body` e do seu padrão de filtro para. `body` Se houver uma incompatibilidade, EventBridge descarta a mensagem. A tabela a seguir resume essa avaliação:

Formato do **body** da mensagem recebida	Formato do **body** do padrão de filtro	Ação resultante
String simples	String simples	EventBridge filtros com base em seus critérios de filtro.
String simples	Nenhum padrão de filtro para propriedades de dados	EventBridge filtros (somente nas outras propriedades de metadados) com base em seus critérios de filtro.
String simples	JSON válido	EventBridge deixa cair a mensagem.
JSON válido	String simples	EventBridge deixa cair a mensagem.
JSON válido	Nenhum padrão de filtro para propriedades de dados	EventBridge filtros (somente nas outras propriedades de metadados) com base em seus critérios de filtro.
JSON válido	JSON válido	EventBridge filtros com base em seus critérios de filtro.

Se você não incluir `body` como parte do seu`FilterCriteria`, EventBridge ignora essa verificação.

Filtrar corretamente as mensagens do Kinesis e do DynamoDB

Depois de filtrar os processos de um registro do Kinesis ou do DynamoDB, o iterador de fluxos avança além desse registro. Se o registro não atender aos critérios de filtro, não será necessário excluir manualmente o registro da origem de eventos. Após o período de retenção, o Kinesis e o DynamoDB excluem esses registros antigos automaticamente. Se quiser que os registros sejam excluídos antes, consulte Alterar o período de retenção de dados.

Para filtrar corretamente eventos de fontes de eventos de fluxo, tanto o campo de dados como os critérios de filtro para o campo de dados devem estar no formato JSON válido. (Para o Kinesis, o campo de dados é `data`. Para o DynamoDB, o campo de dados é `dynamodb`.) Se um dos campos

não estiver em um formato JSON válido, EventBridge descarta a mensagem ou gera uma exceção. A tabela a seguir resume o comportamento específico:

Formato dos dados recebidos (**`data`** ou **`dynamodb`**)	Formato de filtro padrão para propriedades de dados	Ação resultante
JSON válido	JSON válido	EventBridge filtros com base em seus critérios de filtro.
JSON válido	Nenhum padrão de filtro para propriedades de dados	EventBridge filtros (somente nas outras propriedades de metadados) com base em seus critérios de filtro.
JSON válido	Não JSON	EventBridge lança uma exceção no momento do pipe ou da atualização. O padrão de filtro para propriedades de dados deve estar em um formato JSON válido.
Não JSON	JSON válido	EventBridge deixa cair o recorde.
Não JSON	Nenhum padrão de filtro para propriedades de dados	EventBridge filtros (somente nas outras propriedades de metadados) com base em seus critérios de filtro.
Não JSON	Não JSON	EventBridge lança uma exceção no momento da criação ou atualização do pipe. O padrão de filtro para propriedades de dados deve estar em um formato JSON válido.

Filtrar adequadamente mensagens do Amazon Managed Streaming for Apache Kafka, Apache Kafka autogerenciado e Amazon MQ

Para origens do Amazon MQ, o campo de mensagem é `data`. Em origens do Apache Kafka (Amazon MSK e Apache Kafka autogerenciado), há dois campos de mensagem: `key` e `value`.

EventBridge elimina mensagens que não correspondem a todos os campos incluídos no filtro. Para o Apache Kafka, EventBridge confirma compensações para mensagens correspondentes e não correspondidas após invocar a função com sucesso. Para o Amazon MQ, EventBridge confirma as mensagens correspondentes após invocar a função com sucesso e reconhece as mensagens não correspondentes ao filtrá-las.

As mensagens do Apache Kafka e do Amazon MQ devem ser strings codificadas em UTF-8, sejam em texto simples ou no formato JSON. Isso porque EventBridge decodifica matrizes de bytes do Apache Kafka e do Amazon MQ em UTF-8 antes de aplicar os critérios de filtro. Se suas mensagens usarem outra codificação, como UTF-16 ou ASCII, ou se o formato da mensagem não corresponder ao formato, EventBridge processará somente filtros de `FilterCriteria` metadados. A tabela a seguir resume o comportamento específico:

Formato da mensagem recebida (**`data`** ou **`key`** e **`value`**)	Formato padrão de filtro para propriedades de mensagem	Ação resultante
String simples	String simples	EventBridge filtros com base em seus critérios de filtro.
String simples	Nenhum padrão de filtro para propriedades de dados	EventBridge filtros (somente nas outras propriedades de metadados) com base em seus critérios de filtro.
String simples	JSON válido	EventBridge filtros (somente nas outras propriedades de metadados) com base em seus critérios de filtro.
JSON válido	String simples	EventBridge filtros (somente nas outras propriedades de

Formato da mensagem recebida (**`data`** ou **`key`** e **`value`**)	Formato padrão de filtro para propriedades de mensagem	Ação resultante
		metadados) com base em seus critérios de filtro.
JSON válido	Nenhum padrão de filtro para propriedades de dados	EventBridge filtros (somente nas outras propriedades de metadados) com base em seus critérios de filtro.
JSON válido	JSON válido	EventBridge filtros com base em seus critérios de filtro.
String não codificada em UTF-8	JSON, string de texto simples ou nenhum padrão	EventBridge filtros (somente nas outras propriedades de metadados) com base em seus critérios de filtro.

Diferenças entre Lambda ESM e Pipes EventBridge

Ao filtrar eventos, o Lambda ESM EventBridge e o Pipes geralmente operam da mesma maneira. A principal diferença é que o campo `eventSourceKey` não está presente nas cargas úteis do ESM.

Enriquecimentos de eventos do Amazon EventBridge Pipes

Com a etapa de enriquecimento do EventBridge Pipes, você pode aprimorar os dados da origem antes de enviá-los ao destino. Por exemplo, é possível receber eventos Criados pelo tíquete que não incluam os dados completos do tíquete. Ao usar o enriquecimento, é possível fazer com que uma função do Lambda chame a API `get-ticket` para obter os detalhes completos do tíquete. Os pipes podem enviar essas informações para um destino.

É possível configurar os seguintes enriquecimentos ao configurar um pipe no EventBridge:

- Destino da API
- Amazon API Gateway
- Função Lambda

- Máquina de estado do Step Functions

> **Note**
>
> O EventBridge Pipes só é compatível com fluxos de trabalho Express como enriquecimentos.

O EventBridge invoca enriquecimentos de forma síncrona porque precisa esperar por uma resposta do enriquecimento antes de invocar o destino.

As respostas de enriquecimento são limitadas a um tamanho máximo de 6 MB.

Também é possível transformar os dados recebidos da origem antes de enviá-los para aprimoramento. Para obter mais informações, consulte ???.

Filtragem de eventos usando enriquecimento

O EventBridge Pipes aprova as respostas de enriquecimento diretamente para o destino configurado. Isto inclui respostas de matriz para destinos que são compatíveis com lotes. Para obter mais informações sobre o comportamento de lotes, consulte ???. Também é possível usar seu enriquecimento como filtro e transmitir menos eventos do que os recebidos da origem. Se não quiser invocar o destino, retorne uma resposta vazia, como `""`, `{}` ou `[]`.

> **Note**
>
> Se quiser invocar o destino com uma carga vazia, retorne uma matriz com `[{}]` JSON vazio.

Como invocar enriquecimentos

O EventBridge invoca enriquecimentos de forma síncrona (tipo de invocação definido como `REQUEST_RESPONSE`) porque precisa esperar por uma resposta do enriquecimento antes de invocar o destino.

> **Note**
>
> Em máquinas de estado Step Functions, o EventBridge só é compatível com fluxos de trabalho expressos como enriquecimentos, pois eles podem ser invocados de forma síncrona.

Alvos da Amazon EventBridge Pipes

É possível enviar dados em seu pipe para um destino específico. Você pode configurar os seguintes alvos ao configurar um tubo em EventBridge:

- Destino da API
- API Gateway
- Fila de trabalhos em lote
- CloudWatch grupo de registros
- Tarefa do ECS
- Barramento de eventos na mesma conta e região
- Fluxo de entrega do Firehose
- Modelo de avaliação do Inspector
- Fluxo do Kinesis
- Função do Lambda (SYNC ou ASYNC)
- Consultas de API de dados do cluster do Redshift
- SageMaker Pipeline
- Tópico do Amazon SNS (tópicos FIFO do SNS incompatíveis)
- Fila do Amazon SQS
- Máquina de estado do Step Functions
  - Fluxos de trabalho expressos (SYNC ou ASYNC)
  - Fluxos de trabalho padrão (ASYNC)
- Timestream para LiveAnalytics mesa

Parâmetros de destino

Alguns serviços de destino não enviam a carga do evento para o destino. Em vez disso, eles tratam o evento como um gatilho para invocar uma API específica. EventBridge usa o `PipeTargetParameters`para especificar quais informações são enviadas para essa API. Incluindo o seguinte:

- Destinos da API (os dados enviados para o destino da API devem corresponder à estrutura da API. É preciso usar o objeto `InputTemplate` para garantir que os dados sejam estruturados corretamente. Se quiser incluir a carga original do evento, faça referência a ela no `InputTemplate`.)
- API Gateway (os dados enviados para o API Gateway devem corresponder à estrutura da API. É preciso usar o objeto `InputTemplate` para garantir que os dados sejam estruturados corretamente. Se quiser incluir a carga original do evento, faça referência a ela no `InputTemplate`.)
- `PipeTargetRedshiftDataParameters` (Clusters da API de dados do Amazon Redshift)
- `PipeTargetSageMakerPipelineParameters`(Amazon SageMaker Runtime Model Building Pipelines)
- `PipeTargetBatchJobParameters` (AWS Batch)

> **Note**
>
> EventBridge não suporta toda a sintaxe do JSON Path e a avalia em tempo de execução. A sintaxe compatível inclui:
>
> - notação de pontos (por exemplo, `$.detail`)
> - traços
> - sublinhados
> - caracteres alfanuméricos
> - índices de matriz
> - curingas (*)

Parâmetros dinâmicos do caminho

EventBridge Os parâmetros de destino do Pipes oferecem suporte à sintaxe de caminho JSON dinâmico opcional. É possível usar esta sintaxe para especificar caminhos JSON em vez de valores estáticos (por exemplo `$.detail.state`). O valor inteiro precisa ser um caminho JSON, não apenas parte dele. Por exemplo, `RedshiftParameters.Sql` pode ser `$.detail.state`, mas não pode ser `"SELECT * FROM $.detail.state"`. Estes caminhos são substituídos dinamicamente em runtime por dados da própria carga do evento no caminho especificado. Os parâmetros do caminho dinâmico não podem referenciar valores novos ou transformados resultantes da transformação de entrada. A sintaxe compatível com caminhos JSON de parâmetros dinâmicos é a mesma da transformação da entrada. Para ter mais informações, consulte ???.

A sintaxe dinâmica pode ser usada em todos os campos de string, não enumerados, de todos os parâmetros de enriquecimento e de destino do EventBridge Pipes, exceto:

- `PipeTargetCloudWatchLogsParameters.LogStreamName`
- `PipeTargetEventBridgeEventBusParameters.EndpointId`
- `PipeEnrichmentHttpParameters.HeaderParameters`
- `PipeTargetHttpParameters.HeaderParameters`

Por exemplo, para definir o destino `PartitionKey` de um pipe do Kinesis como uma chave personalizada do seu evento de origem, defina o. KinesisTargetParameter PartitionKeypara:

- `"$.data.`*`someKey`*`"` para uma origem do Kinesis
- `"$.body.`*`someKey`*`"` para uma origem do Amazon SQS

Então, se a carga do evento for uma string JSON válida, como`{"`*`someKey`*`":"`*`someValue`*`"}`, EventBridge extrai o valor do caminho JSON e o usa como parâmetro de destino. Neste exemplo, EventBridge definiria o Kinesis como "`PartitionKey`*`someValue`*”.

Permissões

Para fazer chamadas de API nos recursos que você possui, o EventBridge Pipes precisa da permissão apropriada. EventBridge O PIPes usa a função do IAM que você especifica no canal para enriquecimento e chamadas de destino usando o IAM principal`pipes.amazonaws.com`.

Como invocar os destinos

EventBridge tem as seguintes formas de invocar um alvo:

- De forma síncrona (tipo de invocação definido como`REQUEST_RESPONSE`) — EventBridge espera por uma resposta do alvo antes de continuar.
- De forma assíncrona (tipo de invocação definido como`FIRE_AND_FORGET`) — EventBridge não espera por uma resposta antes de continuar.

Por padrão, para canais com fontes ordenadas, EventBridge invoca destinos de forma síncrona porque é necessária uma resposta do destino antes de prosseguir para o próximo evento.

Se uma fonte não impuser um pedido, como uma fila padrão do Amazon SQS, pode invocar um destino compatível de forma EventBridge síncrona ou assíncrona.

Com as funções do Lambda e as máquinas de estado Step Functions, é possível configurar o tipo de invocação.

> **Note**
>
> Para máquinas de estado Step Functions, os Fluxos de trabalho padrão devem ser invocados de forma assíncrona.

EventBridge Especificações do alvo dos tubos

AWS Batch filas de trabalho

Todos os AWS Batch `submitJob` parâmetros são configurados explicitamente com `eBatchParameters`, como acontece com todos os parâmetros do Pipe, eles podem ser dinâmicos usando um caminho JSON para a carga útil do evento de entrada.

CloudWatch Grupo de registros

Quer use um transformador de entrada ou não, a carga útil do evento é usada como mensagem de log. É possível definir o `Timestamp` (ou o explícito `LogStreamName` de seu destino) por meio de `CloudWatchLogsParameters` em `PipeTarget`. Como acontece com todos os parâmetros de pipe, eles podem ser dinâmicos usando um caminho JSON para a carga útil do evento de entrada.

Tarefa do Amazon ECS

Todos os parâmetros `runTask` do Amazon ECS são configurados explicitamente por meio de `EcsParameters`. Como acontece com todos os parâmetros de pipe, eles podem ser dinâmicos usando um caminho JSON para a carga útil do evento de entrada.

Funções do Lambda e fluxos de trabalho do Step Functions

Lambda e Step Functions não têm uma API em lote. Para processar lotes de eventos de uma origem de pipe, o lote é convertido em uma matriz JSON e passado como entrada para o destino Lambda ou Step Functions. Para ter mais informações, consulte ???.

Timestream para LiveAnalytics mesa

As considerações ao especificar uma LiveAnalytics tabela Timestream for como destino de tubulação incluem:

- Atualmente, os streams do Apache Kafka (inclusive de fornecedores terceirizados Amazon MSK ou de terceiros) não são suportados como fonte de canais.
- Se você especificou um DynamoDB fluxo Kinesis ou como a fonte do canal, deverá especificar o número de tentativas de repetição.

  Para ter mais informações, consulte ???.

Lotes e simultaneidade do Amazon EventBridge Pipes

Comportamento de lotes

EventBridge O Pipes suporta o processamento em lotes desde a origem até os alvos que o suportam. Além disso, o processamento de lotes para enriquecimento é suportado para e. AWS Lambda AWS Step Functions Como serviços diferentes oferecem suporte a diferentes níveis de processamento de lotes, não é possível configurar um pipe com um tamanho de lote maior do que o suporte de destino. Por exemplo, as origens de fluxo do Amazon Kinesis são compatíveis com um tamanho máximo de lote de 10.000 registros, mas o Amazon Simple Queue Service é compatível com um máximo de dez mensagens por lote como destino. Portanto, um pipe de um fluxo do Kinesis para uma fila do Amazon SQS pode ter um tamanho máximo de lote configurado na origem de 10.

Se configurar um pipe com um enriquecimento ou destino que seja compatível com o processamento de lotes, não será possível ativar o processamento de lotes na origem.

Quando o processamento de lotes é ativado na origem, matrizes de registros JSON são passadas pelo pipe e mapeadas para a API de lote de um enriquecimento ou destino compatível. Os transformadores de entrada são aplicados separadamente em cada registro JSON individual na matriz, não na matriz como um todo. Para obter exemplos dessas matrizes, consulte ??? e selecione uma origem específica. O Pipes usará a API em lote para o enriquecimento ou destino suportado, mesmo que o tamanho do lote seja 1. Se o enriquecimento ou o destino não tiver uma API em lote, mas receber cargas JSON completas, como Lambda e Step Functions, toda a matriz JSON será enviada em uma única solicitação. A solicitação será enviada como uma matriz JSON, mesmo que o tamanho do lote seja 1.

Se um pipe estiver configurado para o processamento de lotes na origem e o destino for compatível com o processamento de lotes, será possível retornar uma matriz de itens JSON do seu enriquecimento. Esta matriz pode incluir uma matriz menor ou maior do que a origem original. No entanto, se a matriz for maior que o tamanho do lote suportado pelo destino, o pipe não invocará o destino.

Destinos de processamento de lotes compatíveis

Destino	Tamanho máximo do lote
CloudWatch Registros	10.000
EventBridge ônibus de eventos	10
Stream Firehose	500
Fluxo do Kinesis	500
Função do Lambda	definido pelo cliente
Máquina de estado do Step Functions	definido pelo cliente
Tópico do Amazon SNS	10
Fila do Amazon SQS	10

Os seguintes enriquecimentos e destinos recebem a carga útil completa do evento do lote para processamento e são limitados pelo tamanho total da carga útil do evento, em vez do tamanho do lote:

- Máquina de estado do Step Functions (262.144 caracteres)
- Função do Lambda (6 MB)

Lote com falha parcial

Para o Amazon SQS e fontes de streaming, como Kinesis e DynamoDB, o Pipes oferece suporte ao tratamento parcial de falhas em lote das EventBridge falhas de destino. Se o destino suportar o agrupamento em lotes e somente parte do lote for bem-sucedido, EventBridge tentará automaticamente agrupar novamente o restante da carga útil. Para o conteúdo mais up-to-date enriquecido, essa nova tentativa ocorre em todo o canal, incluindo a reinvocação de qualquer enriquecimento configurado.

O tratamento de lotes com falha parcial do enriquecimento não é compatível.

Para destinos do Lambda e Step Functions, também é possível especificar uma falha parcial retornando uma carga com estrutura definida do destino. Isto indica eventos que precisam ser repetidos.

Exemplo de estrutura de carga útil com falha parcial

```
{
  "batchItemFailures": [
    {
      "itemIdentifier": "id2"
    },
    {
      "itemIdentifier": "id4"
    }
]
```

No exemplo, `itemIdentifier` corresponde ao ID dos eventos manipulados pelo seu destino da origem original. No Amazon SQS, esse é o `messageId`. No Kinesis e no DynamoDB, este é o `eventID`. Para que EventBridge os Pipes lidem adequadamente com falhas parciais de lote dos alvos, esses campos precisam ser incluídos em qualquer carga útil da matriz retornada pelo enriquecimento.

Comportamento de throughput e simultaneidade

Cada evento ou lote de eventos recebidos por um pipe que viaja até um enriquecimento ou destino é considerado como uma execução de pipe. Um pipe no estado `STARTED` pesquisa continuamente

os eventos da origem, aumentando e diminuindo a escala dependendo da lista de pendências disponível e das configurações de processamento de lotes definidas.

Para cotas de execuções simultâneas de pipes e o número de pipes por conta e região, consulte ???.

Por padrão, um único pipe escalará para o seguinte máximo de execuções simultâneas, dependendo da origem:

- DynamoDB: as execuções simultâneas podem subir até as `ParallelizationFactor` configuradas no pipe e multiplicadas pelo número de fragmentos no fluxo.
- Apache Kafka: as execuções simultâneas podem aumentar o número de partições no tópico, até mil.
- Kinesis: as execuções simultâneas podem subir tão alto quanto as `ParallelizationFactor` configuradas no pipe multiplicadas pelo número de fragmentos no fluxo.
- Amazon MQ: 5
- Amazon SQS: 1250

Se tiver requisitos para throughputs sondagem ou limites de simultaneidade máximos mais altos, entre em contato com o suporte.

> **Note**
>
> Os limites de execução são considerados limitações de segurança de melhor esforço. Embora a pesquisa não limitada abaixo desses valores, um pipe ou conta pode estourar mais do que esses valores recomendados.

As execuções de pipes são limitadas a um máximo de cinco minutos, incluindo o enriquecimento e o processamento de destino. Este limite não pode ser aumentado.

Os pipes com origens estritamente ordenadas, como filas FIFO do Amazon SQS, Kinesis e DynamoDB Streams ou tópicos do Apache Kafka) são ainda mais limitados em termos de simultaneidade pela configuração da origem, como o número de IDs de grupos de mensagens para filas FIFO ou o número de fragmentos para filas Kinesis. Como o pedido é estritamente garantido dentro dessas restrições, um pipe com uma origem ordenada não pode exceder esses limites de simultaneidade.

Transformação de entrada do Amazon EventBridge Pipes

O Amazon EventBridge Pipes é compatível com transformadores de entrada opcionais ao transmitir dados para o enriquecimento e o destino. É possível usar transformadores de entrada para remodelar a carga útil de entrada de eventos JSON para atender às necessidades do serviço de enriquecimento ou de destino. Para o Amazon API Gateway e destinos de API, é assim que o evento de entrada é moldado para o modelo RESTful da sua API. Os transformadores de entrada são modelados como um parâmetro `InputTemplate`. Eles podem ser texto livre, um caminho JSON para a carga do evento ou um objeto JSON que inclui caminhos JSON embutidos para a carga do evento. Para enriquecimento, a carga útil do evento vem da origem. Para destinos, a carga útil do evento é o que é retornado do enriquecimento, se estiver configurado no pipe. Além dos dados específicos do serviço na carga útil do evento, é possível usar variáveis reservadas `InputTemplate` para referenciar os dados do seu pipe.

Para acessar itens em uma matriz, use a notação de colchetes.

> **Note**
>
> O EventBridge não suporta toda a sintaxe do JSON Path e a avalia em runtime. A sintaxe compatível inclui:
>
> - notação de pontos (por exemplo, `$.detail`)
> - traços
> - sublinhados
> - caracteres alfanuméricos
> - índices de matriz
> - curingas (*)

Os seguintes são parâmetros `InputTemplate` de amostra que fazem referência a uma carga útil de eventos do Amazon SQS:

String estática

```
InputTemplate: "Hello, sender"
```

Caminho JSON

```
InputTemplate: <$.attributes.SenderId>
```

String dinâmica

```
InputTemplate: "Hello, <$.attributes.SenderId>"
```

JSON estático

```
InputTemplate: >
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3",
}
```

JSON dinâmico

```
InputTemplate: >
{
  "key1": "value1"
  "key2": <$.body.key>,
  "d": <aws.pipes.event.ingestion-time>
}
```

Use a notação de colchetes para usar itens em uma matriz:

```
InputTemplate: >
{
  "key1": "value1"
  "key2": <$.body.Records[3]>,
  "d": <aws.pipes.event.ingestion-time>
}
```

> **Note**
>
> O EventBridge substitui os transformadores de entrada em runtime para garantir uma saída JSON válida. Por isso, coloque aspas nas variáveis que se referem aos parâmetros de caminho JSON, mas não coloque aspas nas variáveis que se referem a objetos ou matrizes JSON.

Variáveis reservadas

Os modelos de entrada podem usar as seguintes variáveis reservadas:

- `<aws.pipes.pipe-arn>`: o nome do recurso da Amazon (ARN) do pipe.
- `<aws.pipes.pipe-name>`: o nome do pipe.
- `<aws.pipes.source-arn>`: o ARN de origem de evento do pipe.
- `<aws.pipes.enrichment-arn>`: o ARN de enriquecimento do pipe.
- `<aws.pipes.target-arn>`: o ARN de destino do pipe.
- `<aws.pipes.event.ingestion-time>`: o momento em que o evento foi recebido pelo transformador de entrada. Este é um carimbo de data/hora ISO 8601. Este tempo é diferente para o transformador de entrada de enriquecimento e o transformador de entrada de destino, dependendo de quando o enriquecimento concluiu o processamento do evento.
- `<aws.pipes.event>`: o evento conforme recebido pelo transformador de entrada.

  Para um transformador de entrada de enriquecimento, esse é o evento da origem. Ele contém a carga original da origem, além de metadados específicos do serviço adicionais. Para obter exemplos específicos deste serviço, consulte ???.

  Para um transformador de entrada de destino, esse é o evento retornado pelo enriquecimento, se um estiver configurado, sem metadados adicionais. Dessa forma, uma carga útil retornada por enriquecimento pode não ser JSON. Se nenhum enriquecimento estiver configurado no pipe, esse é o evento da origem com metadados.
- `<aws.pipes.event.json>`: o mesmo que `aws.pipes.event`, mas a variável só tem um valor se a carga original, da origem ou retornada pelo enriquecimento, for JSON. Se o pipe tiver um campo codificado, como o campo `body` do Amazon SQS ou os `data` do Kinesis, esses campos serão decodificados e transformados em JSON válido. Como não passou por um escape, a variável só pode ser usada como um valor para um campo JSON. Para obter mais informações, consulte ???.

Exemplo de transformação de entrada

Veja a seguir um exemplo de evento do Amazon EC2 que podemos usar como nosso exemplo de evento.

```
{
  "version": "0",
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2015-11-11T21:29:54Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"
  ],
  "detail": {
    "instance-id": "i-0123456789",
    "state": "RUNNING"
  }
}
```

Vamos usar o seguinte JSON como nosso Transformador.

```
{
  "instance" : <$.detail.instance-id>,
  "state": <$.detail.state>,
  "pipeArn" : <aws.pipes.pipe-arn>,
  "pipeName" : <aws.pipes.pipe-name>,
  "originalEvent" : <aws.pipes.event.json>
}
```

A seguir estará a Saída resultante:

```
{
  "instance" : "i-0123456789",
  "state": "RUNNING",
  "pipeArn" : "arn:aws:pipe:us-east-1:123456789012:pipe/example",
  "pipeName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

Análise implícita de dados do corpo

Os campos a seguir na carga de entrada podem ser escapados por JSON, como o objeto `body` do Amazon SQS, ou codificados em base64, como o objeto `data` do Kinesis. Tanto para filtragem quanto para transformação de entrada, o EventBridge transforma esses campos em JSON válido para que os subvalores possam ser referenciados diretamente. Por exemplo, `<$.data.someKey>` para o Kinesis.

Para que o destino receba a carga original sem nenhum metadado adicional, use um transformador de entrada com esses dados do corpo, específicos da origem. Por exemplo, `<$.body>` para o Amazon SQS ou `<$.data>` para o Kinesis. Se a carga original for uma string JSON válida (por exemplo, `{"key": "value"}`), o uso do transformador de entrada com dados específicos do corpo da origem resultará na remoção das aspas dentro da carga da origem original. Por exemplo, `{"key": "value"}` se tornará `"{key: value}"` quando for entregue ao destino. Se seu destino exigir cargas JSON válidas (por exemplo, EventBridge Lambda ou Step Functions), isso causará falha na entrega. Para que o destino receba os dados de origem originais sem gerar JSON inválido, envolva o transformador de entrada de dados do corpo de origem em JSON. Por exemplo, `{"data": <$.data>}`.

A análise implícita do corpo também pode ser usada para preencher dinamicamente os valores da maioria dos parâmetros de enriquecimento ou destino do pipe. Para obter mais informações, consulte ???.

> **Note**
>
> Se a carga original for um JSON válido, este campo conterá o JSON sem escape e sem codificação em base64. No entanto, se a carga útil não for um JSON válido, o EventBridge codifica em base64 para os campos listados abaixo, com exceção do Amazon SQS.

- MQ ativo: `data`
- Kinesis: `data`
- Amazon MSK: `key` e `value`
- Rabbit MQ: `data`
- Apache Kafka autogerenciado: `key` e `value`
- Amazon SQS: `body`

Problemas comuns com a transformação de entrada

Estes são alguns problemas comuns ao transformar a entrada nos pipes do EventBridge:

- Para strings, as aspas são necessárias.
- Não há validação ao criar o caminho JSON para o modelo.
- Se especificar uma variável para corresponder a um caminho JSON que não existe no evento, essa variável não será criada e não aparecerá na saída.
- Propriedades JSON, como `aws.pipes.event.json`, só podem ser usadas como o valor de um campo JSON, não embutidas em outras strings.
- O EventBridge não faz o escape de valores extraídos pelo Caminho de entrada, ao preencher o Modelo de entrada para um destino.
- Se um caminho JSON fizer referência a um objeto ou matriz JSON, mas a variável for referenciada em uma string, o EventBridge removerá todas as aspas internas para garantir uma string válida. Por exemplo, "Body is <$.body>" resultaria na remoção das aspas do objeto pelo EventBridge.

  Portanto, se quiser gerar um objeto JSON com base em uma única variável de caminho JSON, deverá colocá-lo como uma chave. Neste exemplo, `{"body": <$.body>}`.
- As aspas não são necessárias para variáveis que representam cadeias de caracteres. Eles são permitidos, mas o EventBridge Pipes adiciona automaticamente aspas aos valores das variáveis de string durante a transformação, para garantir que a saída da transformação seja um JSON válido. O EventBridge Pipes não adiciona aspas às variáveis que representam objetos ou matrizes JSON. Não adicione aspas para variáveis que representem objetos ou matrizes JSON.

  Por exemplo, o seguinte modelo de entrada inclui variáveis que representam cadeias de caracteres e objetos JSON:

```
{
  "pipeArn" : <aws.pipes.pipe-arn>,
  "pipeName" : <aws.pipes.pipe-name>,
  "originalEvent" : <aws.pipes.event.json>
}
```

  Resultando em JSON válido com cotação adequada:

```
{
  "pipeArn" : "arn:aws:events:us-east-2:123456789012:pipe/example",
  "pipeName" : "example",
```

```
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

- Para enriquecimentos ou destinos do Lambda ou Step Functions, os lotes são entregues ao destino como matrizes JSON, mesmo que o tamanho do lote seja 1. No entanto, os transformadores de entrada ainda serão aplicados a registros individuais na matriz JSON, não à matriz como um todo. Para obter mais informações, consulte ???.

Log Amazon EventBridge Pipes

EventBridge O registro de tubulações permite que você faça com que a EventBridge Pipes envie registros detalhando o desempenho da tubulação para os AWS serviços suportados. Use logs para obter informações sobre o desempenho de execução do seu pipe e para ajudar na solução de problemas e na depuração.

Você pode selecionar os seguintes AWS serviços como destinos de log para os quais o EventBridge Pipes entrega registros:

- CloudWatch Registros

  EventBridge entrega registros de registro para o grupo de CloudWatch registros de registros especificado.

  Use o CloudWatch Logs para centralizar os registros de todos os seus sistemas, aplicativos e AWS serviços que você usa, em um único serviço altamente escalável. Para obter mais informações, consulte Como trabalhar com grupos e fluxos de registros no Guia do usuário do Amazon CloudWatch Logs.
- Registros de transmissão do Firehose

  EventBridge entrega registros de log para um stream de entrega do Firehose.

  O Amazon Data Firehose é um serviço totalmente gerenciado para fornecer dados de streaming em tempo real para destinos como determinados AWS serviços, bem como qualquer endpoint HTTP personalizado ou endpoints HTTP de propriedade de provedores de serviços terceirizados compatíveis. Para obter mais informações, consulte Criação de um stream de entrega do Amazon Data Firehose no Guia do usuário do Amazon Data Firehose.
- Logs do Amazon S3

EventBridge entrega registros de log como objetos do Amazon S3 para o bucket especificado.

O Amazon S3 é um serviço de armazenamento de objetos que oferece escalabilidade, disponibilidade de dados, segurança e performance líderes do setor. Para obter mais informações, consulte Fazer upload, baixar e trabalhar com objetos no Amazon S3 no Manual do usuário do Amazon Simple Storage Service.

Como funciona o registro do Amazon EventBridge Pipes

Uma execução de pipe é um evento ou lote de eventos recebidos por um pipe que viaja até um enriquecimento e/ou destino. Se ativado, EventBridge gera um registro de log para cada etapa de execução executada à medida que o lote de eventos é processado. As informações contidas no registro se aplicam ao lote de eventos, seja ele um único evento ou até 10 mil eventos.

Também é possível configurar o tamanho do lote de eventos na origem e no destino do pipe. Para ter mais informações, consulte ???.

Os dados de registro enviados para cada destino de registro são os mesmos.

Se um destino do Amazon CloudWatch Logs estiver configurado, os registros de log entregues a todos os destinos terão um limite de 256 kb. Os campos serão truncados conforme necessário.

Você pode personalizar os registros EventBridge enviados para os destinos de log selecionados da seguinte forma:

- Você pode especificar o nível de log, que determina as etapas de execução para as quais EventBridge envia registros para os destinos de log selecionados. Para ter mais informações, consulte ???.
- Você pode especificar se o EventBridge Pipes inclui dados de execução nos registros das etapas de execução quando forem relevantes. Esses dados incluem:
  - A carga útil do lote de eventos
  - A solicitação enviada ao serviço de AWS enriquecimento ou de destino
  - A resposta retornada pelo serviço de AWS enriquecimento ou de destino

  Para ter mais informações, consulte ???.

Especificando o nível EventBridge de registro de tubos

Você pode especificar os tipos de etapas de execução para as quais os registros são EventBridge enviados para os destinos de log selecionados.

Escolha entre os níveis de detalhe a seguir para incluir nos registros de log. O nível de log se aplica a todos os destinos de log especificados para o pipe. Cada nível de log inclui as etapas de execução dos níveis de log anteriores.

- DESLIGADO — EventBridge não envia nenhum registro para nenhum destino de registro especificado. Essa é a configuração padrão.
- ERROR — EventBridge envia todos os registros relacionados aos erros gerados durante a execução do pipe para os destinos de log especificados.
- INFO — EventBridge envia todos os registros relacionados a erros, bem como seleciona outras etapas executadas durante a execução do pipe para os destinos de log especificados.
- TRACE — EventBridge envia todos os registros gerados durante qualquer etapa da execução do pipe para os destinos de log especificados.

No EventBridge console, CloudWatch os registros são selecionados como um destino de registro por padrão, assim como o nível do `ERROR` registro. Então, por padrão, o EventBridge Pipes cria um novo CloudWatch grupo de registros para o qual envia registros de log contendo o `ERROR` nível de detalhe. Nenhum padrão é selecionado ao configurar os logs programaticamente.

A tabela a seguir lista as etapas de execução inclusas em cada nível do log.

Etapa	RASTREAR	INFO	ERRO	DESL.
Falha na execução	x	x	x	
Execução com falha parcial	x	x	x	
Execução iniciada	x	x		
Execução bem-sucedida	x	x		
Execução limitada	x	x	x	
Tempo limite de execução	x	x	x	

Etapa	RASTREAR	INFO	ERRO	DESL.
Invocação de enriquecimento com falha	x	x	x	
Invocação de enriquecimento ignorada	x	x		
Invocação de enriquecimento iniciada	x			
Invocação de enriquecimento com êxito	x			
Fase de enriquecimento iniciada	x	x		
Fase de enriquecimento com falha	x	x	x	
Estágio de enriquecimento com êxito	x	x		
Transformação de enriquecimento com falha	x	x	x	
Transformação de enriquecimento iniciada	x			
Transformação do enriquecimento com êxito	x			
Invocação de destino com falha	x	x	x	
Invocação de destino com falha parcial	x	x	x	
Invocação do destino ignorada	x			

Etapa	RASTREAR	INFO	ERRO	DESL.
Invocação do destino iniciada	x			
Invocação do destino com êxito	x			
Estágio de destino inserido	x	x		
Estágio de destino com falha	x	x	x	
Falha parcial do estágio de destino	x	x	x	
Estágio de destino ignorado	x			
Estágio de destino com êxito	x	x		
Falha na transformação do alvo	x	x	x	
Transformação de destino iniciada	x			
Transformação do destino com êxito	x			

Incluindo dados de execução nos registros do EventBridge Pipes

Você pode especificar EventBridge para incluir dados de execução nos registros que ele gera. Os dados de execução incluem campos que representam a carga útil do lote de eventos, bem como a solicitação enviada e a resposta do enriquecimento e do destino.

Os dados de execução são úteis para solução de problemas e depuração. O campo `payload` contém o conteúdo real de cada evento incluído no lote, permitindo que você correlacione eventos individuais a uma execução de pipe específica.

Se optar por incluir dados de execução, eles serão incluídos em todos os destinos de log especificados para o pipe.

> **Important**
>
> Esses campos podem conter informações confidenciais. EventBridge não faz nenhuma tentativa de redigir o conteúdo desses campos durante o registro.

Ao incluir dados de execução, EventBridge adicione os seguintes campos aos registros relevantes:

- **`payload`**

  Representa o conteúdo do lote de eventos que está sendo processado pelo pipe.

  EventBridge inclui o `payload` campo nos registros gerados nas etapas em que o conteúdo do lote de eventos pode ter sido atualizado. Isto inclui as seguintes etapas:

  - `EXECUTION_STARTED`
  - `ENRICHMENT_TRANSFORMATION_SUCCEEDED`
  - `ENRICHMENT_STAGE_SUCCEEDED`
  - `TARGET_TRANSFORMATION_SUCCEEDED`
  - `TARGET_STAGE_SUCCEEDED`

- **`awsRequest`**

  Representa a solicitação enviada ao enriquecimento ou ao destino como uma string JSON. Para solicitações enviadas para um destino de API, isto representa a solicitação HTTP enviada para esse endpoint.

  EventBridge inclui o `awsRequest` campo nos registros gerados nas etapas finais de enriquecimento e direcionamento; ou seja, depois EventBridge de ter executado ou tentado executar a solicitação em relação ao serviço de enriquecimento ou destino especificado. Isto inclui as seguintes etapas:

  - `ENRICHMENT_INVOCATION_FAILED`
  - `ENRICHMENT_INVOCATION_SUCCEEDED`
  - `TARGET_INVOCATION_FAILED`
  - `TARGET_INVOCATION_PARTIALLY_FAILED`
  - `TARGET_INVOCATION_SUCCEEDED`

- **`awsResponse`**

Representa a resposta retornada pelo enriquecimento ou destino, no formato JSON. Para solicitações enviadas para um destino de API, isto representa a resposta HTTP retornada desse endpoint.

Da mesma formaawsRequest, EventBridge inclui o awsResponse campo nos registros gerados nas etapas finais de enriquecimento e direcionamento; ou seja, depois de executar EventBridge ou tentar executar uma solicitação no serviço de enriquecimento ou destino especificado e receber uma resposta. Isto inclui as seguintes etapas:

- ENRICHMENT_INVOCATION_FAILED
- ENRICHMENT_INVOCATION_SUCCEEDED
- TARGET_INVOCATION_FAILED
- TARGET_INVOCATION_PARTIALLY_FAILED
- TARGET_INVOCATION_SUCCEEDED

Para uma discussão sobre as etapas de execução do pipe, consulte ???.

Truncando dados de execução nos registros de log do EventBridge Pipes

Se você optar por EventBridge incluir dados de execução nos registros de log de um pipe, existe a possibilidade de que um registro exceda o limite de tamanho de 256 KB. Para evitar isso, trunca EventBridge automaticamente os campos de dados de execução, na seguinte ordem. EventBridge trunca cada campo inteiramente antes de prosseguir para truncar o próximo campo. EventBridge trunca os dados do campo simplesmente removendo caracteres do final da sequência de dados; nenhuma tentativa é feita para truncar com base na importância dos dados, e o truncamento invalidará a formatação JSON.

- payload
- awsRequest
- awsResponse

Se EventBridge truncar campos no evento, o truncatedFields campo incluirá uma lista dos campos de dados truncados.

Relatório de erros nos registros EventBridge de log do Pipes

EventBridge também inclui dados de erro, quando disponíveis, nas etapas de execução do pipe que representam estados de falha. Essas etapas incluem:

- `ExecutionThrottled`
- `ExecutionTimeout`
- `ExecutionFailed`
- `ExecutionPartiallyFailed`
- `EnrichmentTransformationFailed`
- `EnrichmentInvocationFailed`
- `EnrichmentStageFailed`
- `TargetTransformationFailed`
- `TargetInvocationFailed`
- `TargetInvocationPartiallyFailed`
- `TargetStageFailed`
- `TargetStagePartiallyFailed`

EventBridge Etapas de execução de tubos

Compreender o fluxo das etapas de execução do pipe pode ajudá-lo a solucionar problemas ou depurar o desempenho do pipe usando registros.

Uma execução de pipe é um evento ou lote de eventos recebidos por um pipe que viaja até um enriquecimento ou destino. Se ativado, EventBridge gera um registro de log para cada etapa de execução executada à medida que o lote de eventos é processado.

Em um alto nível, a execução contém dois estágios ou coleta de etapas: enriquecimento e meta. Cada um desses estágios consiste em etapas de transformação e invocação.

As principais etapas de uma execução com êxito do pipe seguem este fluxo:

- A execução do pipe é iniciada.
- A execução entra no estágio de enriquecimento se tiver especificado um enriquecimento para os eventos. Se não especificou um enriquecimento, a execução prossegue para o estágio de destino.

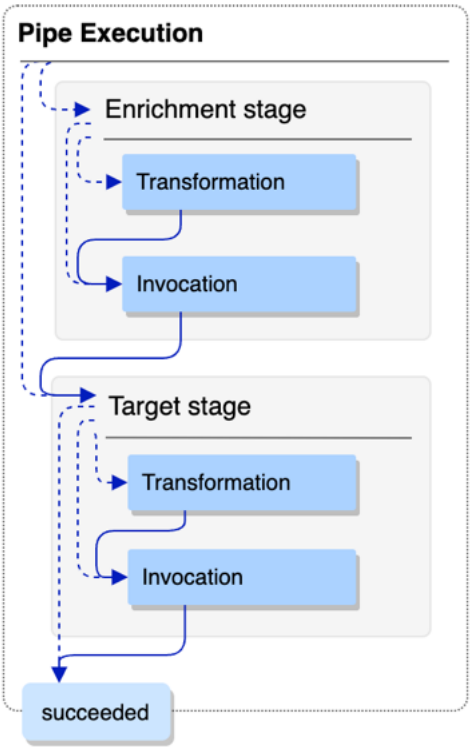
No estágio de enriquecimento, o pipe executa qualquer transformação especificada e invoca o enriquecimento.

- No estágio de destino, o pipe executa qualquer transformação especificada e invoca o destino.

  Se não especificou a transformação ou o alvo, a execução pulará o estágio de destino.

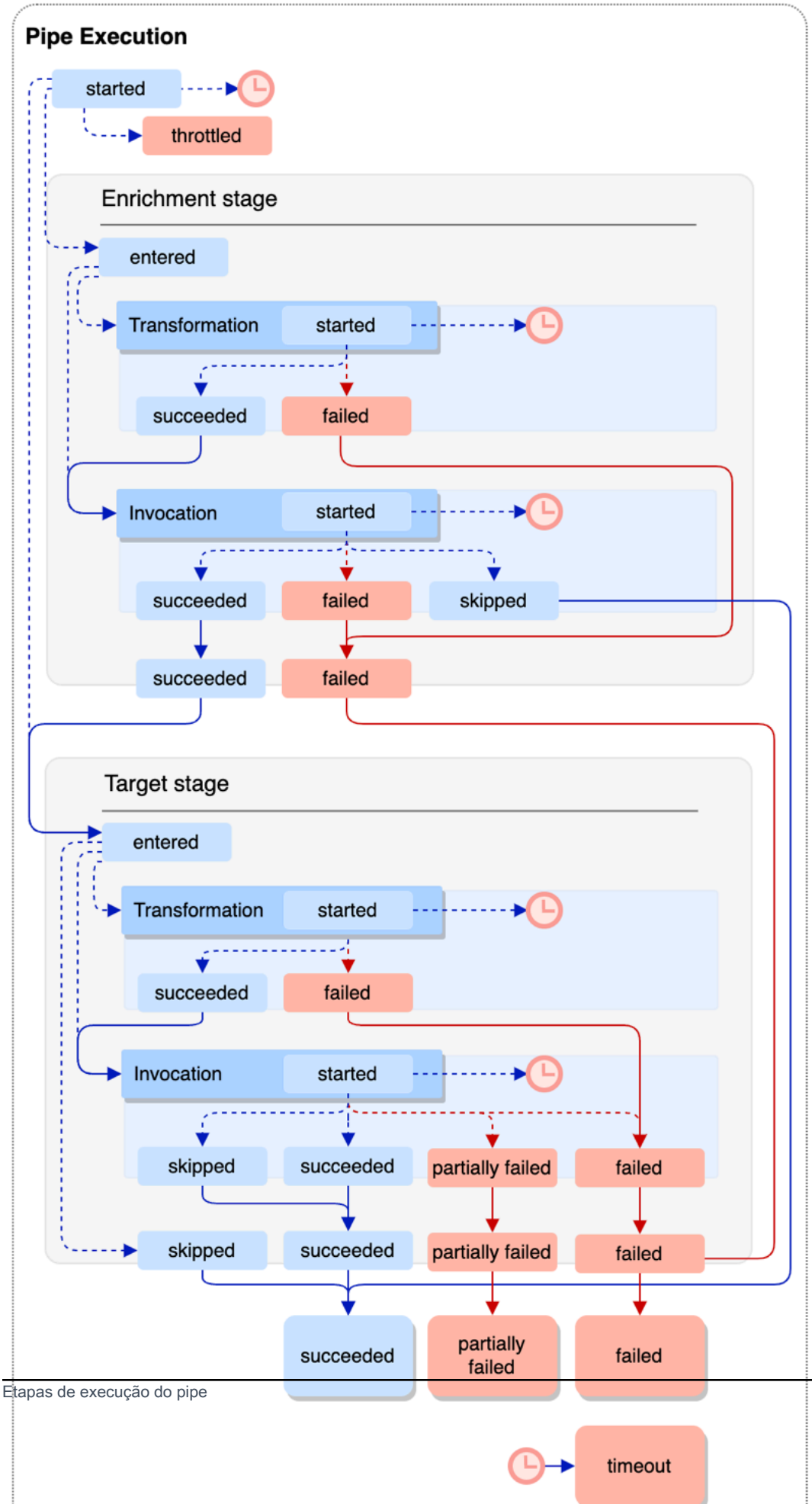
- A execução do pipe é concluída com êxito.

O diagrama abaixo demonstra esse fluxo. Os caminhos divergentes são formatados como linhas pontilhadas.



O diagrama abaixo apresenta uma visão detalhada do fluxo de execução da pipe, com todas as etapas de execução possíveis representadas. Novamente, caminhos divergentes são formatados como linhas pontilhadas

Para obter uma lista completa das etapas de execução do pipe, consulte ???.


Pipe Execution
started
throttled
Enrichment stage
entered
Transformation
started
succeeded
failed
Invocation
started
succeeded
failed
skipped
succeeded
failed
Target stage
entered
Transformation
started
succeeded
failed
Invocation
started
skipped
succeeded
partially failed
failed
skipped
succeeded
partially failed
failed
succeeded
partially failed
failed
timeout

Observe que a invocação do destino pode resultar em uma falha parcial do lote. Para ter mais informações, consulte ???.

EventBridge Referência do esquema de log de tubulações

A referência a seguir detalha o esquema dos registros de log do EventBridge Pipes.

Cada registro de log representa uma etapa de execução do pipe e pode conter até 10 mil eventos se a origem e o destino do pipe tiverem sido configurados para o processamento de lotes.

Para ter mais informações, consulte ???.

```
{
    "executionId": "guid",
    "timestamp": "date_time",
    "messageType": "execution_step",
    "resourceArn": "arn:aws:pipes:region:account:pipe/pipe-name",
    "logLevel": "TRACE | INFO | ERROR",
    "payload": "{}",
    "awsRequest": "{}"
    "awsResponse":"{}"
    "truncatedFields": ["awsRequest","awsResponse","payload"],
    "error": {
        "httpStatusCode": code,
        "message": "error_message",
        "details": "",
        "awsService": "service_name",
        "requestId": "service_request_id"
    }
}
```

executionId

O ID da execução do pipe.

Uma execução de pipe é um evento ou lote de eventos recebidos por um pipe que viaja até um enriquecimento ou destino. Para ter mais informações, consulte ???.

timestamp

A data e a hora em que o evento de log foi emitido.

Unidade: milissegundo

messageType

A etapa de execução do pipe para a qual o registro foi gerado.

Para obter mais informações sobre os passos de execução de pipes, consulte ???.

resourceArn

O nome do recurso da Amazon (ARN) para o pipe.

logLevel

O nível de detalhe especificado para o log do pipe.

Valores válidos: `ERROR` | `INFO` | `TRACE`

Para ter mais informações, consulte ???.

payload

Os conteúdos do lote de eventos que está sendo processado pelo pipe.

EventBridge incluirá esse campo somente se você tiver especificado incluir dados de execução nos registros desse canal. Para mais informações, consulte ???.

> **Important**
>
> Esses campos podem conter informações confidenciais. EventBridge não faz nenhuma tentativa de redigir o conteúdo desses campos durante o registro.

Para ter mais informações, consulte ???.

awsRequest

A solicitação enviada ao serviço de enriquecimento ou de destino, no formato JSON. Para solicitações enviadas para um destino de API, isto representa a solicitação HTTP enviada para esse endpoint.

EventBridge incluirá esse campo somente se você tiver especificado incluir dados de execução nos registros desse canal. Para mais informações, consulte ???.

> **Important**
>
> Esses campos podem conter informações confidenciais. EventBridge não faz nenhuma tentativa de redigir o conteúdo desses campos durante o registro.

Para ter mais informações, consulte ???.

awsResponse

A resposta retornada pelo enriquecimento ou destino, no formato JSON. Para solicitações enviadas para um destino de API, isto representa a resposta HTTP retornada desse endpoint, e não a resposta retornada pelo próprio serviço de destino da API.

EventBridge incluirá esse campo somente se você tiver especificado incluir dados de execução nos registros desse canal. Para mais informações, consulte ???.

> **Important**
>
> Esses campos podem conter informações confidenciais. EventBridge não faz nenhuma tentativa de redigir o conteúdo desses campos durante o registro.

Para ter mais informações, consulte ???.

truncatedFields

Uma lista de todos os campos de dados de execução EventBridge foi truncada para manter o registro abaixo da limitação de tamanho de 256 KB.

Se EventBridge não foi necessário truncar nenhum dos campos de dados de execução, esse campo está presente, mas. `null`

Para ter mais informações, consulte ???.

erro

Contém informações sobre qualquer erro gerado durante esta etapa de execução do pipe.

Se nenhum erro foi gerado durante essa etapa de execução do pipe, esse campo está presente, mas `null`.

httpStatusCode

O código de status HTTP retornado pelo serviço chamado.

mensagem

A mensagem de erro retornada pelo serviço chamado.

detalhes

Qualquer informação de erro detalhada e retornada pelo serviço chamado.

awsService

O nome do serviço chamado.

requestId

O ID de solicitação para essa solicitação do serviço chamado.

Registro e monitoramento de Amazon EventBridge Pipes usando AWS CloudTrail Amazon CloudWatch Logs

Você pode registrar invocações de EventBridge Pipes e usar CloudTrail e monitorar a integridade de seus pipes usando CloudWatch métricas.

CloudWatch métricas

EventBridge O Pipes envia métricas para a Amazon a CloudWatch cada minuto para tudo, desde as execuções de um pipe sendo limitadas até um alvo sendo invocado com sucesso.

Métrica	Descrição	Dimensões	Unidades
`Concurrency`	O número de execuções simultâneas de um pipe.	AwsAccountId	Nenhum
`Duration`	Período de tempo que levou a execução do pipe.	PipeName	Milissegundos
`EventCount`	O número de eventos que um pipe processou.	PipeName	Nenhum

Métrica	Descrição	Dimensões	Unidades
`EventSize`	O tamanho da carga útil do evento que invocou o pipe.	PipeName	Bytes
`ExecutionThrottled`	Quantas execuções de um pipe foram limitadas. **Note** Esse valor será `0` se nenhuma execução for limitada.	AwsAccountId, PipeName	Nenhum
`ExecutionTimeout`	Quantas execuções de um pipe atingiram o tempo limite antes de concluir a execução. **Note** Este valor será `0` se nenhuma execução atingiu o tempo limite.	PipeName	Nenhum
`ExecutionFailed`	Quantas execuções de um pipe falharam. **Note** Este valor será `0` se nenhuma execução falhou.	PipeName	Nenhum
`ExecutionPartiallyFailed`	Quantas execuções de um pipe falharam parcialmente. **Note** Este valor será `0` se nenhuma execução falhou parcialmente.	PipeName	Nenhum

Métrica	Descrição	Dimensões	Unidades
`EnrichmentStageDuration`	Quanto tempo o estágio de enriquecimento levou para ser concluído.	PipeName	Milissegundos
`EnrichmentStageFailed`	Quantas execuções do estágio de enriquecimento de um pipe falharam. **Note** Este valor será `0` se nenhuma execução falhou.	PipeName	Nenhum
`Invocations`	Número total de invocações.	AwsAccountId, PipeName	Nenhum
`TargetStageDuration`	Quanto tempo o estágio de destino levou para ser concluído.	PipeName	Milissegundos
`TargetStageFailed`	Quantas execuções do estágio de alvo de um pipe falharam. **Note** Este valor será `0` se nenhuma execução falhou.	PipeName	Nenhum

Métrica	Descrição	Dimensões	Unidades
`TargetStagePartiallyFailed`	Quantas execuções do estágio de alvo de um pipe falharam parcialmente. **Note** Este valor será `0` se nenhuma execução do estágio de destino falhar parcialmente.	PipeName	Nenhum
`TargetStageSkipped`	Quantas execuções do estágio de destino de um pipe foram ignoradas (por exemplo, devido ao enriquecimento retornar uma carga útil vazia).	PipeName	Contagem

Dimensões para CloudWatch métricas

CloudWatch as métricas têm dimensões ou atributos classificáveis, que estão listados abaixo.

Dimensão	Descrição
`AwsAccountId`	Filtra as métricas disponíveis pelo ID da conta.
`PipeName`	Filtra as métricas disponíveis pelo nome do pipe.

Tratamento de erros e solução de problemas do Amazon EventBridge Pipes

Repetir o comportamento e o tratamento de erros

EventBridge O Pipes tenta automaticamente o enriquecimento e a invocação de destino em qualquer falha que possa ser repetida com o serviço de origem, AWS os serviços de enriquecimento ou de destino, ou. EventBridge No entanto, se houver falhas retornadas pelo enriquecimento ou pelas implementações do cliente de destino, o throughput da sondagem de pipes diminuirá gradualmente.

Para erros 4xx quase contínuos (como problemas de autorização com o IAM ou falta de recursos), o pipe pode ser desativado automaticamente com uma mensagem explicativa no `StateReason`.

Erros de invocação de pipes e comportamento de repetição

Ao invocar um pipe, dois tipos principais de erros podem ocorrer: erros internos do pipe e erros de invocação do cliente.

Erros internos do pipe

Os erros internos do Pipe são erros resultantes de aspectos da invocação gerenciados pelo serviço EventBridge Pipes.

Estes tipos de erros podem incluir problemas como:

- Uma falha na conexão HTTP ao tentar invocar o serviço de destino do cliente
- Uma queda transitória na disponibilidade do próprio serviço de pipe.

Em geral, o EventBridge Pipes repete erros internos um número indefinido de vezes e para somente quando o registro expira na fonte.

Para canais com uma fonte de fluxo, o EventBridge Pipes não contabiliza novas tentativas de erros internos em relação ao número máximo de tentativas especificado na política de repetição da fonte de fluxo. Para pipes com uma fonte do Amazon SQS, o EventBridge Pipes não contabiliza novas tentativas de erros internos em relação à contagem máxima de recebimento da fonte do Amazon SQS.

Erros de invocação do cliente

Os erros de invocação do cliente são erros resultantes da configuração ou do código gerenciado pelo usuário.

Estes tipos de erros podem incluir problemas como:

- Permissões insuficientes no pipe para invocar o destino.
- Um erro lógico em um cliente invocado de forma síncrona Lambda, Step Functions, destino da API ou endpoint do API Gateway do cliente.

Para erros de invocação do cliente, o EventBridge Pipes faz o seguinte:

- Para tubos com uma fonte de fluxo, o EventBridge Pipes tenta novamente até os tempos máximos de repetição configurados na política de repetição de tubulação ou até que a idade máxima do registro expire, o que ocorrer primeiro.
- Para pipes com uma fonte Amazon SQS, o EventBridge Pipes repete um erro do cliente até a contagem máxima de recebimento na fila de origem.
- Para canais com uma fonte Apache Kafka ou Amazon MQ, EventBridge repita os erros do cliente da mesma forma que repete os erros internos.

Para pipes com destinos de computação, você deve invocar o pipe de forma síncrona para que o EventBridge Pipes esteja ciente de quaisquer erros de tempo de execução gerados pela lógica de computação do cliente e tente novamente esses erros. Os Pipes não podem repetir os erros gerados pela lógica de um fluxo de trabalho padrão do Step Functions, pois esse destino deve ser invocado de forma assíncrona.

Para o Amazon SQS e fontes de streaming, como Kinesis e DynamoDB, o Pipes oferece suporte ao tratamento parcial de falhas em lote das EventBridge falhas de destino. Para obter mais informações, consulte Lote com falha parcial.

Comportamento da DLQ no pipe

Um pipe herda o comportamento da fila de mensagens não entregues (DLQ) da origem:

- Se a fila de origem do Amazon SQS tiver uma DLQ configurada, as mensagens serão automaticamente entregues lá após o número especificado de tentativas.
- Para origens de fluxos, como fluxos do DynamoDB e do Kinesis, você pode configurar uma DLQ para os eventos de pipe e rota. As origens de fluxo do DynamoDB e do Kinesis são compatíveis com as filas do Amazon SQS e os tópicos do Amazon SNS como destinos de DLQ.

Se você especificar uma `DeadLetterConfig` para um pipe com uma origem do Kinesis ou do DynamoDB, certifique-se de que a propriedade `MaximumRecordAgeInSeconds` no pipe seja menor que a `MaximumRecordAge` do evento de origem. `MaximumRecordAgeInSeconds` controla quando a sondagem do pipe desistirá do evento e o entregará à DLQ e `MaximumRecordAge` controla por quanto tempo a mensagem ficará visível no fluxo de origem antes de ser excluída. Portanto, defina `MaximumRecordAgeInSeconds` para um valor menor que a origem `MaximumRecordAge` para que haja um tempo adequado entre o envio do evento para a DLQ e o momento em que ele é automaticamente excluído pela origem para que você determine por que o evento foi para a DLQ.

Para origens do Amazon MQ, a DLQ pode ser configurada diretamente no agente de mensagens.

EventBridge O Pipes não suporta DLQs de primeiro a entrar, primeiro a sair (FIFO) para fontes de streaming.

EventBridge O Pipes não suporta DLQ para fontes de stream do Amazon MSK e streams autogerenciados do Apache Kafka.

Estados de falha do pipe

Criar, excluir e atualizar pipes são operações assíncronas que podem resultar em um estado de falha. Da mesma forma, um pipe pode ser desativado automaticamente devido a falhas. Em todos os casos, o pipe `StateReason` fornece informações para ajudar a solucionar a falha.

Veja a seguinte lista de valores `StateReason` possíveis:

- Fluxo não encontrado. Para continuar o processamento, exclua o pipe e crie um novo.
- O Pipes não tem permissões necessárias para realizar operações de fila (sqs:ReceiveMessage, sqs: DeleteMessage e sqs:) GetQueueAttributes
- Erro de conexão. Sua VPC deve ser capaz de se conectar aos pipes. Você pode fornecer acesso configurando um NAT Gateway ou um VPC Endpoint para canalizar dados. Para saber como configurar o gateway NAT ou o VPC Endpoint para dados de canais, consulte a documentação. AWS
- O cluster MSK não tem grupos de segurança associados a ele

Um pipe pode ser interrompido automaticamente com uma atualização `StateReason`. As possíveis causas incluem:

- Um fluxo de trabalho padrão do Step Functions configurado como um enriquecimento.
- Um fluxo de trabalho padrão do Step Functions configurado como um destino a ser invocado de forma síncrona.

Falhas de criptografia personalizadas

Se você configurar uma fonte para usar uma chave de criptografia AWS KMS personalizada (CMK), em vez de uma AWS KMS chave AWS gerenciada, você deve dar explicitamente a permissão de descriptografia da Execution Role do pipe. Para fazer isso, inclua a seguinte permissão adicional na política personalizada da CMK:

```
  {
      "Sid": "Allow Pipes access",
      "Effect": "Allow",
      "Principal": {
          "AWS": "arn:aws:iam::01234567890:role/service-role/
Amazon_EventBridge_Pipe_DDBStreamSourcePipe_12345678"
      },
      "Action": "kms:Decrypt",
      "Resource": "*"
  }
```

Substitua o perfil acima pelo perfil de execução do seu pipe.

Isso vale para todas as fontes de tubulação com AWS KMS CMK, incluindo:

- Amazon DynamoDB Streams
- Amazon Kinesis Data Streams
- Amazon MQ
- Amazon MSK
- Amazon SQS

Tutorial: crie um pipe do EventBridge que filtra os eventos de origem

Neste tutorial, será criado um pipe que conectará uma origem de fluxo do DynamoDB a um destino de fila do Amazon SQS. Isto inclui especificar um padrão de evento para o pipe usar ao filtrar eventos para entrega na fila. Em seguida, o pipe será testado para garantir que somente os eventos desejados sejam entregues.

Pré-requisitos: criar a origem e o destino

Antes de criar o pipe, é preciso criar a origem e o destino aos quais o pipe deve se conectar. Neste caso, um fluxo de dados do Amazon DynamoDB para atuar como origem do pipe e uma fila do Amazon SQS como destino do pipe.

Para simplificar esta etapa, o AWS CloudFormation pode ser usado para provisionar os recursos de origem e de destino. Para isso, será criado um modelo do CloudFormation definindo os seguintes recursos:

- A origem do pipe

  Uma tabela do Amazon DynamoDB, chamada `pipe-tutorial-source`, com um fluxo habilitado para fornecer um fluxo ordenado de informações sobre alterações em itens da tabela do DynamoDB.
- O destino do pipe

  Uma fila do Amazon SQS, chamada `pipe-tutorial-target`, para receber o fluxo de eventos do DynamoDB do seu pipe.

Para criar o modelo do CloudFormation para provisionar recursos de pipe

1. Copie o texto do modelo JSON na seção ??? abaixo.
2. Salve o modelo como um arquivo JSON (por exemplo, `~/pipe-tutorial-resources.json`).

Em seguida, use o arquivo de modelo que acabou de criar para provisionar uma pilha do CloudFormation.

> **Note**
>
> Depois de criar sua pilha do CloudFormation, haverá uma cobrança recursos da AWS que ela provisiona.

Fornecimento de pré-requisitos do tutorial usando a AWS CLI

- Execute o seguinte comando da CLI, onde `--template-body` especifica a localização do seu arquivo de modelo:

```
aws cloudformation create-stack --stack-name pipe-tuturial-resources --template-body file://~/pipe-tutorial-resources.json
```

Fornecimento de pré-requisitos do tutorial usando o console do CloudFormation

1. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Em Pilhas, selecione Criar pilha e Com novos recursos (padrão).

O CloudFormation exibe o assistente Criar pilha.

3. Em Pré-requisito: preparar modelo, deixe o padrão, O modelo está pronto, selecionado.
4. Para Especificar modelo, selecione Fazer upload de um arquivo de modelo e Escolher arquivo.
5. Configure a pilha e os recursos que ela provisionará:
   - Para Stack name (Nome da pilha), insira `pipe-tuturial-resources`.
   - Em Parâmetros, deixe os nomes padrão para a tabela do DynamoDB e a fila do Amazon SQS.
   - Escolha Next (Próximo).
6. Escolha Próximo e Enviar.

   O CloudFormation criará a pilha e provisionará os recursos definidos no modelo.

Para obter mais informações sobre o CloudFormation, consulte O que é o AWS CloudFormation? no Manual do usuário do AWS CloudFormation.

Etapa 1: criar o pipe

Com a origem e o destino do pipe provisionados, agora é possível criar o pipe para conectar os dois serviços.

Crie o pipe usando o console do EventBridge

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Pipes.
3. Escolha Criar pipe.
4. Em Nome nomeie seu pipe `pipe-tutorial`.
5. Especifique a origem do fluxo de dados do DynamoDB:
   a. Em Detalhes, em Origem, selecione Fluxo de dados do DynamoDB.

      O EventBridge exibe as configurações de origem específicas do DynamoDB.
   b. Em fluxo do DynamoDB, selecione `pipe-tutorial-source`.

      Deixe a Posição inicial definida como padrão, `Latest`.
   c. Escolha Next (Próximo).
6. Especifique e teste um padrão de evento para filtrar eventos:

A filtragem permite controlar quais eventos os pipes enviam para enriquecimento ou para o destino. O pipe envia apenas eventos que correspondam ao padrão do evento para o enriquecimento ou para o destino.

Para obter mais informações, consulte ???.

> **Note**
>
> Só há uma cobrança pelos eventos enviados para o enriquecimento ou para o destino.

a. Em Evento de amostra : opcional, deixe Eventos da AWS selecionados e certifique-se de que o evento 1 do DynamoDB Stream Sample esteja selecionado.

   Este é o exemplo de evento que será usado para testar nosso padrão de eventos.

b. Em Padrão de evento, insira o seguinte padrão de evento:

```
{
  "eventName": ["INSERT", "MODIFY"]
}
```

c. Escolha Padrão de teste.

   O EventBridge exibe uma mensagem informando que o evento de amostra corresponde ao padrão do evento. Isto ocorre porque o evento de amostra tem um valor `eventName` de `INSERT`.

d. Escolha Next (Próximo).

7. Escolha Próximo para ignorar a especificação de um enriquecimento.

   Neste exemplo, um enriquecimento não será selecionado. Os enriquecimentos permitem que a seleção um serviço para aprimorar os dados da origem antes de enviá-los ao destino. Para obter mais detalhes, consulte ???.

8. Especifique sua fila do Amazon SQS como destino do pipe:

   a. Em Detalhes, para o Serviço de destino, selecione Fila do Amazon SQS.

   b. Em Fila, selecione `pipe-tutorial-target`.

   c. Deixe a seção do Transformador de entrada de destino vazia.

Para obter mais informações, consulte ???.

9. Escolha Criar pipe.

   O EventBridge cria o pipe e exibe a página de detalhes do pipe. O pipe estará pronto quando seu status for atualizado para `Running`.

Etapa 2: confirme os eventos dos filtros de pipe

O Pipe está configurado, mas ainda não recebeu eventos da tabela.

Para testar o pipe, as entradas na tabela do DynamoDB serão atualizadas. Cada atualização gerará eventos que o fluxo do DynamoDB envia para nosso pipe. Alguns corresponderão ao padrão de evento especificado; outros, não. Em seguida, é possível examinar a fila do Amazon SQS para garantir que o pipe entregou somente os eventos que correspondam ao nosso padrão de eventos.

Atualize os itens da tabela para gerar eventos

1. Abra o console do DynamoDB em https://console.aws.amazon.com/dynamodb/.
2. Na barra de navegação à esquerda, selecione Tabelas. Selecione a tabela `pipe-tutorial-source`.

   O DynamoDB exibe a página de detalhes da tabela para `pipe-tutorial-source`.
3. Selecione Explorar itens da tabela e Criar item.

   O DynamoDB exibe a página Criar item.
4. Em Atributos, crie um novo item de tabela:

   a. Em Álbum, insira `Album A`.

   b. Em Artista, insira `Artist A`.

   c. Selecione Create Item (Criar item).
5. Atualize o item da tabela:

   a. Em Itens retornados, escolha Álbum A.

   b. Selecione Adicionar novo atributo e selecione String.

   c. Insira um novo valor de `Song`, com um valor de `Song A`.

   d. Escolha Save changes (Salvar alterações).

6. Exclua o item da tabela:

    a. Em Itens retornados, marque Álbum A.

    b. No menu Ações, selecione Excluir itens.

Três atualizações foram feitas no item da tabela. Isto gera três eventos para o fluxo de dados do DynamoDB:

- Um evento `INSERT` quando o item foi criado.
- Um evento `MODIFY` quando foi adicionado um atributo ao item.
- Um evento `REMOVE` em que o item foi excluído.

No entanto, o padrão de evento especificado para o pipe deve filtrar todos os eventos que não sejam eventos `INSERT` ou `MODIFY`. Em seguida, confirme se o pipe entregou os eventos esperados à fila.

Confirme se os eventos esperados foram entregues à fila

1. Abra o console do Amazon SQS em https://console.aws.amazon.com/sqs/.
2. Escolha a fila `pipe-tutorial-target`.

    O Amazon SQS exibe a página de detalhes da fila.
3. Selecione Enviar e receber mensagens e, em Receber mensagens, escolha Sondagem de mensagens.

    A fila pesquisa o pipe e lista os eventos que recebe.
4. Escolha o nome do evento para ver o JSON do evento que foi entregue.

Deve haver dois eventos na fila: um com um `eventName` de `INSERT` e outro com um `eventName` de `MODIFY`. No entanto, o pipe não entregou o evento para excluir o item da tabela, pois esse evento tinha um `eventName` de `REMOVE`, que não correspondia ao padrão de evento especificado no pipe.

Etapa 3: Limpar os recursos

Primeiro, exclua o pipe em si.

Exclua o pipe usando o console do EventBridge

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Pipes.
3. Selecione o pipe `pipe-tutorial` e escolha Excluir.

Em seguida, exclua a pilha do CloudFormation para evitar a cobrança pelo uso contínuo dos recursos provisionados nela.

Exclua os pré-requisitos do tutorial usando a AWS CLI

- Execute o seguinte comando da CLI, onde `--stack-name` especifica a localização da sua pilha:

```
aws cloudformation delete-stack --stack-name pipe-tuturial-resources
```

Exclua os pré-requisitos do tutorial usando o console do AWS CloudFormation

1. Abra o console do AWS CloudFormation em https://console.aws.amazon.com/cloudformation.
2. Na página Pilhas, selecione a pilha e Excluir.
3. Selecione Excluir para confirmar sua ação.

Modelo do AWS CloudFormation para gerar pré-requisitos

Use o JSON abaixo para criar um modelo do CloudFormation para provisionar os recursos de origem e destino necessários para este tutorial.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",

 "Description" : "Provisions resources to use with the EventBridge Pipes tutorial. You
 will be billed for the AWS resources used if you create a stack from this template.",

  "Parameters" : {
    "SourceTableName" : {
      "Type" : "String",
      "Default" : "pipe-tutorial-source",
```

```
      "Description" : "Specify the name of the table to provision as the pipe source,
 or accept the default."
    },
  "TargetQueueName" : {
    "Type" : "String",
    "Default" : "pipe-tutorial-target",
    "Description" : "Specify the name of the queue to provision as the pipe target, or
 accept the default."
  }
},
  "Resources": {
    "PipeTutorialSourceDynamoDBTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "AttributeDefinitions": [{
            "AttributeName": "Album",
            "AttributeType": "S"
          },
          {
            "AttributeName": "Artist",
            "AttributeType": "S"
          }

        ],
        "KeySchema": [{
            "AttributeName": "Album",
            "KeyType": "HASH"

          },
          {
            "AttributeName": "Artist",
            "KeyType": "RANGE"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 10,
          "WriteCapacityUnits": 10
        },
        "StreamSpecification": {
          "StreamViewType": "NEW_AND_OLD_IMAGES"
        },
        "TableName": { "Ref" : "SourceTableName" }
      }
    },
```

```
    "PipeTutorialTargetQueue": {
      "Type": "AWS::SQS::Queue",
      "Properties": {
        "QueueName": { "Ref" : "TargetQueueName" }
      }
    }
  }
}
```

Gere um AWS CloudFormation modelo a partir do EventBridge Pipes

AWS CloudFormation permite que você configure e gerencie seus AWS recursos em contas e regiões de forma centralizada e repetível, tratando a infraestrutura como código. CloudFormation faz isso permitindo que você crie modelos, que definem os recursos que você deseja provisionar e gerenciar.

EventBridge permite que você gere modelos a partir dos canais existentes em sua conta, como uma ajuda para ajudá-lo a começar a desenvolver CloudFormation modelos. É possível selecionar um único pipe ou vários pipes para incluir no modelo. Em seguida, você pode usar esses modelos como base para criar pilhas de recursos sob CloudFormation gerenciamento.

Para obter mais informações sobre CloudFormation, consulte o Guia AWS CloudFormation do Usuário.

Para ônibus de eventos, você pode gerar CloudFormation modelos a partir de ônibus de eventos e regras de ônibus de eventos.

Recursos incluídos nos modelos EventBridge do Pipe

Ao EventBridge gerar o CloudFormation modelo, ele cria um AWS::Pipes::Piperecurso para cada canal selecionado. Além disso, EventBridge inclui os seguintes recursos nas condições descritas:

- AWS::Events::ApiDestination

  Se seus canais incluírem destinos de API, como enriquecimentos ou alvos, EventBridge inclua-os no CloudFormation modelo como AWS::Events::ApiDestination recursos.
- AWS::Events::EventBus

Se seus canais incluírem um barramento de eventos como destino, EventBridge inclua-o no CloudFormation modelo como um AWS::Events::EventBus recurso.

- AWS::IAM::Role

  Se você EventBridge criou uma nova função de execução ao configurar o pipe, pode optar por EventBridge incluir essa função no modelo como um AWS::IAM::Role recurso. EventBridge não inclui funções que você cria. (Em ambos os casos, a `RoleArn` propriedade do AWS::Pipes::Pipe recurso contém o ARN da função.)

Considerações ao usar CloudFormation modelos gerados a partir de Pipes EventBridge

Considere os seguintes fatores ao usar um CloudFormation modelo que você gerou a partir de EventBridge:

- EventBridge não inclui nenhuma senha no modelo gerado.

  Você pode editar o modelo para incluir parâmetros de modelo que permitam aos usuários especificar senhas ou outras informações confidenciais ao usar o modelo para criar ou atualizar uma CloudFormation pilha.

  Além disso, os usuários podem usar o Secrets Manager para criar um segredo na região desejada e depois editar o modelo gerado para empregar parâmetros dinâmicos.
- Os destinos no modelo gerado permanecem exatamente como foram especificados no pipe original. Isso poderá resultar em problemas entre regiões se você não editar adequadamente o modelo antes de utilizá-lo para criar pilhas em outras regiões.

  Além disso, o modelo gerado não cria os destinos downstream automaticamente.

Gerando um CloudFormation modelo a partir do EventBridge Pipes

Para gerar um CloudFormation modelo a partir de um ou mais tubos usando o EventBridge console, faça o seguinte:

Para gerar um CloudFormation modelo a partir de um ou mais tubos

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.

2. No painel de navegação, escolha Pipes.
3. Em Tubos, escolha um ou mais tubos que você deseja incluir no CloudFormation modelo gerado.

   Para um único pipe, também é possível escolher o nome do pipe para exibir a página de detalhes do pipe.
4. Escolha CloudFormation Modelo e, em seguida, escolha em qual formato você EventBridge deseja gerar o modelo: JSON ou YAML.

   EventBridge exibe o modelo, gerado no formato selecionado.
5. Se você EventBridge criou uma nova função de execução para qualquer um dos canais selecionados e EventBridge deseja incluir essas funções no modelo, escolha Incluir IAM funções criadas pelo console em seu nome.
6. EventBridge oferece a opção de baixar o arquivo de modelo ou copiar o modelo para a área de transferência.

   - Para baixar o arquivo de modelo, escolha Baixar.
   - Para copiar o modelo para a área de transferência, escolha Copiar.
7. Para sair do modelo, escolha Cancelar.

Como tornar as aplicações tolerantes a falhas regionais com endpoints globais e replicação de eventos

Você pode melhorar a disponibilidade do seu aplicativo com os endpoints EventBridge globais da Amazon. Os endpoints globais ajudam a tornar sua aplicação tolerante a falhas regionais sem custo adicional. Para começar, atribuia uma verificação de integridade do Amazon Route 53 ao endpoint. Quando o failover é iniciado, a verificação de integridade relata um estado "não íntegro". Poucos minutos após o início do failover, todos os eventos personalizados são roteados para um barramento de eventos na região secundária e processados por esse barramento de eventos. Depois que a verificação de integridade relata um estado "saudável", os eventos são processados pelo barramento de eventos na região principal.

Ao usar endpoints globais, é possível ativar a replicação de eventos. A replicação de eventos envia todos os eventos personalizados para os barramentos de eventos nas regiões primária e secundária usando regras gerenciadas.

> **Note**
>
> Se estiver usando barramentos personalizados, será preciso ter um barramento personalizado em cada região com o mesmo nome e na mesma conta para que o failover funcione corretamente.

Tópicos

Objetivos de tempo de recuperação e ponto de recuperação

O objetivo de tempo de recuperação (RTO) é o tempo necessário para que a região secundária comece a receber eventos após uma falha. Para o RTO, o tempo inclui o período para acionar CloudWatch alarmes e atualizar os status das verificações de saúde do Route 53. O objetivo de ponto de recuperação (RPO) é a medida dos dados que não serão processados durante uma falha. No RPO, o tempo inclui eventos que não são replicados para a região secundária e ficam presos na região primária até que o serviço ou a região se recuperem. Com endpoints globais, se seguir nossa orientação prescritiva para configuração de alarmes, poderá esperar que o RTO e o RPO sejam de 360 segundos com um máximo de 420 segundos.

Replicação de eventos

Os eventos são processados na região secundária de forma assíncrona. Isto significa que não é garantido que os eventos sejam processados ao mesmo tempo nas duas regiões. Quando o failover é acionado, os eventos são processados pela região secundária e serão processados pela região primária quando estiverem disponíveis. Habilitar a replicação de eventos aumentará seus custos mensais. Para obter mais informações, consulte os EventBridgepreços da Amazon

É recomendado ativar a replicação de eventos ao configurar endpoints globais pelos seguintes motivos:

- A replicação de eventos ajuda a verificar se seus endpoints globais estão configurados corretamente. Isto ajuda a garantir que você esteja coberto em caso de failover.
- A replicação de eventos é necessária para se recuperar automaticamente de um evento de failover. Se não tiver a replicação de eventos ativada, precisará redefinir manualmente a verificação de integridade do Route 53 para "íntegra" antes que os eventos retornem à região principal.

Carga útil de eventos replicada

A seguir, um exemplo de carga útil de evento replicado.

> **Note**
>
> Em `region`, a região da qual o evento foi replicado é listada.

```
{
    "version": "0",
    "id": "a908baa3-65e5-ab77-367e-527c0e71bbc2",
    "detail-type": "Test",
    "source": "test.service.com",
    "account": "0123456789",
    "time": "1900-01-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:events:us-east-1:0123456789:endpoint/MyEndpoint"
    ],
    "detail": {
        "a": "b"
    }
}
```

Criar um endpoint global

Para configurar um endpoint global, conclua as seguintes etapas:

1. Verifique se você tem barramentos e regras de eventos correspondentes na região primária e secundária.
2. Crie uma verificação de integridade do Route 53 para monitorar seus barramentos de eventos. Para obter ajuda na criação de sua verificação de integridade, escolha Nova verificação de integridade ao criar seu endpoint global.
3. Crie seu endpoint global.

Depois de configurar a verificação de integridade do Route 53, é possível criar um endpoint global.

Para criar um endpoint global usando o console

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Endpoints globais.
3. Escolha Criar Endpoint.
4. Insira um nome e uma descrição para o endpoint.
5. Em Barramento de eventos na região principal, escolha o barramento de eventos ao qual deseja que o endpoint fosse associado.

6. Em Região secundária, escolha a região para a qual gostaria de direcionar os eventos no caso de um failover.

   > **Note**
   >
   > O barramento de eventos na região secundária é preenchido automaticamente e não é editável.

7. Na verificação de integridade do Route 53 para acionar o failover e a recuperação, escolha a verificação de integridade que o endpoint monitorará. Se você ainda não tiver uma verificação de saúde, escolha Nova verificação de saúde para abrir o AWS CloudFormation console e criar uma verificação de saúde usando um CloudFormation modelo.

   > **Note**
   >
   > A falta de dados fará com que a verificação de integridade falhe. Se você só precisar enviar eventos de forma intermitente, considere usar um mais longo MinimumEvaluationPeriodou tratar os dados perdidos como “ausentes” em vez de “violados”.

8. (Opcional) Para Replicação de eventos, faça o seguinte:

   a. Selecione Replicação de eventos ativada.

   b. Em Perfil de execução, escolha se deseja criar um novo perfil do AWS Identity and Access Management ou usar um perfil existente. Faça o seguinte:

      - Escolha Create a new role for this specific resource (Criar uma função para este recurso específico). Opcionalmente, você pode atualizar o nome do perfil para criar um novo perfil.
      - Escolha Usar perfil existente. Em Perfil de execução, escolha o perfil desejado a ser usado.

9. Selecione Create (Criar).

Para criar um endpoint global usando a API

Para criar um endpoint global usando a EventBridge API, consulte CreateEndpointna Amazon EventBridge API Reference.

Para criar um endpoint global usando o AWS CloudFormation

Para criar um endpoint global usando a AWS CloudFormation API, consulte AWS::Events::Endpointso Guia do AWS CloudFormation usuário.

Trabalhando com endpoints globais usando um SDK AWS

> **Note**
>
> Em breve, compatibilidade com C++.

Ao usar um AWS SDK para trabalhar com endpoints globais, lembre-se do seguinte:

- Você precisará ter a biblioteca AWS Common Runtime (CRT) instalada para seu SDK específico. Se não tiver o CRT instalado, receberá uma mensagem de exceção indicando o que precisa ser instalado. Para mais informações, consulte:
  - Bibliotecas do AWS Common Runtime (CRT)
  - lajes/aws-crt-java
  - lajes/aws-crt-nodejs
  - lajes/aws-crt-python
- Depois de criar um endpoint global, será preciso adicionar o `endpointId` e o `EventBusName` a todas as chamadas `PutEvents` que usar.
- Endpoints globais são compatíveis com o Signature versão 4A. Esta versão do SigV4 permite que as solicitações sejam assinadas para várias Regiões da AWS. Isso é útil em operações de API que podem resultar em acesso a dados de uma das várias regiões. Ao usar o AWS SDK, você fornece suas credenciais e as solicitações para endpoints globais usarão o Signature versão 4A sem configuração adicional. Para obter mais informações sobre SigV4A, consulte Assinatura de solicitações de API da AWS na Referência geral da AWS .

  Se você solicitar credenciais temporárias do AWS STS endpoint global (sts.amazonaws.com), vende credenciais que, por padrão, não são AWS STS compatíveis com SigV4a. Consulte Gerenciando AWS STS em uma AWS região no Guia do AWS Identity and Access Management usuário para obter mais informações.

Regiões disponíveis

As seguintes regiões são compatíveis com endpoints globais:

- Leste dos EUA (Norte da Virgínia)
- Leste dos EUA (Ohio)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milão)
- Europa (Paris)
- Europa (Estocolmo)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Osaka)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- América do Sul (São Paulo)

Práticas recomendadas para trabalhar com os endpoints globais do Amazon EventBridge

As práticas recomendadas a seguir são aconselháveis ao configurar endpoints globais.

Tópicos

- Habilitar a replicação de eventos
- Como evitar o controle de utilização de eventos
- Como usar métricas de assinantes nas verificações de integridade do Amazon Route 53

Habilitar a replicação de eventos

É altamente recomendável que ativar a replicação e processar seus eventos na região secundária atribuída ao seu endpoint global. Isto garante que sua aplicação na região secundária seja configurada corretamente. Também é preciso ativar a replicação para garantir a recuperação automática na região principal após a mitigação de um problema.

Os IDs de eventos podem mudar nas chamadas de API. Portanto, correlacionar eventos entre regiões exige que exista um identificador exclusivo e imutável. Os consumidores também devem ser projetados com a idempotência em mente. Dessa forma, se estiver replicando eventos ou reproduzindo-os de arquivos, não haverá efeitos colaterais do processamento dos eventos nas duas regiões.

Como evitar o controle de utilização de eventos

Para evitar que os eventos sejam limitados, é recomendado atualizar seus limites `PutEvents` e os limites de destino para que sejam consistentes em todas as regiões.

Como usar métricas de assinantes nas verificações de integridade do Amazon Route 53

Evite incluir métricas de assinantes em suas verificações de integridade do Amazon Route 53. Incluir essas métricas pode fazer com que seu publicador faça failover para as regiões secundárias se um assinante encontrar um problema, apesar de todos os outros assinantes permanecerem saudáveis na região primária. Se um de seus assinantes não conseguir processar eventos na região principal, a replicação deve ser ativada para garantir que seu assinante na região secundária possa processar eventos com êxito.

Modelo do AWS CloudFormation para configurar a verificação de integridade do Route 53

Ao usar endpoints globais, você precisa fazer uma verificação de integridade do Route 53 para monitorar o status de suas regiões. O modelo a seguir define um alarme do Amazon CloudWatch e o usa para definir uma verificação de integridade do Route 53.

Tópicos

- Modelo do AWS CloudFormation para definir uma verificação de integridade do Route 53
- Propriedades do modelo de alarmes do CloudWatch

- Propriedades do modelo de verificações de integridade do Route 53

Modelo do AWS CloudFormation para definir uma verificação de integridade do Route 53

Use o modelo a seguir para definir sua verificações de integridade do Route 53.

```
Description: |-
  Global endpoints health check that will fail when the average Amazon EventBridge
  latency is above 30 seconds for a duration of 5 minutes. Note, missing data will
  cause the health check to fail, so if you only send events intermittently, consider
  changing the heath check to use a longer evaluation period or instead treat missing
  data as 'missing' instead of 'breaching'.

Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: "Global endpoint health check alarm configuration"
        Parameters:
          - HealthCheckName
          - HighLatencyAlarmPeriod
          - MinimumEvaluationPeriod
          - MinimumThreshold
          - TreatMissingDataAs
    ParameterLabels:
      HealthCheckName:
        default: Health check name
      HighLatencyAlarmPeriod:
        default: High latency alarm period
      MinimumEvaluationPeriod:
        default: Minimum evaluation period
      MinimumThreshold:
        default: Minimum threshold
      TreatMissingDataAs:
        default: Treat missing data as

Parameters:
  HealthCheckName:
    Description: Name of the health check
    Type: String
    Default: LatencyFailuresHealthCheck
```

```
  HighLatencyAlarmPeriod:
    Description: The period, in seconds, over which the statistic is applied. Valid
 values are 10, 30, 60, and any multiple of 60.
    MinValue: 10
    Type: Number
    Default: 60
  MinimumEvaluationPeriod:
    Description: The number of periods over which data is compared to the specified
 threshold. You must have at least one evaluation period.
    MinValue: 1
    Type: Number
    Default: 5
  MinimumThreshold:
    Description: The value to compare with the specified statistic.
    Type: Number
    Default: 30000
  TreatMissingDataAs:
    Description: Sets how this alarm is to handle missing data points.
    Type: String
    AllowedValues:
      - breaching
      - notBreaching
      - ignore
      - missing
    Default: breaching

Mappings:
  "InsufficientDataMap":
    "missing":
      "HCConfig": "LastKnownStatus"
    "breaching":
      "HCConfig": "Unhealthy"

Resources:
  HighLatencyAlarm:
      Type: AWS::CloudWatch::Alarm
      Properties:
        AlarmDescription: High Latency in Amazon EventBridge
        MetricName: IngestionToInvocationStartLatency
        Namespace: AWS/Events
        Statistic: Average
        Period: !Ref HighLatencyAlarmPeriod
        EvaluationPeriods: !Ref MinimumEvaluationPeriod
        Threshold: !Ref MinimumThreshold
```

```
        ComparisonOperator: GreaterThanThreshold
        TreatMissingData: !Ref TreatMissingDataAs

  LatencyHealthCheck:
      Type: AWS::Route53::HealthCheck
      Properties:
        HealthCheckTags:
          - Key: Name
            Value: !Ref HealthCheckName
        HealthCheckConfig:
          Type: CLOUDWATCH_METRIC
          AlarmIdentifier:
            Name:
              Ref: HighLatencyAlarm
            Region: !Ref AWS::Region
          InsufficientDataHealthStatus: !FindInMap [InsufficientDataMap, !Ref
 TreatMissingDataAs, HCConfig]

Outputs:
  HealthCheckId:
    Description: The identifier that Amazon Route 53 assigned to the health check when
 you created it.
    Value: !GetAtt LatencyHealthCheck.HealthCheckId
```

Os IDs de eventos podem mudar nas chamadas de API. Portanto, correlacionar eventos entre regiões exige que exista um identificador exclusivo e imutável. Os consumidores também devem ser projetados com a idempotência em mente. Dessa forma, se estiver replicando eventos ou reproduzindo-os de arquivos, não haverá efeitos colaterais do processamento dos eventos nas duas regiões.

Propriedades do modelo de alarmes do CloudWatch

> **Note**
>
> Para todos os campos **`editable`**, considere seu throughput por segundo. Se só envia eventos de forma intermitente, considere alterar a verificação de integridade para usar um período de avaliação mais longo ou, em vez disso, tratar os dados ausentes como se fossem `missing` em vez de `breaching`.

As seguintes propriedades são usadas na seção de alarme do CloudWatch do modelo:

Métrica	Descrição
`AlarmDescription`	A descrição do alarme. Padrão: **`High Latency in Amazon EventBridge`**
`MetricName`	O nome da métrica associada ao alarme. Isso é necessário para um alarme com base em uma métrica. Para um alarme com base em uma expressão matemática, use `Metrics`, e você não pode especificar `MetricName` . Padrão: IngestionToInvocationStartLatency
`Namespace`	O namespace da métrica associada ao alarme. Isso é necessário para um alarme com base em uma métrica. Para um alarme com base em uma expressão matemática, não é possível especificar `Namespace` , em vez disso, use `Metrics`. Padrão: `AWS/Events`
`Statistic`	A estatística da métrica associada ao alarme, diferente do percentil. Padrão: médio
`Period`	O período, em segundos, durante o qual a estatística é aplicada. Isso é necessário para um alarme com base em uma métrica. Os valores válidos são 10, 30, 60 e qualquer múltiplo de 60. Padrão: **`60`**
`EvaluationPeriods`	O número de períodos com os quais os dados são comparados ao limite especificado. Se você estiver configurando um alarme que exija que vários pontos de dados consecutivos estejam em violação para acionar o alarme, esse valor especificará esse número. Se você estiver definindo um alarme “M de N”, esse valor será o N e `DatapointsToAlarm` será o M. Padrão: **`5`**
`Threshold`	O valor para comparar com a estatística especificada.

Métrica	Descrição
	Padrão: **`30,000`**
`ComparisonOperator`	A operação aritmética a ser usada ao comparar a estatística e o limite especificados. O valor da estatística especificada é usado como o primeiro operando. Padrão: `GreaterThanThreshold`
`TreatMissingData`	Define como esse alerta deve lidar com pontos de dados ausentes. Valores válidos: `breaching` , `notBreaching` , `ignore` e `missing` Padrão: `breaching`

Propriedades do modelo de verificações de integridade do Route 53

> **Note**
>
> Para todos os campos **`editable`**, considere seu throughput por segundo. Se só envia eventos de forma intermitente, considere alterar a verificação de integridade para usar um período de avaliação mais longo ou, em vez disso, tratar os dados ausentes como se fossem `missing` em vez de `breaching`.

As seguintes propriedades são usadas na seção de verificação de integridade do Route 53 do modelo:

Métrica	Descrição
`HealthCheckName`	O nome da verificação de integridade. Padrão: **`LatencyFailuresHealthCheck`**
`InsufficientDataHealthStatus`	Quando o CloudWatch tem dados insuficientes sobre a métrica para determinar o estado do alarme, o status que você deseja que o Amazon Route 53 atribua à verificação de integridade

Métrica	Descrição
	Valores válidos: • `Healthy`: o Route 53 considera a verificação de integridade como íntegra. • `Unhealthy` : o Route 53 considera a verificação de integridade como não íntegra. • `LastKnownStatus` : o Route 53 usa o status da verificação de integridade obtido na última vez em que o CloudWatch apresentou dados suficientes para determinar o estado do alarme. Para novas verificações de integridade que não têm último status conhecido, o status padrão indicará a integridade como íntegra. Padrão: não íntegro **Note** Este campo é atualizado com base na entrada do campo `TreatMissingData` . If `TreatingMissingData` for definido como `Missing`, será atualizado para `LastKnown Status` . Se `TreatingMissingData` for definido como `Breaching` , será atualizado para `Unhealthy` .

EventBridge Esquemas da Amazon

Um esquema define a estrutura dos eventos para os quais são enviados EventBridge. EventBridge fornece esquemas para todos os eventos gerados pelos AWS serviços. Também é possível criar ou fazer upload de esquemas, ou automaticamente inferir esquemas diretamente de eventos em um barramento de eventos. Depois de ter encontrado ou criado um esquema para um evento, faça download das vinculações de código para linguagens de programação populares e acelere o desenvolvimento. Você pode trabalhar com vinculações de código para esquemas e gerenciar esquemas a partir do EventBridge console, usando a API, ou diretamente no seu IDE usando os kits de ferramentas. AWS Para criar aplicações com tecnologia sem servidor que usam eventos, use o AWS Serverless Application Model.

> **Note**
>
> Ao usar o atributo transformador de entrada, o evento original é inferido pela descoberta do esquema, não pelo evento transformado que é enviado ao destino.

EventBridge suporta os formatos OpenAPI 3 e JSONSchema Draft4.

Para o AWS Toolkit for JetBrains e o AWS Toolkit for VS Code, você pode procurar ou pesquisar esquemas e baixar vinculações de código para esquemas diretamente no seu IDE.

O seguinte vídeo fornece uma visão geral dos esquemas e registros de esquemas: Como usar o Schema Registry

Tópicos

- Mascaramento de valor de propriedade da API de registro de esquema
- Encontrando um EventBridge esquema da Amazon
- Registros de EventBridge esquemas da Amazon
- Criação de um EventBridge esquema da Amazon
- Associações EventBridge de código da Amazon

Mascaramento de valor de propriedade da API de registro de esquema

Alguns valores de propriedade de eventos usados para criar um registro de esquema podem conter informações confidenciais do cliente. Para proteger as informações do cliente, os valores serão mascarados com asteriscos (*). Como estamos mascarando esses valores, recomendamos EventBridge não criar aplicativos que dependam explicitamente das seguintes propriedades ou de seus valores:

- CreateSchema— A `Content` propriedade do `requestParameters` corpo
- GetDiscoveredSchema— A `Events` propriedade do `requestParameters` corpo e a `Content` propriedade do `responseElements` corpo
- SearchSchemas— A `keywords` propriedade do `requestParameters`
- UpdateSchema— A `Content` propriedade do `requestParameters`

Encontrando um EventBridge esquema da Amazon

EventBridge inclui esquemas para todos os AWS serviços que geram eventos. Você pode encontrar esses esquemas no EventBridge console ou pode encontrá-los usando a ação `SearchSchemas`da API.

Para encontrar esquemas para AWS serviços no console EventBridge

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Schemas (Esquemas).
3. Na página Esquemas, selecione Registro do esquema de evento da AWS .
   <result>

   A primeira página de esquemas disponíveis será exibida.
   </result>
4. Para encontrar um esquema, em Pesquisar esquemas de AWS eventos, insira um termo de pesquisa.

   A pesquisa retorna as correspondências para o nome e conteúdo dos esquemas disponíveis e exibe quais versões do esquema contém correspondências.
5. Abra um esquema de evento selecionando o nome do esquema.

Registros de EventBridge esquemas da Amazon

Os registros de esquema são contêineres para esquemas. Os registros coletam e organizam esquemas para que os esquemas estejam em grupos lógicos. Os registros de esquema padrão são:

- Todos os esquemas — todos os esquemas dos registros de AWS eventos, descobertos e esquemas personalizados.
- AWS registro do esquema de eventos — Os esquemas integrados.
- Registro de esquemas descobertos: os esquemas descobertos pela descoberta de esquemas.

Também é possível criar registros personalizados para organizar os esquemas que criar ou fizer upload.

Como criar um registro personalizado

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Esquemas e Criar registro.
3. Na página Detalhes do registro insira um Nome.
4. (Opcional) Insira uma descrição para o novo registro
5. Selecione Create (Criar).

Para criar um esquema personalizado em seu novo registro, selecione Criar esquema personalizado. Para adicionar um esquema ao seu registro, selecione esse registro ao criar um novo esquema.

Para criar um registro usando a API, use `CreateRegistry`. Para obter mais informações, consulte Amazon EventBridge Schema Registry API Reference.

Para obter informações sobre como usar o registro do EventBridge esquema AWS CloudFormation, consulte Referência do tipo de EventSchemas recurso em AWS CloudFormation.

Criação de um EventBridge esquema da Amazon

São criados esquemas usando arquivos JSON com a especificação OpenAPI ou a especificação JSONSchema Draft4. Você pode criar ou carregar seus próprios esquemas EventBridge usando um modelo ou gerando um esquema com base no JSON de um evento. Também é possível inferir o esquema de eventos em um barramento de eventos. Para criar um esquema usando a API EventBridge Schema Registry, use a ação da `CreateSchema`API.

Ao escolher entre os formatos OpenAPI 3 e JSONSchema Draft4, considere as seguintes diferenças:

- O formato JSONSchema suporta palavras-chave adicionais que não são compatíveis com a OpenAPI, como `$schema, additionalItems`.
- Há pequenas diferenças na forma como as palavras-chave são tratadas, como `type` e `format`.
- A OpenAPI não é compatível com hiperlinks JSONSchema Hyper-Schema em documentos JSON.
- As ferramentas para OpenAPI tendem a se concentrar no tempo de construção, enquanto as ferramentas para JSONSchema tendem a se concentrar em operações em tempo de execução, como ferramentas de cliente para validação de esquemas.

Recomendamos usar o formato JSONSchema para implementar a validação do lado do cliente para que os eventos enviados estejam em conformidade com o EventBridge esquema. É possível usar o JSONSchema para definir um contrato para documentos JSON válidos e usar um validador de esquema JSON antes de enviar os eventos associados.

Depois de ter um novo esquema, é possível baixar vinculações de código para ajudar a criar aplicações para eventos com esse esquema.

Tópicos

- Crie um esquema usando um modelo
- Editar um modelo de esquema diretamente no console
- Crie um esquema do JSON de um evento
- Crie um esquema de eventos em um barramento de eventos

Crie um esquema usando um modelo

Você pode criar um esquema a partir de um modelo ou editando um modelo diretamente no EventBridge console. Para obter o modelo, o console deve ser baixado. É possível editar o modelo

para que o esquema corresponda aos seus eventos. Depois, faça o upload do novo modelo por meio do console.

Para baixar o arquivo de modelo

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Schema registry (Registro do esquema).
3. Na seção Getting started (Introdução), em Schema template (Modelo de esquema), escolha Download (Fazer download).

Como alternativa, é possível copiar o modelo JSON do exemplo de código a seguir.

```
{
    "openapi": "3.0.0",
    "info": {
      "version": "1.0.0",
      "title": "Event"
    },
    "paths": {},
    "components": {
      "schemas": {
        "Event": {
          "type": "object",
          "properties": {
            "ordinal": {
              "type": "number",
              "format": "int64"
            },
            "name": {
              "type": "string"
            },
            "price": {
              "type": "number",
              "format": "double"
            },
            "address": {
              "type": "string"
            },
            "comments": {
              "type": "array",
              "items": {
```

```
                "type": "string"
              }
            },
            "created_at": {
              "type": "string",
              "format": "date-time"
            }
          }
        }
      }
    }
  }
```

Para fazer upload de um modelo de esquema

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Esquemas e Criar esquema.
3. (Opcional) Selecione ou crie um registro de esquema.
4. Em Detalhes do esquema, insira um nome para o esquema.
5. (Opcional) Insira uma descrição para seu esquema.
6. Em Tipo de esquema, escolha OpenAPI 3.0 ou JSON Schema Draft 4.
7. Com a guia Criar selecionada, arraste o arquivo de esquema para a caixa de texto ou cole a origem do esquema.
8. Escolha Criar.

Editar um modelo de esquema diretamente no console

Para editar um esquema no console

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Esquemas e Criar esquema.
3. (Opcional) Selecione ou crie um registro de esquema.
4. Em Detalhes do esquema, insira um nome para o esquema.
5. Em Tipo de esquema, escolha OpenAPI 3.0 ou JSON Schema Draft 4.
6. (Opcional) Insira uma descrição para o esquema a ser criado.
7. Com a guia Criar selecionada, escolha Carregar modelo.

8. Na caixa de texto, edite o modelo para que o esquema corresponda aos seus eventos.
9. Escolha Criar.

Crie um esquema do JSON de um evento

Se tiver o JSON de um evento, poderá criar automaticamente um esquema para esse tipo de evento.

Para criar um esquema com base no JSON de um evento

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Esquemas e Criar esquema.
3. (Opcional) Selecione ou crie um registro de esquema.
4. Em Schema details (Detalhes do esquema) insira um nome para o esquema.
5. (Opcional) Insira uma descrição para o esquema criado.
6. Em Tipo de esquema, escolha OpenAPI 3.0.

   Não é possível usar JSONSchema ao criar um esquema a partir do JSON de um evento.
7. Selecione Discover from JSON (Descobrir do JSON)
8. Na caixa de texto em JSON, cole ou arraste a origem JSON de um evento.

   Por exemplo, você pode colar a fonte desse AWS Step Functions evento para uma execução com falha.

```
{
    "version": "0",
    "id": "315c1398-40ff-a850-213b-158f73e60175",
    "detail-type": "Step Functions Execution Status Change",
    "source": "aws.states",
    "account": "012345678912",
    "time": "2019-02-26T19:42:21Z",
    "region": "us-east-1",
    "resources": [
      "arn:aws:states:us-east-1:012345678912:execution:state-machine-
name:execution-name"
    ],
    "detail": {
        "executionArn": "arn:aws:states:us-east-1:012345678912:execution:state-
machine-name:execution-name",
```

```
        "stateMachineArn": "arn:aws:states:us-
east-1:012345678912:stateMachine:state-machine",
        "name": "execution-name",
        "status": "FAILED",
        "startDate": 1551225146847,
        "stopDate": 1551225151881,
        "input": "{}",
        "output": null
    }
}
```

9. Escolha Discover schema (Descobrir esquema).
10. EventBridge gera um esquema OpenAPI para o evento. Por exemplo, o esquema a seguir é gerado para o evento Step Functions anterior.

```
{
  "openapi": "3.0.0",
  "info": {
    "version": "1.0.0",
    "title": "StepFunctionsExecutionStatusChange"
  },
  "paths": {},
  "components": {
    "schemas": {
      "AWSEvent": {
        "type": "object",
        "required": ["detail-type", "resources", "detail", "id", "source", "time",
 "region", "version", "account"],
        "x-amazon-events-detail-type": "Step Functions Execution Status Change",
        "x-amazon-events-source": "aws.states",
        "properties": {
          "detail": {
            "$ref": "#/components/schemas/StepFunctionsExecutionStatusChange"
          },
          "account": {
            "type": "string"
          },
          "detail-type": {
            "type": "string"
          },
          "id": {
            "type": "string"
          },
```

```
          "region": {
            "type": "string"
          },
          "resources": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "source": {
            "type": "string"
          },
          "time": {
            "type": "string",
            "format": "date-time"
          },
          "version": {
            "type": "string"
          }
        }
      },
      "StepFunctionsExecutionStatusChange": {
        "type": "object",
        "required": ["output", "input", "executionArn", "name", "stateMachineArn",
"startDate", "stopDate", "status"],
        "properties": {
          "executionArn": {
            "type": "string"
          },
          "input": {
            "type": "string"
          },
          "name": {
            "type": "string"
          },
          "output": {},
          "startDate": {
            "type": "integer",
            "format": "int64"
          },
          "stateMachineArn": {
            "type": "string"
          },
          "status": {
```

```
            "type": "string"
          },
          "stopDate": {
            "type": "integer",
            "format": "int64"
          }
        }
      }
    }
  }
}
```

11. Depois que o esquema tiver sido gerado, escolha Criar.

Crie um esquema de eventos em um barramento de eventos

EventBridge pode inferir esquemas descobrindo eventos. Para inferir esquemas, a descoberta de eventos é ativada em um barramento de eventos e cada esquema exclusivo é adicionado ao registro do esquema, incluindo aqueles para eventos entre contas. Os esquemas descobertos por EventBridge aparecem no registro de esquemas descobertos na página Esquemas.

Se o conteúdo dos eventos no barramento de eventos for alterado, EventBridge criará novas versões do EventBridge esquema relacionado.

> **Note**
>
> Habilitar a descoberta de eventos em um barramento de eventos pode incorrer em custos. Os primeiros cinco milhões de eventos processados a cada mês são gratuitos.

> **Note**
>
> EventBridge infere esquemas de eventos entre contas por padrão, mas você pode desativá-los atualizando a propriedade. `cross-account` Para obter mais informações, consulte Discoverers na Referência da API EventBridge Schema Registry.

Como habilitar a descoberta de esquema em um barramento de eventos

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.

2. No painel de navegação, selecione Event buses (Barramentos de eventos).
3. Execute um destes procedimentos:
   - Para habilitar a descoberta no Barramento de eventos padrão, selecione Iniciar descoberta.
   - Para habilitar a descoberta em um Barramento de evento personalizado, selecione o botão de opção do barramento de eventos personalizado e escolha Iniciar descoberta.

Associações EventBridge de código da Amazon

Você pode gerar vinculações de código para esquemas de eventos para acelerar o desenvolvimento em Golang, Java, Python e. TypeScript As associações de código estão disponíveis para eventos de serviço da AWS , esquemas que você cria e para esquemas que você gera com base em eventos em um barramento de eventos. Você pode gerar vinculações de código para um esquema usando o EventBridge console, a API EventBridge Schema Registry ou em seu IDE com um kit de ferramentas. AWS

Para gerar vinculações de código a partir de um esquema EventBridge

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Schemas (Esquemas).
3. Encontre um esquema para o qual você deseja fazer vinculações de código, examinando os registros de esquema ou procurando por um esquema.
4. Selecione o nome do esquema.
5. Na página de Detalhes do esquema, na seção Versão, escolha Baixar vinculações de código.
6. Na página Download code bindings (Fazer download de vinculações de código), selecione a linguagem das vinculações de código que você deseja fazer download.
7. Selecione Download (Fazer download).

   Pode demorar alguns segundos para que o download seja iniciado. O arquivo de download é um arquivo zip .com vinculações de código para a linguagem selecionada.

Serviços e ferramentas relacionados ao Amazon EventBridge

O Amazon EventBridge funciona com outros serviços da AWS para processar eventos ou invocar um recurso como o destino de uma regra. Para obter mais informações sobre as integrações do EventBridge com outros produtos da AWS, consulte:

Tópicos

- Usar o Amazon EventBridge com endpoints de interface do Amazon VPC
- Integração do Amazon EventBridge com o AWS X-Ray
- Usando EventBridge com o kit de teste de aplicação AWS integrado
- Incluindo EventBridge recursos da Amazon em AWS CloudFormation pilhas

Usar o Amazon EventBridge com endpoints de interface do Amazon VPC

Se usar a Amazon Virtual Private Cloud (Amazon VPC) para hospedar seus recursos da AWS, poderá estabelecer uma conexão privada entre a VPC e o EventBridge. Seus recursos na VPC podem usar esta conexão para se comunicar com o EventBridge Pipes.

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 sua VPC ao EventBridge, basta definir um endpoint da VPC de interface para o EventBridge. O endpoint fornece uma conectividade confiável e escalável ao EventBridge sem a necessidade de um gateway da Internet, de uma instância de conversão de endereços 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 endpoints da VPC de interface são desenvolvidos pelo AWS PrivateLink, que permite a comunicação privada entre os serviços da AWS usando uma interface de rede elástica com endereços IP privados. Para obter mais informações, consulte AWS PrivateLink e endpoints da VPC.

Ao usa um endpoint da VPC de interface privada, os eventos personalizados que sua VPC envia para o EventBridge usam esse endpoint. O EventBridge envia esses eventos para outros serviços da AWS com base nas regras e destinos que você configurou. Depois que os eventos são enviados para outro serviço, é possível recebê-los por meio do endpoint público ou de um endpoint da VPC desse serviço. Por exemplo, se criar uma regra para enviar eventos para uma fila do Amazon SQS, poderá configurar uma interface do endpoint da VPC para que o Amazon SQS receba mensagens dessa fila na sua VPC sem usar o endpoint público.

Disponibilidade

No momento, o EventBridge pe compatível com endpoints da VPC nas seguintes regiões:

- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)
- África (Cidade do Cabo)
- Ásia-Pacífico (Mumbai)

- Ásia-Pacífico (Hyderabad)
- Ásia-Pacífico (Hong Kong)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Melbourne)
- Ásia-Pacífico (Tóquio)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Osaka)
- Canadá (Central)
- Oeste do Canadá (Calgary)
- China (Pequim)
- China (Ningxia)
- Europe (Frankfurt)
- Europa (Zurique)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milão)
- Europa (Espanha)
- Europa (Paris)
- Europa (Estocolmo)
- Oriente Médio (Emirados Árabes Unidos)
- Oriente Médio (Barém)
- América do Sul (São Paulo)
- Israel (Tel Aviv)
- AWS GovCloud (Oeste dos EUA)
- AWS GovCloud (Leste dos EUA)

Como criar um endpoint da VPC para o EventBridge

Para usar o EventBridge Pipes com a VPC, crie um endpoint da VPC de interface para o EventBridge Pipes e escolha com.amazonaws. ***Região*** .events como nome do serviço. Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

Especificações do EventBridge Pipes

A compatibilidade completa do EventBridge Pipes para os endpoints da VPC de interface não está disponível. Para usar as seguintes origens em uma VPC com EventBridge Pipes, veja o seguinte:

- Configuração de rede do Amazon MSK
- Parâmetros de configuração autogerenciados do Apache Kafka
- Configuração de rede do Amazon MQ

Integração do Amazon EventBridge com o AWS X-Ray

É possível usar o AWS X-Ray para rastrear eventos que passam pelo EventBridge. O EventBridge aprova o cabeçalho de rastreamento original para o destino para que os serviços de destino possam rastrear, analisar e depurar.

O EventBridge pode aprovar um cabeçalho de rastreamento para um evento somente se o evento vier de uma solicitação `PutEvents` que passou pelo contexto de rastreamento. O X-Ray não rastreia eventos originados de parceiros terceirizados, eventos programados ou serviços da AWS, e essas origens de eventos não aparecem no mapa do serviço X-Ray.

O X-Ray valida os cabeçalhos de rastreamento, e os cabeçalhos de rastreamento que não são válidos são descartados. No entanto, o evento ainda está sendo processado.

> **Important**
>
> O cabeçalho de rastreamento não está disponível no evento entregue ao destino da invocação.
>
> - Se tiver um arquivo de eventos, o cabeçalho de rastreamento não estará disponível nos eventos arquivados. Se reproduzir eventos arquivados, o cabeçalho de rastreamento não será incluído.
> - Se tiver uma fila de mensagens não entregues (DLQ), o cabeçalho de rastreamento será incluído na solicitação `SendMessage` que envia o evento para a DLQ. Se recuperar eventos (mensagens) da DLQ usando `ReceiveMessage`, o cabeçalho de rastreamento associado ao evento será incluído no atributo de mensagem do Amazon SQS, mas não será incluído na mensagem do evento.

Para obter informações sobre como um nó de eventos do EventBridge conecta os serviços de origem e destino, consulte Como visualizar a origem e os destinos no mapa do serviço X-Ray no Guia do Desenvolvedor do AWS X-Ray.

É possível aprovar as seguintes informações do cabeçalho de rastreamento por meio do EventBridge:

- Cabeçalho HTTP padrão: o X-Ray SDK preenche automaticamente o cabeçalho de rastreamento como cabeçalho `X-Amzn-Trace-Id` HTTP para todos os destinos de invocação. Para saber mais

sobre o cabeçalho HTTP padrão, consulte o cabeçalho de rastreamento no Guia do desenvolvedor do AWS X-Ray.

- Atributo **`TraceHeader`** do sistema: `TraceHeader` é um atributo PutEventsRequestEntry reservado pelo EventBridge para transportar o cabeçalho de rastreamento do X-Ray até um destino. Se `PutEventsRequestEntry` também é usado, `PutEventsRequestEntry` substitui o cabeçalho de rastreamento HTTP.

> **Note**
>
> O cabeçalho do rastreamento não conta para o tamanho do evento `PutEventsRequestEntry`. Para obter mais informações, consulte Calculando o tamanho da entrada em EventBridge PutEvents eventos da Amazon.

O vídeo a seguir demonstra o uso do X-Ray e do EventBridge juntos: Como usar AWS X-Ray para rastreamento

Usando EventBridge com o kit de teste de aplicação AWS integrado

Quando você cria aplicativos compostos por serviços sem servidor, como Lambda ou EventBridge Step Functions, muitos dos seus componentes de arquitetura não podem ser implantados em seu desktop, mas existem apenas na nuvem. AWS Ao contrário de trabalhar com aplicativos implantados localmente, esses tipos de aplicativos se beneficiam de estratégias baseadas em nuvem para realizar testes automatizados. AWS O Integrated Application Test Kit (AWS IATK) ajuda você a implementar algumas dessas estratégias para seus aplicativos.

AWS A IATK é uma biblioteca de software que ajuda você a escrever testes automatizados para aplicativos baseados em nuvem.

EventBridge integração com o AWS IATK

Você pode usar EventBridge eventos e barramentos de eventos com a AWS IATK para implementar seus testes automatizados, incluindo:

Implementar as baterias de testes

Para escrever testes de integração para arquiteturas orientadas por eventos, estabeleça limites lógicos dividindo a aplicação em subsistemas. Uma técnica útil para testar subsistemas é criar baterias de testes, ou seja, recursos que você cria especificamente para testar subsistemas.

Por exemplo, um teste de integração pode iniciar um processo de subsistema passando um evento de teste de entrada para ele. AWS A IATK pode criar um equipamento de teste para você que escuta os eventos de saída. EventBridge (Sob o capô, o chicote é composto por uma EventBridge regra que encaminha o evento de saída para o Amazon SQS.) O teste de integração consulta a bateria de testes para examinar a saída e determinar se o teste é aprovado ou não.

Gerar eventos simulados

AWS O IATK fornece a capacidade de gerar eventos simulados a partir de um esquema armazenado no registro do esquema. EventBridge Isso permite gerar um evento simulado e invocar qualquer consumidor (como uma função do Lambda ou uma máquina de estado do Step Functions) com o evento gerado.

Para obter mais informações, consulte Visão geral do kit de teste de aplicativos AWS integrados em GitHub.

Incluindo EventBridge recursos da Amazon em AWS CloudFormation pilhas

AWS CloudFormation permite que você configure e gerencie seus AWS recursos em contas e regiões de forma centralizada e repetível, tratando a infraestrutura como código. CloudFormation faz isso permitindo que você crie modelos, que definem os recursos que você deseja provisionar e gerenciar. Esses recursos podem incluir EventBridge artefatos como barramentos e regras de eventos, canais, esquemas e programações, entre outros. Use esses recursos para incluir EventBridge funcionalidades nas pilhas de tecnologia que você provisiona e gerencia CloudFormation.

EventBridge Recursos da Amazon disponíveis em AWS CloudFormation

EventBridge fornece recursos para uso em CloudFormation modelos nos seguintes namespaces de recursos:

- AWS::Events

  Os exemplos de modelos incluem:

  - Crie um destino de API para PagerDuty
  - Criar um destino da API para o Slack
  - Crie uma conexão com parâmetros ApiKey de autorização
  - Criar uma conexão com parâmetros de autorização do OAuth
  - Criar um endpoint global com replicação de eventos
  - Negar política utilizando várias entidades principais e ações
  - Conceder permissão a uma organização utilizando um barramento de eventos personalizado
  - Criar uma regra entre regiões
  - Criar uma regra que inclua uma fila de mensagens não entregues a um destino
  - Invocar uma função do Lambda regularmente
  - Invocar uma função do Lambda em resposta a um evento
  - Notificar um tópico em resposta a uma entrada de log
- AWS::EventEsquemas
- AWS::Pipes

  Os exemplos de modelos incluem:

  - Criar um pipe com um filtro de eventos
- AWS::Scheduler

Gerando definições EventBridge de recursos da Amazon para AWS CloudFormation modelos

Como ajuda para ajudar você a começar a desenvolver CloudFormation modelos, o EventBridge console permite que você crie CloudFormation modelos a partir dos barramentos de eventos, regras e canais existentes em sua conta.

- ???
- ???
- ???

Trazendo o barramento de eventos padrão sob AWS CloudFormation gerenciamento

Como EventBridge provisiona o barramento de eventos padrão em sua conta automaticamente, você não pode criá-lo usando um CloudFormation modelo, como faria normalmente com qualquer recurso que desejasse incluir em uma CloudFormation pilha. Para incluir o barramento de eventos padrão em uma CloudFormation pilha, você deve primeiro importá-lo para uma pilha. Depois de importar o barramento de eventos padrão para uma pilha, você pode atualizar as propriedades do barramento de eventos conforme desejado.

Para mais informações, consulte ???.

Gerenciando eventos AWS CloudFormation de pilha usando EventBridge

Além de incluir EventBridge recursos em suas CloudFormation pilhas, você pode usar EventBridge para gerenciar os eventos gerados pelas próprias CloudFormation pilhas. CloudFormation envia eventos para EventBridge sempre que uma operação de criação, atualização, exclusão ou detecção de desvios é executada em uma pilha. CloudFormation também envia eventos EventBridge para alterações de status em conjuntos de pilhas e instâncias de conjuntos de pilhas. Você pode usar EventBridge regras para rotear eventos para seus alvos definidos.

Para obter mais informações, consulte Gerenciamento de CloudFormation eventos usando EventBridge o Guia AWS CloudFormation do usuário.

Tutoriais do Amazon EventBridge

O EventBridge se integra a vários serviços da AWS e parceiros de SaaS. Esses tutoriais foram criados para ajudar você a se familiarizar com os conceitos básicos do EventBridge e como ele pode fazer parte da sua arquitetura com tecnologia sem servidor.

Tutoriais:

- Tutoriais de introdução do Amazon EventBridge
- Tutoriais do Amazon EventBridge para a integração com outros serviços da AWS
- Tutoriais do Amazon EventBridge para integração com provedores de SaaS

Tutoriais de introdução do Amazon EventBridge

Os tutoriais a seguir ajudam a explorar os atributos do EventBridge e como usá-los.

Tutoriais:

- Arquivamento e reprodução de eventos do Amazon EventBridge
- Crie uma aplicação de amostra do Amazon EventBridge
- Tutorial: fazer download de vinculações de código para eventos usando o registro de esquemas do EventBridge
- Tutorial: usar o transformador de entrada para personalizar o que o EventBridge aprova para o evento de destino

Arquivamento e reprodução de eventos do Amazon EventBridge

É possível usar o EventBridge para rotear eventos para funções específicas do AWS Lambda usando regras.

Neste tutorial, será criada uma função para usar como destino da regra do EventBridge usando o console do Lambda. Em seguida, será criado um arquivo e uma regra que arquivarão os eventos de teste usando o console do EventBridge. Quando houver eventos neste arquivo, eles poderão estar em reprodução.

Etapas:

- Etapa 1: criar uma função do Lambda
- Etapa 2: criar um arquivo
- Etapa 3: criar regra
- Etapa 4: enviar eventos de teste
- Etapa 5: reproduzir eventos
- Etapa 6: Limpar os recursos

Etapa 1: criar uma função do Lambda

Crie uma função do Lambda para registrar os eventos em log.

Como criar uma função do Lambda:

1. Abra o console do AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Escolha Create function (Criar função).
3. Escolha Author from scratch (Criar do zero).
4. Digite um nome e uma descrição para a função Lambda. Por exemplo, atribua à função o nome `LogScheduledEvent`.
5. Deixe o resto das opções como padrão e escolha Criar função.
6. Na guia Código da página da função, clique duas vezes em index.js.
7. Substitua o código existente em JavaScript pelo seguinte:

```
'use strict';
```

```
exports.handler = (event, context, callback) => {
    console.log('LogScheduledEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Escolha Implantar.

Etapa 2: criar um arquivo

Em seguida, crie o arquivo que conterá todos os eventos de teste.

Para criar um arquivamento

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, selecione Arquivos.
3. Escolha Criar arquivo.
4. Insira um nome e uma descrição para o arquivo. Por exemplo, nomeie o arquivo como `ArchiveTest`.
5. Use os valores padrão para o restante das opções e escolha Próximo.
6. Escolha Criar arquivo.

Etapa 3: criar regra

Crie uma regra para arquivar eventos que são enviados ao barramento de eventos.

Para criar uma regra do

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Insira um nome e uma descrição para a regra. Por exemplo, nomeie a regra como `ARTestRule`.

   Uma regra não pode ter o mesmo nome que outra regra na mesma região e no mesmo barramento de eventos.
5. Em Event bus (Barramento de eventos), escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos

provenientes da sua conta, selecione default. Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.

6. Em Rule type (Tipo de regra), escolha Rule with an event pattern (Regra com um padrão de evento).
7. Escolha Next (Próximo).
8. Em Event source (Origem do evento), escolha Other (Outra).
9. Em Padrão de evento, insira um do seguintes:

```
{
  "detail-type": [
    "customerCreated"
  ]
}
```

10. Escolha Next (Próximo).
11. Em Target types (Tipos de destinos), escolha AWS service (Serviço da).
12. Em Selecionar um destino, escolha a função do Lambda na lista suspensa.
13. Em Função, selecione a função do Lambda criada na seção Etapa 1: criar uma função do Lambda. Neste exemplo, selecione `LogScheduledEvent`.
14. Escolha Next (Próximo).
15. Escolha Next (Próximo).
16. Analise os detalhes da regra e escolha Create rule (Criar regra).

Etapa 4: enviar eventos de teste

Agora que o arquivo e a regra do SNS foram configurados, serão enviados eventos de teste para garantir que o arquivo esteja funcionando corretamente.

> **Note**
>
> Pode levar algum tempo para que os eventos cheguem ao arquivo.

Para enviar eventos de teste (console)

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.

2. No painel de navegação, selecione Event buses (Barramentos de eventos).
3. No quadro Barramento de eventos padrão, escolha Ações, Enviar eventos.
4. Insira a origem de eventos. Por exemplo, `TestEvent`.
5. Em Tipo de detalhe, insira `customerCreated`.
6. Em Detalhes do evento, insira `{}`.
7. Selecione Send (Enviar).

Etapa 5: reproduzir eventos

Depois que os eventos de teste estiverem no arquivo, será possível reproduzi-los.

Para reproduzir eventos arquivados (console)

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Reproduções.
3. Escolha Iniciar nova repetição.
4. Insira um nome e uma descrição para a reprodução. Por exemplo, nomeie a reprodução como `ReplayTest`.
5. Em Origem, selecione o arquivo que você criou na seção Etapa 2: criar arquivo.
6. Em Reproduzir período, faça o seguinte:
   a. Em Hora de início, selecione a data em que enviou os eventos de teste e um horário antes de enviá-los. Por exemplo, `2021/08/11` e `08:00:00`.
   b. Em Hora de término, selecione a data e hora atuais. Por exemplo, `2021/08/11` e `09:15:00`.
7. Escolha Iniciar repetição.

Etapa 6: Limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir os recursos da AWS que não estão mais sendo usados, são evitadas cobranças desnecessárias em sua conta da AWS.

Para excluir as funções do Lambda

1. Abra a página Funções do console do Lambda.

2. Selecione as funções que foram criadas.
3. Escolha Actions, Delete.
4. Escolha Delete (Excluir).

Para excluir os arquivos do EventBridge

1. Abra a página Arquivos do console do EventBridge.
2. Selecione os arquivos que foram criados.
3. Escolha Delete (Excluir).
4. Insira o nome do arquivo e escolha Excluir.

Para excluir as regras do EventBridge

1. Abra a página Regras no console do EventBridge.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Crie uma aplicação de amostra do Amazon EventBridge

É possível usar o EventBridge para rotear eventos para funções específicas do Lambda usando regras.

Neste tutorial, serão usados a AWS CLI, o Node.js e o código no repositório do GitHub para criar o seguinte:

- Uma função AWS Lambda que produz eventos para transações bancárias em caixas eletrônicos.
- Para usar uma função do Lambda como destinos de uma regra do EventBridge.
- e a regra que roteia os eventos criados para a função downstream correta com base em um padrão de evento.

Este exemplo usa modelos do AWS SAM para definir as regras do EventBridge. Para saber mais sobre o uso de modelos do AWS SAM com o EventBridge, consulte ???.

No repositório, o subdiretório atmProducer contém `handler.js`, que representa o serviço ATM que produz eventos. Este código é um manipulador do Lambda escrito em Node.js e publica eventos no EventBridge por meio do AWS SDK usando esta linha de código em JavaScript.

```
const result = await eventbridge.putEvents(params).promise()
```

Esse diretório também contém `events.js`, listando várias transações de teste em uma matriz de entradas. Um único evento é definido em JavaScript da seguinte forma:

```
{
  // Event envelope fields
  Source: 'custom.myATMapp',
  EventBusName: 'default',
  DetailType: 'transaction',
  Time: new Date(),

  // Main event body
  Detail: JSON.stringify({
    action: 'withdrawal',
    location: 'MA-BOS-01',
    amount: 300,
    result: 'approved',
    transactionId: '123456',
```

```
    cardPresent: true,
    partnerBank: 'Example Bank',
    remainingFunds: 722.34
  })
}
```

A seção Detalhes do evento especifica os atributos da transação. Isso inclui a localização do caixa eletrônico, o valor, o banco parceiro e o resultado da transação.

O arquivo `handler.js` no subdiretório atmConsumer contém três funções:

```
exports.case1Handler = async (event) => {
  console.log('--- Approved transactions ---')
  console.log(JSON.stringify(event, null, 2))
}

exports.case2Handler = async (event) => {
  console.log('--- NY location transactions ---')
  console.log(JSON.stringify(event, null, 2))
}

exports.case3Handler = async (event) => {
  console.log('--- Unapproved transactions ---')
  console.log(JSON.stringify(event, null, 2))
}
```

Cada função recebe eventos de transação, que são registrados por meio das declarações `console.log` no Amazon CloudWatch Logs. As funções do consumidor operam independentemente do produtor e desconhecem a origem dos eventos.

A lógica de roteamento está contida nas regras do EventBridge que são implantadas pelo modelo da aplicação do AWS SAM. As regras avaliam o fluxo de entrada de eventos e roteiam os eventos correspondentes para as funções do Lambda de destino.

As regras usam padrões de eventos que são objetos JSON com a mesma estrutura dos eventos aos quais correspondem. Aqui está o padrão do evento para uma das regras.

```
{
  "detail-type": ["transaction"],
  "source": ["custom.myATMapp"],
  "detail": {
    "location": [{
```

```
      "prefix": "NY-"
    }]
  }
}
```

Etapas:

- Pré-requisitos
- Etapa 1: criar uma aplicação
- Etapa 2: executar a aplicação
- Etapa 3: conferir os logs e verificar se a aplicação funciona
- Etapa 4: limpar os recursos

Pré-requisitos

Para concluir este tutorial, serão necessários os seguintes recursos:

- Uma conta da AWS. Crie uma conta da AWS, se ainda não tiver uma.
- AWS CLI instalada. Para instalar a AWS CLI, consulte Instalar, atualizar e desinstalar a AWS CLI versão 2.
- Node.js 12.x instalado. Para instalar o Node.js, consulte Downloads.

Etapa 1: criar uma aplicação

Para configurar a aplicação de exemplo, serão usados a AWS CLI e o Git para criar os recursos da AWS necessários.

Para criar o aplicativo

1. Faça login na AWS.
2. Instale o Git e instale a CLI do AWS Serverless Application Model na sua máquina local.
3. Crie um novo diretório e navegue até esse diretório em um terminal.
4. Na linha de comando, insira `git clone https://github.com/aws-samples/amazon-eventbridge-producer-consumer-example`.
5. Na linha de comando, execute o seguinte comando:

```
cd ./amazon-eventbridge-producer-consumer-example
```

```
sam deploy --guided
```

6. No terminal, faça o seguinte:

   a. Em **`Stack Name`**, insira um nome para a pilha. Por exemplo, nomeie a pilha como `Test`.

   b. Em **`AWS Region`**, insira a região. Por exemplo, `us-west-2`.

   c. Em **`Confirm changes before deploy`**, digite Y.

   d. Em **`Allow SAM CLI IAM role creation`**, insira Y

   e. Em **`Save arguments to configuration file`**, insira Y

   f. Em **`SAM configuration file`**, digite `samconfig.toml`.

   g. Em **`SAM configuration environment`**, digite `default`.

Etapa 2: executar a aplicação

Agora que os recursos foram configurados, o console será usado para testar as funções.

Executar o aplicativo

1. Abra o console do Lambda na mesma região em que implementou a aplicação do AWS SAM.
2. Existem quatro funções do Lambda com o prefixo atm-demo. Selecione a função atmProducerFn e escolha Ações, Testar.
3. Insira `Test` para o Nome.
4. Escolha Test (Testar).

Etapa 3: conferir os logs e verificar se a aplicação funciona

Agora que executou a aplicação, usará o console para verificar os CloudWatch Logs.

Para verificar os logs

1. Abra o console do CloudWatch na mesma região em que executou a aplicação do AWS SAM.
2. Escolha Logs e depois escolha Log groups (Grupo de logs).
3. Selecione o grupo de logs que contém atmConsumerCase1. São vistos dois fluxos representando as duas transações aprovadas pelo caixa eletrônico. Escolha o fluxo de logs para visualizar a saída.

4. Navegue de volta para a lista de grupos de logs e selecione o grupo de log que contém atmConsumerCase2. Serão vistos dois fluxos representando as duas transações correspondentes ao filtro de localização de Nova York.
5. Navegue de volta para a lista de grupos de logs e selecione o grupo de log que contém atmConsumerCase3. Abra o fluxo para ver as transações negadas.

Etapa 4: limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir os recursos da AWS que não estão mais sendo usados, são evitadas cobranças desnecessárias em sua conta da AWS.

Para excluir as regras do EventBridge

1. Abra a página Regras no console do EventBridge.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Para excluir as funções do Lambda

1. Abra a página Funções do console do Lambda.
2. Selecione as funções que foram criadas.
3. Escolha Actions, Delete.
4. Escolha Delete (Excluir).

Para excluir o grupo de logs do CloudWatch Logs

1. Abra o console do CloudWatch.
2. Escolha Logs, Grupos de logs.
3. Selecione os grupos de logs que foram criados neste tutorial.
4. Escolha Actions (Ações), Delete log group(s) (Excluir grupo(s) de log).
5. Escolha Delete (Excluir).

Tutorial: fazer download de vinculações de código para eventos usando o registro de esquemas do EventBridge

É possível gerar vinculações de código para esquemas de eventos para acelerar o desenvolvimento com Java, Python e TypeScript. É possível obter vinculações de código para serviços da AWS existentes, esquemas criados e esquemas gerados com base em eventos em um barramento de eventos. É possível gerar associações de código para um esquema usando uma das formas a seguir:

- Console do EventBridge
- API de registro do esquema EventBridge
- Seu IDE com um AWS Toolkit

Neste tutorial, as vinculações de código de um esquema de código serão geradas e baixadas do EventBridge para os eventos de um serviço da AWS.

Para gerar vinculações de código de um esquema do EventBridge

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Schemas (Esquemas).
3. Selecione a guia Registro do esquema de evento da AWS.
4. Encontre um esquema para um serviço da AWS para o qual você deseja fazer vinculações de código, navegando até o registro do esquema ou pesquisando um esquema.
5. Selecione o nome do esquema.
6. Na página de Detalhes do esquema, na seção Versão, selecione Baixar vinculações de código.
7. Na página Download code bindings (Fazer download de vinculações de código), selecione a linguagem das vinculações de código que você deseja fazer download.
8. Selecione Download (Fazer download).

   Pode demorar alguns segundos para que o download seja iniciado. O arquivo de download será um arquivo .zip com vinculações de código para a linguagem selecionada.
9. Descompacte o arquivo baixado e adicione-o ao seu projeto.

   O pacote baixado contém um arquivo README que explica como configurar as dependências do pacote em várias estruturas.

Use essas vinculações de código em seu próprio código para ajudar a criar aplicações rapidamente usando esse evento do EventBridge.

Tutorial: usar o transformador de entrada para personalizar o que o EventBridge aprova para o evento de destino

É possível usar o texto de um Transformador de entrada no EventBridge para personalizar o texto de um evento antes de enviá-lo a um destino de uma regra.

Para fazer isso, defina vários caminhos JSON do evento e atribua as saídas para variáveis diferentes. Em seguida, use essas variáveis no modelo de entrada. Não é possível efetuar o escape dos caracteres < e >. Para obter mais informações, consulte Transformação EventBridge de insumos da Amazon.

> **Note**
>
> Se você especificar uma variável para corresponder a um caminho JSON que não existe no evento, essa variável não será criada e não aparecerá na saída.

Neste tutorial, é criada uma regra que combina um evento com`detail-type: "customerCreated"`. O transformador de entrada mapeia a variável `type` para o caminho JSON do tipo $.detail-type a partir do evento. Em seguida, o EventBridge coloca a variável no modelo de entrada "Este evento foi <type>". O resultado é a seguinte mensagem do Amazon SNS.

```
"This event was of customerCreated type."
```

Etapas:

- Etapa 1: criar um tópico do Amazon SNS
- Etapa 2: criar uma assinatura do Amazon SNS
- Etapa 3: criar uma regra
- Etapa 4: enviar eventos de teste
- Etapa 5: confirmar o êxito
- Etapa 6: Limpar os recursos

Etapa 1: criar um tópico do Amazon SNS

Crie um tópico para receber os eventos do EventBridge.

Para criar um tópico

1. Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No painel de navegação, escolha Topics (Tópicos).
3. Escolha Create topic.
4. Em Type (Tipo), escolha Standard (Padrão).
5. Insira **`eventbridge-IT-test`** como o nome do tópico.
6. Escolha Create topic.

Etapa 2: criar uma assinatura do Amazon SNS

Criar uma assinatura para receber e-mails com as informações transformadas.

Criar uma assinatura

1. Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No painel de navegação, escolha Subscriptions.
3. Selecione Create subscription.
4. Em ARN do tópico, escolha o tópico criado na etapa 1. Para este tutorial, escolha eventbridge-IT-test.
5. Em Protocol (Protocolo), escolha Email.
6. Para Endpoint, insira seu endereço de e-mail.
7. Selecione Create subscription.
8. Confirme a assinatura escolhendo Confirmar assinatura no e-mail que você recebe das notificações AWS.

Etapa 3: criar uma regra

Crie uma regra para usar o transformador de entrada para personalizar as informações do estado da instância que vão para um destino.

Para criar uma regra do

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).

3. Escolha Create rule (Criar regra).
4. Insira um nome e uma descrição para a regra. Por exemplo, nomeie a regra como `ARTestRule`
5. Em Event bus (Barramento de eventos), escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, selecione default. Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Rule type (Tipo de regra), escolha Rule with an event pattern (Regra com um padrão de evento).
7. Escolha Next (Próximo).
8. Em Event source (Origem do evento), escolha Other (Outra).
9. Em Padrão de evento, insira um do seguintes:

```
{
  "detail-type": [
    "customerCreated"
  ]
}
```

10. Escolha Next (Próximo).
11. Em Target types (Tipos de destinos), escolha AWS service (Serviço da).
12. Em Selecionar um destino, escolha o tópico do SNS na lista suspensa.
13. Em Tópico escolha o nome do tópico do Amazon SNS criado na etapa 1. Para este tutorial, escolha eventbridge-IT-test.
14. Para Configurações de atualização, faça o seguinte:

    a. Em Configurar entrada de destino, escolha Transformador de entrada na lista suspensa.

    b. Escolha Configurar transformador de entrada

    c. em Eventos de amostra, insira o seguinte:

```
{
  "detail-type": "customerCreated"
}
```

    d. Para o Transformador de entrada de destino, faça o seguinte:

        i. Em Caminho de entrada, insira o seguinte:

```
{"detail-type":"$.detail-type"}
```

ii. Em Modelo de entrada, insira o seguinte:

```
"This event was of <detail-type> type."
```

e. Escolha Confirmar.

15. Escolha Next (Próximo).
16. Escolha Next (Próximo).
17. Analise os detalhes da regra e escolha Create rule (Criar regra).

Etapa 4: enviar eventos de teste

Agora que o tópico e a regra do SNS foram configurados, serão enviados eventos de teste para garantir que a regra esteja funcionando corretamente.

Para enviar eventos de teste (console)

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, selecione Event buses (Barramentos de eventos).
3. No quadro Barramento de eventos padrão, escolha Ações, Enviar eventos.
4. Insira a origem de eventos. Por exemplo, `TestEvent`.
5. Em Tipo de detalhe, insira `customerCreated`.
6. Em Detalhes do evento, insira `{}`.
7. Selecione Send (Enviar).

Etapa 5: confirmar o êxito

Se receber um e-mail de notificações da AWS que corresponda à saída esperada, o tutorial terá sido concluído com êxito.

Etapa 6: Limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir os recursos da AWS que não estão mais sendo usados, são evitadas cobranças desnecessárias em sua conta da AWS.

Para excluir o tópico do SNS

1. Abra a página Tópicos do console do SNS.
2. Selecione o tópico que foi criado.
3. Escolha Delete (Excluir).
4. Digite **`delete me`**.
5. Escolha Delete (Excluir).

Para excluir a assinatura do SNS

1. Abra a página de Assinaturas no console do Amazon SNS.
2. Selecione a assinatura que foi criada.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Para excluir as regras do EventBridge

1. Abra a página Regras no console do EventBridge.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Tutoriais do Amazon EventBridge para a integração com outros serviços da AWS

O Amazon EventBridge trabalha com outros serviços da AWS para processar eventos ou invocar um recurso da AWS como destino de uma regra. Os seguintes tutoriais mostram como integrar o EventBridge aos serviços da AWS.

Tutoriais:

- Tutorial: registrar o estado do grupo do Auto Scaling usando o EventBridge
- Tutorial: registre chamadas de AWS API usando EventBridge
- Tutorial: registre o estado de uma instância do Amazon EC2 usando EventBridge
- Tutorial: registrar operações no nível do objeto do Amazon S3 usando o EventBridge
- Tutorial: Envie eventos para um stream do Amazon Kinesis usando EventBridge e o esquema aws.events
- Tutorial: programar snapshots automatizados do Amazon EBS usando o EventBridge
- Tutorial: enviar uma notificação quando um objeto do S3 é criado
- Tutorial: agendar funções do AWS Lambda usando o EventBridge

Tutorial: registrar o estado do grupo do Auto Scaling usando o EventBridge

É possível executar uma função do AWS Lambda que registre um evento sempre que um grupo do Auto Scaling iniciar ou terminar uma instância do Amazon EC2, além de registrar se o evento de inicialização ou término teve êxito.

Para obter informações sobre outros casos que usam eventos do Amazon EC2 Auto Scaling, consulte Use o EventBridge para lidar com eventos de ajuste de escala automático no Manual do usuário do Amazon EC2 Auto Scaling.

Neste tutorial, é criada uma função do Lambda e uma regra no console do EventBridge que chama essa função quando um grupo do Amazon EC2 Auto Scaling inicia ou encerra uma instância.

Etapas:

- Pré-requisitos
- Etapa 1: criar uma função do Lambda
- Etapa 2: Criar uma regra
- Etapa 3: Testar a regra
- Etapa 4: confirmar o êxito
- Etapa 5: limpar os recursos

Pré-requisitos

Para concluir este tutorial, serão necessários os seguintes recursos:

- Um grupo do Auto Scaling. Para obter mais informações, consulte Como criar um grupo do Auto Scaling usando uma configuração de execução no Guia do usuário do Amazon EC2 Auto Scaling.

Etapa 1: criar uma função do Lambda

Crie uma função do Lambda para registrar os eventos de aumento e redução da escala na horizontal para seu grupo do Auto Scaling.

Como criar uma função do Lambda

1. Abra o console do AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Escolha Create function (Criar função).

3. Escolha Author from scratch (Criar do zero).
4. Insira um nome para a função do Lambda. Por exemplo, atribua à função o nome `LogAutoScalingEvent`.
5. Deixe o resto das opções como padrão e escolha Criar função.
6. Na guia Código da página da função, clique duas vezes em index.js.
7. Substitua o código existente pelo código a seguir.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogAutoScalingEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Escolha Deploy (Implantar).

Etapa 2: Criar uma regra

Crie uma regra para executar a função do Lambda criada na etapa 1. A regra é executada quando seu grupo do Auto Scaling inicia ou interrompe uma instância.

Para criar uma regra do

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Insira um nome e uma descrição para a regra. Por exemplo, nomeie a regra como `TestRule`
5. Em Event bus (Barramento de eventos), escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, selecione default. Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Rule type (Tipo de regra), escolha Rule with an event pattern (Regra com um padrão de evento).
7. Escolha Next (Próximo).
8. Em Event source (Origem do evento), escolha AWS services (Serviços da).

9. Em Event pattern (Padrão de evento), faça o seguinte:

    a. Em Origem do evento, selecione ajuste de escala automático na lista suspensa.

    b. Em Tipo de evento, selecione Iniciar e encerrar instância na lista suspensa.

    c. Escolha Qualquer evento de instância e Qualquer nome de grupo.

10. Escolha Next (Próximo).
11. Em Target types (Tipos de destinos), escolha AWS service (Serviço da).
12. Em Selecionar um destino, escolha a função do Lambda na lista suspensa.
13. Em Função, selecione a função do Lambda criada na seção Etapa 1: criar uma função do Lambda. Neste exemplo, selecione `LogAutoScalingEvent`.
14. Escolha Next (Próximo).
15. Escolha Next (Próximo).
16. Analise os detalhes da regra e escolha Create rule (Criar regra).

Etapa 3: Testar a regra

Você pode testar a regra escalando manualmente um grupo do Auto Scaling para que ele inicie uma instância. Espere alguns minutos para que o evento que aumenta a escala horizontalmente ocorra, verifique se sua função do Lambda foi invocada.

Para testar a regra usando um grupo do Auto Scaling

1. Para aumentar o tamanho de seu grupo do Auto Scaling, faça o seguinte:

    a. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.

    b. No painel de navegação, escolha Auto Scaling, Auto Scaling Groups (Grupos de Auto Scaling).

    c. Marque a caixa de seleção para seu grupo do Auto Scaling.

    d. Na guia Details (Detalhes), escolha Edit (Editar). Em Desejada, aumente a capacidade desejada para um. Por exemplo, se o valor atual for 2, digite 3. A capacidade desejada deve ser menor ou igual ao tamanho máximo do grupo. Se o seu novo valor para Desejado for maior que Máx, você deve atualizar Máx. Ao terminar, escolha Save (Salvar).

2. Para visualizar a saída da função do Lambda, faça o seguinte:

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

b. No painel de navegação, selecione Logs.

c. Selecione o nome do grupo de logs para sua função do Lambda (`/aws/lambda/function-name`).

d. Selecione o nome do fluxo de logs para visualizar os dados fornecidos pela função para a instância que você iniciou.

3. (Opcional) Ao concluir, é possível diminuir a capacidade desejada em um nível para que o grupo do Auto Scaling volte ao tamanho anterior.

Etapa 4: confirmar o êxito

Se vir o evento Lambda nos registros do CloudWatch, este tutorial foi concluído com com êxito. Se o evento não estiver nos seus logs do CloudWatch, comece a solucionar problemas verificando se a regra foi criada com êxito e, se a regra parecer correta, verifique se o código da sua função do Lambda está correto.

Etapa 5: limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir os recursos da AWS que não estão mais sendo usados, são evitadas cobranças desnecessárias em sua conta da AWS.

Para excluir as regras do EventBridge

1. Abra a página Regras no console do EventBridge.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Para excluir as funções do Lambda

1. Abra a página Funções do console do Lambda.
2. Selecione as funções que foram criadas.
3. Escolha Actions, Delete.
4. Escolha Delete (Excluir).

Tutorial: registre chamadas de AWS API usando EventBridge

Você pode usar EventBridge as regras da Amazon para reagir às chamadas de API feitas por um AWS serviço que são registradas por AWS CloudTrail.

Neste tutorial, você cria uma AWS CloudTrailtrilha, uma função Lambda e uma regra no EventBridge console. A regra invoca a função do Lambda quando uma instância do Amazon EC2 é interrompida.

Etapas:

- Etapa 1: criar uma AWS CloudTrail trilha
- Etapa 2: criar uma função do AWS Lambda
- Etapa 3: criar uma regra
- Etapa 4: testar a regra
- Etapa 5: confirmar o êxito
- Etapa 6: Limpar os recursos

Etapa 1: criar uma AWS CloudTrail trilha

Se já tiver uma trilha configurada, avance para a etapa 5.

Para criar uma trilha

1. Abra o CloudTrail console em https://console.aws.amazon.com/cloudtrail/.
2. Selecione Trails (Trilhas), Create trail (Criar trilha).
3. Em Nome da trilha, digite um nome para a trilha.
4. Em Local de armazenamento, em Criar um novo bucket do S3.
5. Em alias do AWS KMS , digite um alias para a chave do KMS.
6. Escolha Próximo.
7. Escolha Próximo.
8. Escolha Create Trail (Criar trilha).

Etapa 2: criar uma função do AWS Lambda

Crie uma função do Lambda para registrar os eventos de chamada de API.

Criar uma função do Lambda

1. Abra o AWS Lambda console em https://console.aws.amazon.com/lambda/.
2. Escolha Create function (Criar função).
3. Escolha Author from scratch (Criar do zero).
4. Digite um nome e uma descrição para a função Lambda. Por exemplo, atribua à função o nome `LogEC2StopInstance`.
5. Deixe o resto das opções como padrão e escolha Criar função.
6. Na guia Código da página da função, clique duas vezes em index.js.
7. Substitua o código existente pelo código a seguir.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2StopInstance');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Escolha Implantar.

Etapa 3: criar uma regra

Crie uma regra para executar a função do Lambda criada na etapa 2 sempre que interromper uma instância do Amazon EC2.

Para criar uma regra do

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra. Por exemplo, nomeie a regra como `TestRule`
5. Em Barramento de Eventos, escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, selecione padrão. Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Tipo de Regra, escolha Regra com Padrão de Evento.

7. Selecione Next (Próximo).
8. Em Fonte do evento, selecione Serviços da AWS .
9. Em Event pattern (Padrão de evento), faça o seguinte:
    a. Em Origem do evento, selecione EC2 na lista suspensa.
    b. Para Tipo de evento, selecione AWS API Call via na CloudTrail lista suspensa.
    c. Escolha Operações específicas e insira `StopInstances`.
10. Selecione Next (Próximo).
11. Em Tipos de destino, escolha Serviço da AWS .
12. Em Selecionar um destino, escolha a função do Lambda na lista suspensa.
13. Em Função, selecione a função do Lambda criada na seção Etapa 1: criar uma função do Lambda. Neste exemplo, selecione `LogEC2StopInstance`.
14. Escolha Próximo.
15. Escolha Próximo.
16. Analise os detalhes da regra e selecione Criar regra.

Etapa 4: testar a regra

Você pode testar a regra ao interromper uma instância do Amazon EC2 usando o console do Amazon EC2. Aguarde alguns minutos até que a instância pare e, em seguida, verifique suas AWS Lambda métricas no CloudWatch console para verificar se sua função foi executada.

Como testar a regra ao parar uma instância

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. Execute uma instância. Para obter mais informações, consulte Launch Your Instance no Guia do usuário do Amazon EC2.
3. Pare a instância. Para obter mais informações, consulte Stop and Start Your Instance no Guia do usuário do Amazon EC2.
4. Para visualizar a saída da função do Lambda, faça o seguinte:
    a. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
    b. No painel de navegação, selecione Logs.
    c. Selecione o nome do grupo de logs para sua função do Lambda (`/aws/lambda/`*`function-name`*).

d. Selecione o nome do fluxo de logs para visualizar os dados fornecidos pela função para a instância que você interrompeu.

5. (Opcional) Ao terminar, encerre a instância interrompida. Para obter mais informações, consulte Encerre sua instância no Guia do usuário do Amazon EC2.

Etapa 5: confirmar o êxito

Se você ver o evento Lambda nos CloudWatch registros, concluiu com êxito este tutorial. Se o evento não estiver em seus CloudWatch registros, comece a solucionar problemas verificando se a regra foi criada com sucesso e, se a regra parecer correta, verifique se o código da sua função Lambda está correto.

Etapa 6: Limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir AWS recursos que você não está mais usando, você evita cobranças desnecessárias em sua AWS conta.

Para excluir a (s) EventBridge regra (s)

1. Abra a página Regras do EventBridge console.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Para excluir as funções do Lambda

1. Abra a página Funções do console do Lambda.
2. Selecione as funções que foram criadas.
3. Escolha Ações, Excluir.
4. Escolha Excluir.

Para excluir a (s) CloudTrail trilha (s)

1. Abra a página Trilhas do CloudTrail console.
2. Selecione as trilhas que foram criadas.

3. Escolha Delete (Excluir).
4. Escolha Excluir.

Tutorial: registre o estado de uma instância do Amazon EC2 usando EventBridge

Também é possível criar uma função do AWS Lambda que registra as alterações de estado para uma instância do Amazon EC2. Você pode optar por criar uma regra que execute a função do Lambda sempre que houver uma transição de estado ou uma transição para um ou mais estados de interesse. Neste tutorial, você registra a execução de qualquer nova instância.

Etapas:

- Etapa 1: Criar uma função do AWS Lambda
- Etapa 2: Criar uma regra
- Etapa 3: Testar a regra
- Etapa 4: confirmar o êxito
- Etapa 5: limpar os recursos

Etapa 1: Criar uma função do AWS Lambda

Crie uma função do Lambda para registrar em log os eventos de alteração de estado. Ao criar sua regra na etapa 2, essa função é especificada.

Criar uma função do Lambda

1. Abra o AWS Lambda console em https://console.aws.amazon.com/lambda/.
2. Escolha Create function (Criar função).
3. Escolha Author from scratch (Criar do zero).
4. Digite um nome e uma descrição para a função Lambda. Por exemplo, atribua à função o nome `LogEC2InstanceStateChange`.
5. Deixe o resto das opções como padrão e escolha Criar função.
6. Na guia Código da página da função, clique duas vezes em index.js.
7. Substitua o código existente pelo código a seguir.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2InstanceStateChange');
    console.log('Received event:', JSON.stringify(event, null, 2));
```

```
    callback(null, 'Finished');
};
```

8. Escolha Deploy (Implantar).

Etapa 2: Criar uma regra

Crie uma regra para executar a função do Lambda criada na etapa 1. A regra é executada quando você iniciará uma instância do Amazon EC2.

Para criar a EventBridge regra

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra. Por exemplo, nomeie a regra como `TestRule`
5. Em Barramento de Eventos, escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, selecione padrão. Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Tipo de Regra, escolha Regra com Padrão de Evento.
7. Selecione Next (Próximo).
8. Em Fonte do evento, selecione Serviços da AWS .
9. Em Event pattern (Padrão de evento), faça o seguinte:
    a. Em Origem do evento, selecione EC2 na lista suspensa.
    b. Em Tipo de evento, escolha Notificação de mudança de estado da instância do EC2 na lista suspensa.
    c. Escolha Estados específicos e escolha em execução na lista suspensa.
    d. Escolha Qualquer instância
10. Selecione Next (Próximo).
11. Em Tipos de destino, escolha Serviço da AWS .
12. Em Selecionar um destino, escolha a função do Lambda na lista suspensa.
13. Em Função, selecione a função do Lambda criada na seção Etapa 1: criar uma função do Lambda. Neste exemplo, selecione `LogEC2InstanceStateChange`.

14. Escolha Próximo.
15. Escolha Próximo.
16. Analise os detalhes da regra e selecione Criar regra.

Etapa 3: Testar a regra

Você pode testar a regra ao interromper uma instância do Amazon EC2 usando o console do Amazon EC2. Aguarde alguns minutos até que a instância pare e, em seguida, verifique suas AWS Lambda métricas no CloudWatch console para verificar se sua função foi executada.

Como testar a regra ao parar uma instância

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. Execute uma instância. Para obter mais informações, consulte Launch Your Instance no Guia do usuário do Amazon EC2.
3. Pare a instância. Para obter mais informações, consulte Stop and Start Your Instance no Guia do usuário do Amazon EC2.
4. Para visualizar a saída da função do Lambda, faça o seguinte:
    a. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
    b. No painel de navegação, selecione Logs.
    c. Selecione o nome do grupo de logs para sua função do Lambda (`/aws/lambda/function-name`).
    d. Selecione o nome do fluxo de logs para visualizar os dados fornecidos pela função para a instância que você interrompeu.
5. (Opcional) Ao terminar, encerre a instância interrompida. Para obter mais informações, consulte Encerre sua instância no Guia do usuário do Amazon EC2.

Etapa 4: confirmar o êxito

Se você ver o evento Lambda nos CloudWatch registros, concluiu com êxito este tutorial. Se o evento não estiver em seus CloudWatch registros, comece a solucionar problemas verificando se a regra foi criada com sucesso e, se a regra parecer correta, verifique se o código da sua função Lambda está correto.

Etapa 5: limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir AWS recursos que você não está mais usando, você evita cobranças desnecessárias em sua AWS conta.

Para excluir a (s) EventBridge regra (s)

1. Abra a página Regras do EventBridge console.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Para excluir as funções do Lambda

1. Abra a página Funções do console do Lambda.
2. Selecione as funções que foram criadas.
3. Escolha Ações, Excluir.
4. Escolha Excluir.

Tutorial: registrar operações no nível do objeto do Amazon S3 usando o EventBridge

É possível registrar as operações de API no nível do objeto em seus buckets do Amazon S3. Para que o Amazon EventBridge possa fazer a correspondência com esses eventos, é preciso usar o AWS CloudTrail para configurar uma trilha para receber esses eventos.

Neste tutorial, é criada uma uma trilha do CloudTrail, uma função do AWS Lambda e uma regra no console do EventBridge que invoca essa função em resposta a um evento de dados do S3.

Etapas:

- Etapa 1: Configurar a trilha do AWS CloudTrail
- Etapa 2: criar uma função do AWS Lambda
- Etapa 3: Criar uma regra
- Etapa 4: Testar a regra
- Etapa 5: confirmar o êxito
- Etapa 6: Limpar os recursos

Etapa 1: Configurar a trilha do AWS CloudTrail

Para registrar os eventos de dados de um bucket do S3 no AWS CloudTrail e no EventBridge, primeiro crie uma trilha. A trilha captura chamadas de API e eventos relacionados em sua conta e fornece os arquivos de log para um bucket do S3 especificado. É possível atualizar uma trilha existente ou criar uma.

Para obter mais informações, consulte Eventos de dados, no Guia do usuário do AWS CloudTrail.

Para criar uma trilha

1. Abra o console do CloudTrail em https://console.aws.amazon.com/cloudfront/.
2. Selecione Trails (Trilhas), Create trail (Criar trilha).
3. Em Nome da trilha, digite um nome para a trilha.
4. Em Local de armazenamento, em Criar um novo bucket do S3.
5. Em alias do AWS KMS, digite um alias para a chave do KMS.
6. Escolha Next (Próximo).
7. Em Tipo de evento, escolha Eventos de dados.

8. Em Eventos de dados, siga um destes procedimentos:

   - Para registrar eventos de dados de todos os objetos do Amazon S3 em um bucket, especifique um bucket do S3 e um prefixo vazio. Quando ocorre um evento em um objeto nesse bucket, a trilha processa e registra o evento.
   - Para registrar eventos de dados de objetos específicos do Amazon S3 em um bucket, especifique um bucket do S3 e o prefixo do objeto. Quando ocorre um evento em um objeto nesse bucket e o objeto começa com o prefixo especificado, a trilha processa e registra o evento.
9. Para cada recurso, especifique se deseja registrar em log os eventos de Leitura, Gravação ou ambos.
10. Escolha Next (Próximo).
11. Escolha Create Trail (Criar trilha).

Etapa 2: criar uma função do AWS Lambda

Crie uma função do Lambda para registrar os eventos de dados de seus buckets do S3.

Como criar uma função do Lambda

1. Abra o console do AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Escolha Create function (Criar função).
3. Escolha Author from scratch (Criar do zero).
4. Digite um nome e uma descrição para a função Lambda. Por exemplo, atribua à função o nome `LogS3DataEvents`.
5. Deixe o resto das opções como padrão e escolha Criar função.
6. Na guia Código da página da função, clique duas vezes em index.js.
7. Substitua o código existente pelo código a seguir.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogS3DataEvents');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Escolha Implantar.

Etapa 3: Criar uma regra

Crie uma regra para executar a função do Lambda criada na etapa 2. Esta regra é executada em resposta a um evento de dados do Amazon S3.

Para criar uma regra do

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Insira um nome e uma descrição para a regra. Por exemplo, nomeie a regra como `TestRule`
5. Em Event bus (Barramento de eventos), escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, selecione default. Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Rule type (Tipo de regra), escolha Rule with an event pattern (Regra com um padrão de evento).
7. Escolha Next (Próximo).
8. Em Event source (Origem do evento), escolha AWS services (Serviços da).
9. Em Event pattern (Padrão de evento), faça o seguinte:
    a. Em Origem do evento, selecione Simple Storage Service (S3) na lista suspensa.
    b. Em Tipo de evento, selecione Chamada de API no nível de objeto via CloudTrail na lista suspensa.
    c. Escolha Specific operation(s) (Operações específicas) e, em seguida, escolha PutObject.
    d. Por padrão, a regra corresponde a eventos de dados para todos os buckets na região. Para corresponder eventos de dados a buckets específicos, escolha Especificar bucket(s) pelo nome e, depois, especifique um ou mais buckets.
10. Escolha Next (Próximo).
11. Em Target types (Tipos de destinos), escolha AWS service (Serviço da).
12. Em Selecionar um destino, escolha a função do Lambda na lista suspensa.
13. Em Função, selecione a função do Lambda `LogS3DataEvents` criada na etapa 1.

14. Escolha Next (Próximo).
15. Escolha Next (Próximo).
16. Analise os detalhes da regra e escolha Create rule (Criar regra).

Etapa 4: Testar a regra

Para testar a regra, coloque um objeto no seu bucket do S3. Você pode verificar se a sua função do Lambda foi chamada.

Para visualizar os logs da sua função do Lambda

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Logs.
3. Selecione o nome do grupo de logs para sua função do Lambda (`/aws/lambda/`*`function-name`*).
4. Selecione o nome do fluxo de logs para visualizar os dados fornecidos pela função para a instância que você iniciou.

Também é possível conferir o conteúdo dos logs do CloudTrail no bucket do S3 especificado para a trilha. Para obter mais informações, consulte Obtenção e exibição dos seus arquivos de log do CloudTrail no Guia do usuário do AWS CloudTrail.

Etapa 5: confirmar o êxito

Se vir o evento Lambda nos registros do CloudWatch, este tutorial foi concluído com com êxito. Se o evento não estiver nos seus logs do CloudWatch, comece a solucionar problemas verificando se a regra foi criada com êxito e, se a regra parecer correta, verifique se o código da sua função do Lambda está correto.

Etapa 6: Limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir os recursos da AWS que não estão mais sendo usados, são evitadas cobranças desnecessárias em sua conta da AWS.

Para excluir as regras do EventBridge

1. Abra a página Regras no console do EventBridge.

2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Para excluir as funções do Lambda

1. Abra a página Funções do console do Lambda.
2. Selecione as funções que foram criadas.
3. Escolha Actions, Delete.
4. Escolha Delete (Excluir).

Para excluir as trilhas do CloudTrail

1. Abra a página Trails (Trilhas) do console do CloudTrail.
2. Selecione as trilhas que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Tutorial: Envie eventos para um stream do Amazon Kinesis usando EventBridge e o esquema **aws.events**

Você pode enviar eventos de chamada de AWS API EventBridge para um stream do Amazon Kinesis, criar aplicativos do Kinesis Data Streams e processar grandes quantidades de dados. Neste tutorial, você cria um stream do Kinesis e, em seguida, cria uma regra no EventBridge console que envia eventos para esse stream quando uma instância do Amazon EC2 é interrompida.

Etapas:

- Pré-requisitos
- Etapa 1: criar um fluxo do Amazon Kinesis
- Etapa 2: Criar uma regra
- Etapa 3: Testar a regra
- Etapa 4: verificar se o evento foi enviado
- Etapa 5: limpar os recursos

Pré-requisitos

Neste tutorial, use o seguinte:

- Use o AWS CLI para trabalhar com os streams do Kinesis.

  Para instalar o AWS CLI, consulte Instalando, atualizando e desinstalando a AWS CLI versão 2.

> **Note**
>
> Este tutorial usa AWS eventos e o registro de `aws.events` esquema incorporado. Você também pode criar uma EventBridge regra com base no esquema de seus eventos personalizados adicionando-os manualmente a um registro de esquema personalizado ou usando a descoberta de esquemas.
> Para obter mais informações sobre esquemas, consulte ???. Para obter mais informações sobre como criar uma regra usando outras opções de padrões de eventos, consulte ???

Etapa 1: criar um fluxo do Amazon Kinesis

Para criar um stream, em um prompt de comando, use o `create-stream` AWS CLI comando.

```
aws kinesis create-stream --stream-name test --shard-count 1
```

Quando o status do fluxo é `ACTIVE`, ele está pronto. Para verificar o status do fluxo, use o comando `describe-stream`.

```
aws kinesis describe-stream --stream-name test
```

Etapa 2: Criar uma regra

Crie uma regra para enviar eventos ao seu fluxo ao interromper uma instância do Amazon EC2.

Para criar uma regra do

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra. Por exemplo, nomeie a regra como `TestRule`
5. Em Barramento de eventos, selecione padrão.
6. Em Tipo de Regra, escolha Regra com Padrão de Evento.
7. Escolha Próximo.
8. Em Origem do evento, escolha AWS eventos ou eventos de EventBridge parceiros.
9. Em Método de criação, escolha Usar esquema.
10. Em Event pattern (Padrão de evento), faça o seguinte:

    a. Em Tipo de esquema, escolha Selecionar esquema no registro do esquema.

    b. Para Registro do esquema, escolha aws.events na lista suspensa.

    c. Para Esquema, escolha aws.ec2 @EC2 na lista InstanceStateChangeNotification suspensa.

    EventBridge exibe o esquema do evento em Modelos.

    EventBridge exibe um asterisco vermelho ao lado de todas as propriedades necessárias para o evento, não para o padrão do evento.

    d. Em Modelos, defina as seguintes propriedades do filtro de eventos:

i. Selecione + Editar ao lado da propriedade de estado.

   Deixe o Relacionamento vazio. Em Valor, insira `running`. Escolha Definir.

ii. Selecione + Editar ao lado da propriedade de origem.

   Deixe o Relacionamento vazio. Em Valor, insira `aws.ec2`. Escolha Definir.

iii. Selecione + Editar ao lado da propriedade do tipo de detalhe.

   Deixe o Relacionamento vazio. Em Valor, insira `EC2 Instance State-change Notification`. Escolha Definir.

e. Para ver o padrão de evento que você construiu, escolha Gerar padrão de evento em JSON

   EventBridge exibe o padrão do evento em JSON:

```
{
  "detail": {
    "state": ["running"]
  },
  "detail-type": ["EC2 Instance State-change Notification"],
  "source": ["aws.ec2"]
}
```

11. Selecione Next (Próximo).
12. Em Tipos de destino, escolha Serviço da AWS .
13. Em Selecionar um destino, escolha a o fluxo do Kinesis na lista suspensa.
14. Para Fluxo, selecione o fluxo do Kinesis que foi criado na seção Etapa 1: criar um fluxo do Amazon Kinesis. Neste exemplo, selecione `test`.
15. Em Perfil de execução, escolha Criar um novo perfil para este recurso específico.
16. Escolha Próximo.
17. Escolha Próximo.
18. Analise os detalhes da regra e selecione Criar regra.

Etapa 3: Testar a regra

Para testar a regra, interrompa uma instância do Amazon EC2. Aguarde alguns minutos até que a instância pare e, em seguida, verifique suas CloudWatch métricas para verificar se sua função foi executada.

Como testar a regra ao parar uma instância

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. Execute uma instância. Para obter mais informações, consulte Launch Your Instance no Guia do usuário do Amazon EC2.
3. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
4. No painel de navegação, escolha Regras.

   Escolha o nome da regra que você criou e escolha Metrics for the rule (Métricas para a regra).
5. (Opcional) Ao terminar, encerre a instância. Para obter mais informações, consulte Encerre sua instância no Guia do usuário do Amazon EC2.

Etapa 4: verificar se o evento foi enviado

Você pode usar o AWS CLI para obter o registro do stream para verificar se o evento foi enviado.

Como obter o registro

1. Para começar a ler do seu fluxo do Kinesis, use o comando em um prompt de comando `get-shard-iterator`.

   ```
   aws kinesis get-shard-iterator --shard-id shardId-000000000000 --shard-iterator-type TRIM_HORIZON --stream-name test
   ```

   O seguinte é um exemplo de saída.

   ```
   {
       "ShardIterator": "AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp+KEd9I6AJ9ZG4lNR1EMi+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd+efGN2aHFdkH1rJl4BL9Wyrk+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg="
   }
   ```
2. Para obter o registro, use o comando `get-records` a seguir. Use o iterador de fragmentos da saída na etapa anterior.

   ```
   aws kinesis get-records --shard-iterator AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp
   ```

```
+KEd9I6AJ9ZG4lNR1EMi+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd
+efGN2aHFdkH1rJl4BL9Wyrk+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg=
```

Se o comando for bem-sucedido, ele solicitará registros do seu fluxo para o estilhaço especificado. Você pode receber zero ou mais registros. Os registros retornados podem não representar todos os registros em seu fluxo. Se você não receber os dados esperados, continue a chamar `get-records`.

3. Os registros no Kinesis são codificados por Base64. Use um decodificador Base64 para decodificar os dados para que você possa verificar se é o evento que foi enviado ao fluxo no formato JSON.

Etapa 5: limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir AWS recursos que você não está mais usando, você evita cobranças desnecessárias em sua AWS conta.

Para excluir a (s) EventBridge regra (s)

1. Abra a página Regras do EventBridge console.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Para excluir fluxos do Kinesis

1. Abra a página Fluxo de dados do console do Kinesis.
2. Selecione os fluxos criados.
3. Escolha Ações, Excluir.
4. Insira excluir no campo e escolha Excluir.

Tutorial: programar snapshots automatizados do Amazon EBS usando o EventBridge

É possível executar as regras do EventBridge de acordo com uma programação. Neste tutorial, um snapshot automatizado de um volume do Amazon Elastic Block Store (Amazon EBS) é criado em uma programação. É possível escolher uma taxa fixa para criar um snapshot em intervalos de alguns minutos ou usar uma expressão cron para especificar que o snapshot é feito em uma hora específica do dia.

> **Important**
>
> Para criar regras com destinos integrados, é preciso usar o AWS Management Console.

Etapas:

- Etapa 1: criar a regra
- Etapa 2: testar a regra
- Etapa 3: confirmar o êxito
- Etapa 4: limpar os recursos

Etapa 1: criar a regra

Crie uma regra que tire snapshots em uma programação. Você pode usar uma expressão taxa ou cron para especificar a programação. Para obter mais informações, consulte Como criar uma regra do Amazon EventBridge que é executada de acordo com uma programação..

Para criar uma regra do

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Insira um nome e uma descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra regra na mesma região e no mesmo barramento de eventos.

5. Em Event bus (Barramento de eventos), escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, escolha Barramento de eventos padrão da AWS. Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Rule type (Tipo de regra), escolha Schedule (Programação).
7. Escolha Next (Próximo).
8. Em Padrão de programação, escolha Uma programação que é executada em uma taxa regular, como a cada 10 minutos, insira **5** e escolha Minutos na lista suspensa.
9. Escolha Next (Próximo).
10. Em Target types (Tipos de destinos), escolha AWS service (Serviço da).
11. Em Selecionar o destino, escolha Criar snapshot do EBS na lista suspensa.
12. Em ID do volume, insira o ID referente ao volume de destino do Amazon EBS.
13. Em Perfil de execução, escolha Criar um novo perfil para este recurso específico.
14. Escolha Next (Próximo).
15. Escolha Next (Próximo).
16. Analise os detalhes da regra e escolha Create rule (Criar regra).

Etapa 2: testar a regra

É possível verificar a regra exibindo o primeiro snapshot depois de feito.

Como testar sua regra

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação, selecione Elastic Block Store, Snapshots.
3. Verifique se o primeiro snapshot aparece na lista.

Etapa 3: confirmar o êxito

Se vir um snapshot na lista, este tutorial foi concluído com êxito. Se o snapshot não estiver na lista, inicie a solução de problemas verificando se a regra foi criada com êxito.

Etapa 4: limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir os recursos da AWS que não estão mais sendo usados, são evitadas cobranças desnecessárias em sua conta da AWS.

Para excluir as regras do EventBridge

1. Abra a página Regras no console do EventBridge.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Tutorial: enviar uma notificação quando um objeto do S3 é criado

É possível enviar notificações por e-mail quando objetos do Amazon Simple Storage Service (Amazon S3) são criados usando o Amazon EventBridge e o Amazon SNS. Neste tutorial, serão criados um tópico e uma assinatura do SNS. Em seguida, será criada uma regra no console do EventBridge que enviará eventos para esse tópico quando os eventos `Object Created` do Amazon S3 forem recebidos.

Etapas:

- Pré-requisitos
- Etapa 1: criar um tópico do Amazon SNS
- Etapa 2: criar uma assinatura do Amazon SNS
- Etapa 3: criar uma regra
- Etapa 4: testar a regra
- Etapa 5: limpar os recursos

Pré-requisitos

Para receber eventos do Amazon S3 no EventBridge, é preciso habilitar o EventBridge no console do Amazon S3. Este tutorial pressupõe que o EventBridge esteja habilitado. Para obter mais informações, consulte Como ativar o Amazon EventBridge no console do S3.

Etapa 1: criar um tópico do Amazon SNS

Crie um tópico para receber os eventos do EventBridge.

Para criar um tópico

1. Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No painel de navegação, escolha Topics (Tópicos).
3. Escolha Create topic.
4. Em Type (Tipo), escolha Standard (Padrão).
5. Insira **`eventbridge-test`** como o nome do tópico.
6. Escolha Create topic.

Etapa 2: criar uma assinatura do Amazon SNS

Crie uma assinatura para receber notificações por e-mail do Amazon S3 quando os eventos forem recebidos pelo tópico.

Criar uma assinatura

1. Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. No painel de navegação, escolha Subscriptions.
3. Selecione Create subscription.
4. Em ARN do tópico, escolha o tópico criado na etapa 1. Para este tutorial, escolha eventbridge-test.
5. Em Protocol (Protocolo), escolha Email.
6. Para Endpoint, insira seu endereço de e-mail.
7. Selecione Create subscription.
8. Confirme a assinatura escolhendo Confirmar assinatura no e-mail que você recebe das notificações AWS.

Etapa 3: criar uma regra

Crie uma regra para enviar eventos ao seu tópico quando um objeto do Amazon S3 é criado.

Para criar uma regra do

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Insira um nome e uma descrição para a regra. Por exemplo, nomeie a regra como `s3-test`
5. Em Barramento de eventos, selecione padrão.
6. Em Rule type (Tipo de regra), escolha Rule with an event pattern (Regra com um padrão de evento).
7. Escolha Next (Próximo).
8. Em Event source (Origem do evento), escolha Eventos da AWS ou eventos de parceiro do EventBridge.
9. Em Creation method, escolha Use pattern form.

10. Em Event pattern (Padrão de evento), faça o seguinte:
    a. Em Origem do evento, selecione serviços da AWS na lista suspensa.
    b. Para o serviço da AWS, selecione Simple Storage Service (S3) na lista suspensa.
    c. Para Tipo de evento, escolha Notificação de eventos do Amazon S3 na lista suspensa.
    d. Escolha Eventos específicos e escolha Objeto criado na lista suspensa.
    e. Escolha Adicionar bucket
11. Escolha Next (Próximo).
12. Em Target types (Tipos de destinos), escolha AWS service (Serviço da).
13. Em Selecionar um destino, escolha o tópico do SNS na lista suspensa.
14. Em Tópico, selecione o tópico do Amazon SNS criado na seção Etapa 1: criar um tópico do SNS. Neste exemplo, selecione `eventbridge-test`.
15. Escolha Next (Próximo).
16. Escolha Next (Próximo).
17. Analise os detalhes da regra e escolha Create rule (Criar regra).

Etapa 4: testar a regra

Para testar sua regra, crie um objeto do Amazon S3 fazendo o upload de um arquivo em um bucket compatível com o Eventbridge. Em seguida, aguarde alguns minutos e verifique se recebeu um e-mail de notificações da AWS.

Etapa 5: limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir os recursos da AWS que não estão mais sendo usados, são evitadas cobranças desnecessárias em sua conta da AWS.

Para excluir o tópico do SNS

1. Abra a página Tópicos do console do SNS.
2. Selecione o tópico que foi criado.
3. Escolha Delete (Excluir).
4. Digite **`delete me`**.
5. Escolha Delete (Excluir).

Para excluir a assinatura do SNS

1. Abra a página de Assinaturas no console do Amazon SNS.
2. Selecione a assinatura que foi criada.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Para excluir as regras do EventBridge

1. Abra a página Regras no console do EventBridge.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Tutorial: agendar funções do AWS Lambda usando o EventBridge

É possível configurar uma regra para executar uma função do AWS Lambda em uma programação. Este tutorial mostra como usar o AWS Management Console ou a AWS CLI para criar a regra. Se quiser usar a AWS CLI, mas ainda não a instalou, consulte Instalar, atualizar e desinstalar a AWS CLI versão 2.

Para programações, o EventBridge não fornece precisão no segundo nível em expressões de programação. A melhor resolução ao usar uma expressão cron é um minuto. Por conta da natureza distribuída do EventBridge e aos serviços de destino, pode haver um atraso de diversos segundos entre o momento em que a regra programada é acionada e o momento em que o serviço de destino executa o recurso de destino.

Etapas:

- Etapa 1: criar uma função do Lambda
- Etapa 2: Criar uma regra
- Etapa 3: verificar a regra
- Etapa 4: confirmar o êxito
- Etapa 5: limpar os recursos

Etapa 1: criar uma função do Lambda

Crie uma função do Lambda para registrar os eventos programados.

Como criar uma função do Lambda

1. Abra o console do AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Escolha Create function (Criar função).
3. Escolha Author from scratch (Criar do zero).
4. Digite um nome e uma descrição para a função Lambda. Por exemplo, atribua à função o nome `LogScheduledEvent`.
5. Deixe o resto das opções como padrão e escolha Criar função.
6. Na guia Código da página da função, clique duas vezes em index.js.
7. Substitua o código existente pelo código a seguir.

```
'use strict';
```

```
exports.handler = (event, context, callback) => {
    console.log('LogScheduledEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Escolha Deploy (Implantar).

Etapa 2: Criar uma regra

Crie uma regra para executar a função do Lambda criada na etapa 1 em uma programação.

É possível o console ou a AWS CLI para criar as instâncias. Para usar a AWS CLI, é preciso primeiro conceder à regra permissão para invocar a função do Lambda. Em seguida, você pode criar a regra e adicionar a função do Lambda como destino.

Para criar uma regra (console)

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Insira um nome e uma descrição para a regra.

   Uma regra não pode ter o mesmo nome que outra regra na mesma região e no mesmo barramento de eventos.
5. Em Event bus (Barramento de eventos), escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, escolha Barramento de eventos padrão da AWS. Quando um serviço da AWS em sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
6. Em Rule type (Tipo de regra), escolha Schedule (Programação).
7. Escolha Next (Próximo).
8. Em Padrão de programação, escolha Uma programação que é executada em uma taxa regular, como a cada 10 minutos, insira **5** e escolha Minutos na lista suspensa.
9. Escolha Next (Próximo).
10. Em Target types (Tipos de destinos), escolha AWS service (Serviço da).

11. Em Selecionar um destino, escolha a função do Lambda na lista suspensa.
12. Em Função, selecione a função do Lambda criada na seção Etapa 1: criar uma função do Lambda. Neste exemplo, selecione `LogScheduledEvent`.
13. Escolha Next (Próximo).
14. Escolha Next (Próximo).
15. Analise os detalhes da regra e escolha Create rule (Criar regra).

Para criar uma regra (AWS CLI)

1. Para criar uma regra que seja executada em um cronograma, use o comando `put-rule`.

```
aws events put-rule \
--name my-scheduled-rule \
--schedule-expression 'rate(5 minutes)'
```

Quando essa regra é executada, ela cria um evento e o envia aos destinos. O comando a seguir é um exemplo de evento.

```
{
    "version": "0",
    "id": "53dc4d37-cffa-4f76-80c9-8b7d4a4d2eaa",
    "detail-type": "Scheduled Event",
    "source": "aws.events",
    "account": "123456789012",
    "time": "2015-10-08T16:53:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule"
    ],
    "detail": {}
}
```

2. Para conceder permissão à entidade principal do serviço do EventBridge (`events.amazonaws.com`) para executar a regra, use o comando `add-permission`.

```
aws lambda add-permission \
--function-name LogScheduledEvent \
--statement-id my-scheduled-event \
--action 'lambda:InvokeFunction' \
```

```
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule
```

3. Crie o arquivo `targets.json` com o seguinte conteúdo.

```
[
  {
    "Id": "1",
    "Arn": "arn:aws:lambda:us-east-1:123456789012:function:LogScheduledEvent"
  }
]
```

4. Para adicionar a função do Lambda que você criou na etapa 1 à regra, use o comando `put-targets`.

```
aws events put-targets --rule my-scheduled-rule --targets file://targets.json
```

Etapa 3: verificar a regra

Espere ao menos cinco minutos após concluir a etapa 2, é possível verificar se a sua função do Lambda foi invocada.

Visualize a saída da sua função do Lambda

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Logs.
3. Selecione o nome do grupo de logs para sua função do Lambda (`/aws/lambda/function-name`).
4. Selecione o nome do fluxo de logs para visualizar os dados fornecidos pela função para a instância que você iniciou.

Etapa 4: confirmar o êxito

Se vir o evento Lambda nos registros do CloudWatch, este tutorial foi concluído com com êxito. Se o evento não estiver nos seus logs do CloudWatch, comece a solucionar problemas verificando se a regra foi criada com êxito e, se a regra parecer correta, verifique se o código da sua função do Lambda está correto.

Etapa 5: limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir os recursos da AWS que não estão mais sendo usados, são evitadas cobranças desnecessárias em sua conta da AWS.

Para excluir as regras do EventBridge

1. Abra a página Regras no console do EventBridge.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Para excluir as funções do Lambda

1. Abra a página Funções do console do Lambda.
2. Selecione as funções que foram criadas.
3. Escolha Actions, Delete.
4. Escolha Delete (Excluir).

Tutoriais do Amazon EventBridge para integração com provedores de SaaS

O EventBridge pode trabalhar diretamente com aplicações e serviços de parceiros de SaaS para enviar e receber eventos. Os seguintes tutoriais mostram como integrar o EventBridge aos parceiros de SaaS.

Tutoriais:

- Tutorial: criar uma conexão com o Datadog como destino de API
- Tutorial: criar uma conexão com o Salesforce como destino de API
- Tutorial: criar uma conexão com o Zendesk como destino de API

Tutorial: criar uma conexão com o Datadog como destino de API

É possível usar o EventBridge para encaminhar eventos para serviços de terceiros, como Datadog.

Neste tutorial, será usado o console do EventBridge para criar uma conexão aoDatadog, um Destino de API que aponta para o Datadog e uma regra que roteia eventos para o Datadog.

Etapas:

- Pré-requisitos
- Etapa 1: criar uma conexão
- Etapa 2: criar um destino de API
- Etapa 3: criar regra
- Etapa 4: testar a regra
- Etapa 5: limpar os recursos

Pré-requisitos

Para concluir este tutorial, serão necessários os seguintes recursos:

- Uma conta do Datadog.
- Uma chave de API do Datadog.
- Um bucket do Amazon Simple Storage Service (Amazon S3) habilitado pelo EventBridge.

Etapa 1: criar uma conexão

Para enviar eventos para o Datadog, primeiro é preciso estabelecer uma conexão com a API do Datadog.

Para criar a conexão

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Destinos da API.
3. Escolha a guia Conexões e Criar conexão.
4. Insira um nome e uma descrição para a conexão. Por exemplo, insira **`Datadog`** como um nome e **`Datadog API Connection`** como uma descrição.
5. Em Tipo de autorização, escolha Chave de API.

6. Em Nome da API, insira **`DD-API-KEY`**.
7. Em Valor, cole sua chave secreta de API do Datadog.
8. Escolha Create (Criar).

Etapa 2: criar um destino de API

Agora que criou a conexão, criará o destino da API para usar como destino da regra.

Para criar o destino de API

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Destinos da API.
3. Escolha Criar destino de API.
4. Insira um nome e uma descrição para o destino de API. Por exemplo, insira **`DatadogAD`** para o nome e **`Datadog API Destination`** para a descrição.
5. Em Endpoint de destino da API, insira **`https://http-intake.logs.datadoghq.com/api/v2/logs`**.
6. Em Método HTTP, escolha POST.
7. Em Limite de taxa de invocação, insira **`300`**.
8. Em Conexão, escolha Usar uma conexão existente e escolha a conexão `Datadog` criada na etapa 1.
9. Escolha Create (Criar).

Etapa 3: criar regra

Em seguida, será criada uma regra para enviar eventos para o Datadog quando um objeto do Amazon S3 é criado.

Para criar uma regra do

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Insira um nome e uma descrição para a regra. Por exemplo, insira **`DatadogRule`** para o nome e **`Rule to send events to Datadog for S3 object creation`** para a descrição.

5. Em Event Bus (Barramento de eventos), escolha default (padrão).
6. Em Rule type (Tipo de regra), escolha Rule with an event pattern (Regra com um padrão de evento).
7. Escolha Next (Próximo).
8. Em Event source (Origem do evento), escolha Other (Outra).
9. Em Padrão de evento, insira um do seguintes:

```
{
  "source": ["aws.s3"]
}
```

10. Escolha Next (Próximo).
11. Em Tipos de destino, escolha o Destino da API do EventBridge.
12. Em Destino da API, escolha Usar um destino de API existente e escolha o destino do `DatadogAD` criado na etapa 2.
13. Em Perfil de execução, escolha Criar um novo perfil para este recurso específico.
14. Para Configurações de atualização, faça o seguinte:
    a. Em Configurar entrada de destino, escolha Transformador de entrada na lista suspensa.
    b. Escolha Configurar transformador de entrada
    c. em Eventos de amostra, insira o seguinte:

```
{
  "detail":[]
}
```

    d. Para o Transformador de entrada de destino, faça o seguinte:
        i. Em Caminho de entrada, insira o seguinte:

```
{"detail":"$.detail"}
```

        ii. Em Modelo de entrada, insira o seguinte:

```
{"message": <detail>}
```

    e. Escolha Confirmar.
15. Escolha Next (Próximo).

16. Escolha Next (Próximo).
17. Analise os detalhes da regra e escolha Create rule (Criar regra).

Etapa 4: testar a regra

Para testar sua regra, crie um objeto do Amazon S3 fazendo o upload de um arquivo em um bucket compatível com o Eventbridge. O objeto criado será registrado em log no console de logs do Datadog.

Etapa 5: limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir os recursos da AWS que não estão mais sendo usados, são evitadas cobranças desnecessárias em sua conta da AWS.

Para excluir as conexões do EventBridge

1. Abra a página de Destino da API do console do EventBridge.
2. Escolha a guia Connections (Conexões).
3. Selecione as Conexões que foram criadas.
4. Escolha Delete (Excluir).
5. Insira o nome da conexão e escolha Excluir.

Para excluir os destinos da API do EventBridge

1. Abra a página de Destino da API do console do EventBridge.
2. Selecione os destinos da API que foram criados.
3. Escolha Delete (Excluir).
4. Insira o nome do destino da API e escolha Excluir.

Para excluir as regras do EventBridge

1. Abra a página Regras no console do EventBridge.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).

4. Escolha Delete (Excluir).

Tutorial: criar uma conexão com o Salesforce como destino de API

Você pode usar EventBridge para rotear eventos para serviços de terceiros, como Salesforce.

Neste tutorial, você usará o EventBridge console para criar uma conexãoSalesforce, um destino de API que aponta e uma regra para a qual rotear eventosSalesforce. Salesforce

Etapas:

- Pré-requisitos
- Etapa 1: criar uma conexão
- Etapa 2: criar um destino de API
- Etapa 3: criar regra
- Etapa 4: testar a regra
- Etapa 5: limpar os recursos

Pré-requisitos

Para concluir este tutorial, serão necessários os seguintes recursos:

- Uma conta do Salesforce.
- Uma aplicação Salesforce conectada.
- Um token de segurança do Salesforce.
- Um evento de plataforma personalizado do Salesforce.
- Um bucket EventBridge habilitado do Amazon Simple Storage Service (Amazon S3).

Etapa 1: criar uma conexão

Para enviar eventos para o Salesforce, primeiro é preciso estabelecer uma conexão com a API do Salesforce.

Para criar a conexão

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Destinos da API.
3. Escolha a guia Conexões e Criar conexão.

4. Insira um nome e uma descrição para a conexão. Por exemplo, insira **`Salesforce`** como um nome e **`Salesforce API Connection`** como uma descrição.
5. Em Tipo de destino, escolha Parceiros e, em Destinos de parceiros, selecione o Salesforce na lista suspensa.
6. Em Endpoint de autorização, insira um destes:

   - Se estiver usando uma organização de produção, insira **`https://`*`MyDomainName`*`.my.salesforce.com./services/oauth2/token`**
   - Se estiver usando uma sandbox sem domínios aprimorados, digite **`https://`*`MyDomainName`*`--`*`SandboxName`*`.my. salesforce.com/services /oauth2/token`**
   - Se estiver usando uma sandbox com domínios aprimorados, digite **`https://`*`MyDomainName`*`-- `*`SandboxName`*`.sandbox.my.salesforce.com/services/oauth2/token`**
7. Em Método HTTP , escolha POST na lista suspensa.
8. Em ID do cliente, insira o ID do cliente da sua aplicação do Salesforce conectada.
9. Em Segredo do cliente, insira o segredo do cliente da sua aplicação do Salesforce conectada.
10. Para Parâmetros Http do OAuth, insira o seguinte par de chave/valor:

Chave	Valor
grant_type	client_credentials

11. Escolha Criar.

Etapa 2: criar um destino de API

Agora que criou a conexão, criará o destino da API para usar como destino da regra.

Para criar o destino de API

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Destinos da API.
3. Escolha Criar destino de API.
4. Insira um nome e uma descrição para o destino de API. Por exemplo, insira **`SalesforceAD`** para o nome e **`Salesforce API Destination`** para a descrição.

5. Em Endpoint de destino da API, insira **`https://MyDomainName.my.salesforce.com/services/data/v54.0/sobjects/MyEvent__e`** onde MyEvent__e é o evento da plataforma para o qual deseja enviar informações.
6. Em Método HTTP , escolha POST na lista suspensa.
7. Em Limite de taxa de invocação, insira **`300`**.
8. Em Conexão, escolha Usar uma conexão existente e escolha a conexão `Salesforce` criada na etapa 1.
9. Escolha Criar.

Etapa 3: criar regra

Em seguida, será criada uma regra para enviar eventos para o Salesforce quando um objeto do Amazon S3 é criado.

Para criar uma regra do

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Regras.
3. Selecione Criar regra.
4. Insira um nome e uma descrição para a regra. Por exemplo, insira **`SalesforceRule`** para o nome e **`Rule to send events to Salesforce for S3 object creation`** para a descrição.
5. Em Event Bus (Barramento de eventos), escolha default (padrão).
6. Em Rule type (Tipo de regra), selecione Rule with an event pattern (Regra com um padrão de evento).
7. Escolha Próximo.
8. Em Fonte do evento, escolha Outra.
9. Em Padrão de evento, insira um do seguintes:

```
{
  "source": ["aws.s3"]
}
```

10. Escolha Próximo.
11. Em Tipos de destino, escolha o destino EventBridge da API.

12. Em Destino da API, escolha Usar um destino de API existente e escolha o destino do `SalesforceAD` criado na etapa 2.
13. Em Perfil de execução, escolha Criar um novo perfil para este recurso específico.
14. Para Configurações de atualização, faça o seguinte:
    a. Em Configurar entrada de destino, escolha Transformador de entrada na lista suspensa.
    b. Escolha Configurar transformador de entrada
    c. em Eventos de amostra, insira o seguinte:

```
{
  "detail":[]
}
```

    d. Para o Transformador de entrada de destino, faça o seguinte:
        i. Em Caminho de entrada, insira o seguinte:

```
{"detail":"$.detail"}
```

        ii. Em Modelo de entrada, insira o seguinte:

```
{"message": <detail>}
```

    e. Escolha Confirmar.
15. Escolha Próximo.
16. Escolha Próximo.
17. Analise os detalhes da regra e selecione Criar regra.

Etapa 4: testar a regra

Para testar sua regra, crie um objeto do Amazon S3 fazendo o upload de um arquivo em um bucket habilitado. EventBridge As informações sobre o objeto criado serão enviadas para o evento da plataforma Salesforce.

Etapa 5: limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir AWS recursos que você não está mais usando, você evita cobranças desnecessárias em sua AWS conta.

Para excluir as EventBridge conexões

1. Abra a página de destino da API do EventBridge console.
2. Escolha a guia Connections (Conexões).
3. Selecione as Conexões que foram criadas.
4. Escolha Excluir.
5. Insira o nome da conexão e escolha Excluir.

Para excluir o (s) destino (s) da EventBridge API

1. Abra a página de destino da API do EventBridge console.
2. Selecione os destinos da API que foram criados.
3. Escolha Excluir.
4. Insira o nome do destino da API e escolha Excluir.

Para excluir a (s) EventBridge regra (s)

1. Abra a página Regras do EventBridge console.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Excluir.

Tutorial: criar uma conexão com o Zendesk como destino de API

É possível usar o EventBridge para encaminhar eventos para serviços de terceiros, como Zendesk.

Neste tutorial, será usado o console do EventBridge para criar uma conexão aoZendesk, um Destino de API que aponta para o Zendesk e uma regra que roteia eventos para o Zendesk.

Etapas:

- Pré-requisitos
- Etapa 1: criar uma conexão
- Etapa 2: criar um destino de API
- Etapa 3: criar regra
- Etapa 4: testar a regra
- Etapa 5: limpar os recursos

Pré-requisitos

Para concluir este tutorial, serão necessários os seguintes recursos:

- Uma conta do Zendesk.
- Um bucket do Amazon Simple Storage Service (Amazon S3) habilitado pelo EventBridge.

Etapa 1: criar uma conexão

Para enviar eventos para o Zendesk, primeiro é preciso estabelecer uma conexão com a API do Zendesk.

Para criar a conexão

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Destinos da API.
3. Escolha a guia Conexões e Criar conexão.
4. Insira um nome e uma descrição para a conexão. Por exemplo, insira **`Zendesk`** para o nome e **`Connection to Zendesk API`** para a descrição.
5. Em Tipo de autorização, escolha Básico (nome de usuário/senha).
6. Em Nome de usuário, insira seu nome de usuário do Zendesk.

7. Em Senha, insira sua senha do Zendesk.
8. Escolha Create (Criar).

Etapa 2: criar um destino de API

Agora que criou a conexão, criará o destino da API para usar como destino da regra.

Para criar o destino de API

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Destinos da API.
3. Escolha Criar destino de API.
4. Insira um nome e uma descrição para o destino de API. Por exemplo, insira **`ZendeskAD`** para o nome e **`Zendesk API destination`** para a descrição.
5. Em Endpoint de destino da API, insira **`https://`*`your-subdomain`*`.zendesk.com/api/v2/tickets.json`**, onde *`your-subdomain`* é o subdomínio associado à sua conta Zendesk.
6. Em Método HTTP, escolha POST.
7. Em Limite de taxa de invocação, insira **`10`**.
8. Em Conexão, escolha Usar uma conexão existente e escolha a conexão `Zendesk` criada na etapa 1.
9. Escolha Create (Criar).

Etapa 3: criar regra

Em seguida, crie uma regra para enviar eventos para o Zendesk quando um objeto do Amazon S3 é criado.

Para criar uma regra do

1. Abra o console do Amazon EventBridge em https://console.aws.amazon.com/events/.
2. No painel de navegação, escolha Rules (Regras).
3. Escolha Create rule (Criar regra).
4. Insira um nome e uma descrição para a regra. Por exemplo, insira **`ZendeskRule`** para o nome e **`Rule to send events to Zendesk when S3 objects are created`** para a descrição.
5. Em Event Bus (Barramento de eventos), escolha default (padrão).

6. Em Rule type (Tipo de regra), escolha Rule with an event pattern (Regra com um padrão de evento).
7. Escolha Next (Próximo).
8. Em Event source (Origem do evento), escolha Other (Outra).
9. Em Padrão de evento, insira um do seguintes:

```
{
  "source": ["aws.s3"]
}
```

10. Escolha Next (Próximo).
11. Em Tipos de destino, escolha o Destino da API do EventBridge.
12. Em Destino da API, escolha Usar um destino de API existente e escolha o destino do `ZendeskAD` criado na etapa 2.
13. Em Perfil de execução, escolha Criar um novo perfil para este recurso específico.
14. Para Configurações de atualização, faça o seguinte:
    a. Em Configurar entrada de destino, escolha Transformador de entrada na lista suspensa.
    b. Escolha Configurar transformador de entrada
    c. em Eventos de amostra, insira o seguinte:

```
{
  "detail":[]
}
```

    d. Para o Transformador de entrada de destino, faça o seguinte:
        i. Em Caminho de entrada, insira o seguinte:

```
{"detail":"$.detail"}
```

        ii. Em Modelo de entrada, insira o seguinte:

```
{"message": <detail>}
```

    e. Escolha Confirmar.
15. Escolha Next (Próximo).
16. Escolha Next (Próximo).

17. Analise os detalhes da regra e escolha Create rule (Criar regra).

Etapa 4: testar a regra

Para testar sua regra, crie um objeto do Amazon S3 fazendo o upload de um arquivo em um bucket compatível com o Eventbridge. Quando o evento corresponder à regra, o EventBridge chamará a API Criar tíquete do Zendesk. O novo tíquete aparecerá no painel Zendesk.

Etapa 5: limpar os recursos

Agora você pode excluir os recursos criados para este tutorial, a menos que queira mantê-los. Ao excluir os recursos da AWS que não estão mais sendo usados, são evitadas cobranças desnecessárias em sua conta da AWS.

Para excluir as conexões do EventBridge

1. Abra a página de Destino da API do console do EventBridge.
2. Escolha a guia Connections (Conexões).
3. Selecione as Conexões que foram criadas.
4. Escolha Delete (Excluir).
5. Insira o nome da conexão e escolha Excluir.

Para excluir os destinos da API do EventBridge

1. Abra a página de Destino da API do console do EventBridge.
2. Selecione os destinos da API que foram criados.
3. Escolha Delete (Excluir).
4. Insira o nome do destino da API e escolha Excluir.

Para excluir as regras do EventBridge

1. Abra a página Regras no console do EventBridge.
2. Selecione as regras que foram criadas.
3. Escolha Delete (Excluir).
4. Escolha Delete (Excluir).

Usando EventBridge com um AWS SDK

AWS kits de desenvolvimento de software (SDKs) 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++	AWS SDK for C++ exemplos de código
AWS CLI	AWS CLI exemplos de código
AWS SDK for Go	AWS SDK for Go exemplos de código
AWS SDK for Java	AWS SDK for Java exemplos de código
AWS SDK for JavaScript	AWS SDK for JavaScript exemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin exemplos de código
AWS SDK for .NET	AWS SDK for .NET exemplos de código
AWS SDK for PHP	AWS SDK for PHP exemplos de código
AWS Tools for PowerShell	Ferramentas para exemplos PowerShell de código
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) exemplos de código
AWS SDK for Ruby	AWS SDK for Ruby exemplos de código
AWS SDK para Rust	AWS SDK para Rust exemplos de código
SDK da AWS para SAP ABAP	SDK da AWS para SAP ABAP exemplos de código
AWS SDK for Swift	AWS SDK for Swift exemplos de código

Para exemplos específicos de EventBridge, consulteExemplos de código para EventBridge usar 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 EventBridge usar AWS SDKs

Os exemplos de código a seguir mostram como usar EventBridge com um kit AWS de desenvolvimento de software (SDK).

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 AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Conceitos básicos

Olá EventBridge

Os exemplos de código a seguir mostram como começar a usar EventBridge.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
using Amazon.EventBridge;
using Amazon.EventBridge.Model;

namespace EventBridgeActions;
```

```
public static class HelloEventBridge
{
    static async Task Main(string[] args)
    {
        var eventBridgeClient = new AmazonEventBridgeClient();

        Console.WriteLine($"Hello Amazon EventBridge! Following are some of your
 EventBuses:");
        Console.WriteLine();

        // You can use await and any of the async methods to get a response.
        // Let's get the first five event buses.
        var response = await eventBridgeClient.ListEventBusesAsync(
            new ListEventBusesRequest()
            {
                Limit = 5
            });

        foreach (var eventBus in response.EventBuses)
        {
            Console.WriteLine($"\tEventBus: {eventBus.Name}");
            Console.WriteLine($"\tArn: {eventBus.Arn}");
            Console.WriteLine($"\tPolicy: {eventBus.Policy}");
            Console.WriteLine();
        }
    }
}
```

- Para obter detalhes da API, consulte ListEventBusesa Referência AWS SDK for .NET da API.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
/**
 * 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 HelloEventBridge {
    public static void main(String[] args) {
        Region region = Region.US_WEST_2;
        EventBridgeClient eventBrClient = EventBridgeClient.builder()
                .region(region)
                .build();

        listBuses(eventBrClient);
        eventBrClient.close();
    }

    public static void listBuses(EventBridgeClient eventBrClient) {
        try {
            ListEventBusesRequest busesRequest = ListEventBusesRequest.builder()
                    .limit(10)
                    .build();

            ListEventBusesResponse response =
 eventBrClient.listEventBuses(busesRequest);
            List<EventBus> buses = response.eventBuses();
            for (EventBus bus : buses) {
                System.out.println("The name of the event bus is: " +
 bus.name());
                System.out.println("The ARN of the event bus is: " + bus.arn());
            }

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte ListEventBusesa Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import aws.sdk.kotlin.services.eventbridge.EventBridgeClient
import aws.sdk.kotlin.services.eventbridge.model.ListEventBusesRequest
import aws.sdk.kotlin.services.eventbridge.model.ListEventBusesResponse

suspend fun main() {
    listBusesHello()
}

suspend fun listBusesHello() {
    val request = ListEventBusesRequest {
        limit = 10
    }

    EventBridgeClient { region = "us-west-2" }.use { eventBrClient ->
        val response: ListEventBusesResponse =
 eventBrClient.listEventBuses(request)
        response.eventBuses?.forEach { bus ->
            println("The name of the event bus is ${bus.name}")
            println("The ARN of the event bus is ${bus.arn}")
        }
    }
}
```

- Para obter detalhes da API, consulte a ListEventBusesreferência da API AWS SDK for Kotlin.

Exemplos de código

Ações para EventBridge usar AWS SDKs

Os exemplos de código a seguir demonstram como realizar EventBridge ações individuais com AWS SDKs. Esses trechos chamam a EventBridge API e são trechos de código de programas maiores que devem ser executados em contexto. Cada exemplo inclui um link para GitHub, onde você pode 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 Amazon EventBridge API Reference.

Exemplos

- Use EnableRule com um AWS SDK ou CLI
- Use ListRuleNamesByTarget com um AWS SDK ou CLI
- Use ListRules com um AWS SDK ou CLI
- Use ListTargetsByRule com um AWS SDK ou CLI
- Use PutEvents com um AWS SDK ou CLI
- Use PutRule com um AWS SDK ou CLI
- Use PutTargets com um AWS SDK ou CLI
- Use RemoveTargets com um AWS SDK ou CLI

Use **`DeleteRule`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `DeleteRule`.

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:

- Conceitos básicos de regras e destinos

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Excluir uma regra pelo nome.

```
    /// <summary>
    /// Delete an event rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the event rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteRuleByName(string ruleName)
    {
        var response = await _amazonEventBridge.DeleteRuleAsync(
```

```
            new DeleteRuleRequest()
            {
                Name = ruleName
            });

        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obter detalhes da API, consulte DeleteRulea Referência AWS SDK for .NET da API.

CLI

AWS CLI

Para excluir uma regra de CloudWatch eventos

Este exemplo exclui a regra chamada InstanceStateChanges EC2:

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- Para obter detalhes da API, consulte DeleteRulena Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
   public static void deleteRuleByName(EventBridgeClient eventBrClient, String
ruleName) {
       DeleteRuleRequest ruleRequest = DeleteRuleRequest.builder()
               .name(ruleName)
               .build();

       eventBrClient.deleteRule(ruleRequest);
```

```
        System.out.println("Successfully deleted the rule");
    }
```

- Para obter detalhes da API, consulte DeleteRulea Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun deleteRuleByName(ruleName: String?) {
    val ruleRequest = DeleteRuleRequest {
        name = ruleName
    }
    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.deleteRule(ruleRequest)
        println("Successfully deleted the rule")
    }
}
```

- Para obter detalhes da API, consulte a DeleteRulereferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DescribeRule`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `DescribeRule`.

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:

- Conceitos básicos de regras e destinos

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Obter o estado de uma regra usando a descrição da regra.

```
    /// <summary>
    /// Get the state for a rule by the rule name.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="eventBusName">The optional name of the event bus. If empty,
uses the default event bus.</param>
    /// <returns>The state of the rule.</returns>
    public async Task<RuleState> GetRuleStateByRuleName(string ruleName, string?
eventBusName = null)
    {
        var ruleResponse = await _amazonEventBridge.DescribeRuleAsync(
            new DescribeRuleRequest()
            {
                Name = ruleName,
                EventBusName = eventBusName
            });
        return ruleResponse.State;
    }
```

- Para obter detalhes da API, consulte DescribeRulea Referência AWS SDK for .NET da API.

CLI

AWS CLI

Para exibir informações sobre uma regra de CloudWatch Eventos

Este exemplo exibe informações sobre a regra chamada DailyLambdaFunction:

```
aws events describe-rule --name "DailyLambdaFunction"
```

- Para obter detalhes da API, consulte DescribeRulena Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
   public static void checkRule(EventBridgeClient eventBrClient, String
eventRuleName) {
       try {
           DescribeRuleRequest ruleRequest = DescribeRuleRequest.builder()
                   .name(eventRuleName)
                   .build();

           DescribeRuleResponse response =
eventBrClient.describeRule(ruleRequest);
           System.out.println("The state of the rule is " +
response.stateAsString());

       } catch (EventBridgeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Para obter detalhes da API, consulte DescribeRulea Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun checkRule(eventRuleName: String?) {
    val ruleRequest = DescribeRuleRequest {
        name = eventRuleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.describeRule(ruleRequest)
        println("The state of the rule is $response")
    }
}
```

- Para obter detalhes da API, consulte a DescribeRulereferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DisableRule`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `DisableRule`.

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:

- Conceitos básicos de regras e destinos

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Desabilitar uma regra pelo nome da regra.

```
/// <summary>
/// Disable a particular rule on an event bus.
/// </summary
/// <param name="ruleName">The name of the rule.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DisableRuleByName(string ruleName)
{
    var ruleResponse = await _amazonEventBridge.DisableRuleAsync(
        new DisableRuleRequest()
        {
            Name = ruleName
        });
    return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
}
```

- Para obter detalhes da API, consulte DisableRulea Referência AWS SDK for .NET da API.

CLI

AWS CLI

Para desativar uma regra de CloudWatch eventos

Este exemplo desativa a regra chamada DailyLambdaFunction. A regra não é excluída:

```
aws events disable-rule --name "DailyLambdaFunction"
```

- Para obter detalhes da API, consulte DisableRulena Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Desabilitar uma regra usando o nome da regra.

```
   public static void changeRuleState(EventBridgeClient eventBrClient, String
eventRuleName, Boolean isEnabled) {
       try {
           if (!isEnabled) {
               System.out.println("Disabling the rule: " + eventRuleName);
               DisableRuleRequest ruleRequest = DisableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();

               eventBrClient.disableRule(ruleRequest);
           } else {
               System.out.println("Enabling the rule: " + eventRuleName);
               EnableRuleRequest ruleRequest = EnableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();
               eventBrClient.enableRule(ruleRequest);
           }

       } catch (EventBridgeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Para obter detalhes da API, consulte DisableRulea Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun changeRuleState(eventRuleName: String, isEnabled: Boolean?) {
    if (!isEnabled!!) {
        println("Disabling the rule: $eventRuleName")
        val ruleRequest = DisableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.disableRule(ruleRequest)
        }
    } else {
        println("Enabling the rule: $eventRuleName")
        val ruleRequest = EnableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.enableRule(ruleRequest)
        }
    }
}
```

- Para obter detalhes da API, consulte a DisableRulereferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`EnableRule`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `EnableRule`.

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:

- Conceitos básicos de regras e destinos

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Habilitar uma regra pelo nome da regra.

```
/// <summary>
/// Enable a particular rule on an event bus.
/// </summary>
/// <param name="ruleName">The name of the rule.</param>
/// <returns>True if successful.</returns>
public async Task<bool> EnableRuleByName(string ruleName)
{
    var ruleResponse = await _amazonEventBridge.EnableRuleAsync(
        new EnableRuleRequest()
        {
            Name = ruleName
        });
    return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
}
```

- Para obter detalhes da API, consulte EnableRulea Referência AWS SDK for .NET da API.

CLI

AWS CLI

Para habilitar uma regra de CloudWatch Eventos

Este exemplo ativa a regra chamada DailyLambdaFunction, que havia sido desativada anteriormente:

```
aws events enable-rule --name "DailyLambdaFunction"
```

- Para obter detalhes da API, consulte EnableRulena Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Habilitar uma regra usando o nome da regra.

```
   public static void changeRuleState(EventBridgeClient eventBrClient, String
eventRuleName, Boolean isEnabled) {
       try {
           if (!isEnabled) {
               System.out.println("Disabling the rule: " + eventRuleName);
               DisableRuleRequest ruleRequest = DisableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();

               eventBrClient.disableRule(ruleRequest);
           } else {
               System.out.println("Enabling the rule: " + eventRuleName);
               EnableRuleRequest ruleRequest = EnableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();
               eventBrClient.enableRule(ruleRequest);
           }

       } catch (EventBridgeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
```

```
    }
```

- Para obter detalhes da API, consulte EnableRulea Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun changeRuleState(eventRuleName: String, isEnabled: Boolean?) {
    if (!isEnabled!!) {
        println("Disabling the rule: $eventRuleName")
        val ruleRequest = DisableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.disableRule(ruleRequest)
        }
    } else {
        println("Enabling the rule: $eventRuleName")
        val ruleRequest = EnableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.enableRule(ruleRequest)
        }
    }
}
```

- Para obter detalhes da API, consulte a EnableRulereferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`ListRuleNamesByTarget`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `ListRuleNamesByTarget`.

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:

- Conceitos básicos de regras e destinos

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Liste todos os nomes das regras usando o destino.

```
/// <summary>
/// List names of all rules matching a target.
/// </summary>
/// <param name="targetArn">The ARN of the target.</param>
/// <returns>The list of rule names.</returns>
public async Task<List<string>> ListAllRuleNamesByTarget(string targetArn)
{
    var results = new List<string>();
    var request = new ListRuleNamesByTargetRequest()
    {
        TargetArn = targetArn
    };
    ListRuleNamesByTargetResponse response;
    do
    {
```

```
            response = await
_amazonEventBridge.ListRuleNamesByTargetAsync(request);
            results.AddRange(response.RuleNames);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }
```

- Para obter detalhes da API, consulte ListRuleNamesByTargeta Referência AWS SDK for .NET da API.

CLI

AWS CLI

Como exibir todas as regras que têm um destino especificado

Este exemplo exibe todas as regras que têm a função Lambda chamada "MyFunctionName" como destino:

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-
east-1:123456789012:function:MyFunctionName"
```

- Para obter detalhes da API, consulte ListRuleNamesByTargetna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Listar todos os nomes das regras usando o destino.

```
   public static void listTargetRules(EventBridgeClient eventBrClient, String
topicArn) {
       ListRuleNamesByTargetRequest ruleNamesByTargetRequest =
ListRuleNamesByTargetRequest.builder()
               .targetArn(topicArn)
               .build();

       ListRuleNamesByTargetResponse response =
eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest);
       List<String> rules = response.ruleNames();
       for (String rule : rules) {
           System.out.println("The rule name is " + rule);
       }
   }
```

- Para obter detalhes da API, consulte ListRuleNamesByTargeta Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun listTargetRules(topicArnVal: String?) {
    val ruleNamesByTargetRequest = ListRuleNamesByTargetRequest {
        targetArn = topicArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response =
eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest)
        response.ruleNames?.forEach { rule ->
            println("The rule name is $rule")
        }
    }
```

```
}
```

- Para obter detalhes da API, consulte a ListRuleNamesByTargetreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`ListRules`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `ListRules`.

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:

- Conceitos básicos de regras e destinos

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Liste todas as regras para um barramento de eventos.

```
    /// <summary>
    /// List the rules on an event bus.
    /// </summary>
    /// <param name="eventBusArn">The optional ARN of the event bus. If empty,
uses the default event bus.</param>
    /// <returns>The list of rules.</returns>
    public async Task<List<Rule>> ListAllRulesForEventBus(string? eventBusArn =
null)
```

```
    {
        var results = new List<Rule>();
        var request = new ListRulesRequest()
        {
            EventBusName = eventBusArn
        };
        // Get all of the pages of rules.
        ListRulesResponse response;
        do
        {
            response = await _amazonEventBridge.ListRulesAsync(request);
            results.AddRange(response.Rules);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }
```

- Para obter detalhes da API, consulte ListRulesa Referência AWS SDK for .NET da API.

CLI

AWS CLI

Para exibir uma lista de todas as regras de CloudWatch eventos

Este exemplo exibe todas as regras de CloudWatch eventos na região:

```
aws events list-rules
```

Para exibir uma lista de regras de CloudWatch eventos começando com uma determinada string.

Este exemplo exibe todas as regras de CloudWatch eventos na região que têm um nome começando com “Diário”:

```
aws events list-rules --name-prefix "Daily"
```

- Para obter detalhes da API, consulte ListRulesna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Habilitar uma regra usando o nome da regra.

```
    public static void listRules(EventBridgeClient eventBrClient) {
        try {
            ListRulesRequest rulesRequest = ListRulesRequest.builder()
                    .eventBusName("default")
                    .limit(10)
                    .build();

            ListRulesResponse response = eventBrClient.listRules(rulesRequest);
            List<Rule> rules = response.rules();
            for (Rule rule : rules) {
                System.out.println("The rule name is : " + rule.name());
                System.out.println("The rule description is : " +
rule.description());
                System.out.println("The rule state is : " +
rule.stateAsString());
            }

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte ListRulesa Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun listRules() {
    val rulesRequest = ListRulesRequest {
        eventBusName = "default"
        limit = 10
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listRules(rulesRequest)
        response.rules?.forEach { rule ->
            println("The rule name is ${rule.name}")
            println("The rule ARN is ${rule.arn}")
        }
    }
}
```

- Para obter detalhes da API, consulte a ListRulesreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`ListTargetsByRule`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `ListTargetsByRule`.

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:

- Conceitos básicos de regras e destinos

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Listar todos os destinos para uma regra usando o nome da regra.

```
    /// <summary>
    /// List all of the targets matching a rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>The list of targets.</returns>
    public async Task<List<Target>> ListAllTargetsOnRule(string ruleName)
    {
        var results = new List<Target>();
        var request = new ListTargetsByRuleRequest()
        {
            Rule = ruleName
        };
        ListTargetsByRuleResponse response;
        do
        {
            response = await _amazonEventBridge.ListTargetsByRuleAsync(request);
            results.AddRange(response.Targets);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }
```

- Para obter detalhes da API, consulte ListTargetsByRulea Referência AWS SDK for .NET da API.

CLI

AWS CLI

Para exibir todos os alvos de uma regra de CloudWatch Eventos

Este exemplo exibe todos os alvos da regra chamada DailyLambdaFunction:

```
aws events list-targets-by-rule --rule  "DailyLambdaFunction"
```

- Para obter detalhes da API, consulte ListTargetsByRulena Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Liste todos os destinos de uma regra usando o nome da regra.

```
   public static void listTargets(EventBridgeClient eventBrClient, String
ruleName) {
       ListTargetsByRuleRequest ruleRequest = ListTargetsByRuleRequest.builder()
               .rule(ruleName)
               .build();

       ListTargetsByRuleResponse res =
eventBrClient.listTargetsByRule(ruleRequest);
       List<Target> targetsList = res.targets();
       for (Target target: targetsList) {
           System.out.println("Target ARN: "+target.arn());
       }
   }
```

- Para obter detalhes da API, consulte ListTargetsByRulea Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun listTargets(ruleName: String?) {
    val ruleRequest = ListTargetsByRuleRequest {
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(ruleRequest)
        response.targets?.forEach { target ->
            println("Target ARN: ${target.arn}")
        }
    }
}
```

- Para obter detalhes da API, consulte a ListTargetsByRulereferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`PutEvents`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `PutEvents`.

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:

- Criar e acionar uma regra
- Conceitos básicos de regras e destinos

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Envie um evento que corresponda a um padrão personalizado para uma regra.

```
    /// <summary>
    /// Add an event to the event bus that includes an email, message, and time.
    /// </summary>
    /// <param name="email">The email to use in the event detail of the custom
event.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutCustomEmailEvent(string email)
    {
        var eventDetail = new
        {
            UserEmail = email,
            Message = "This event was generated by example code.",
            UtcTime = DateTime.UtcNow.ToString("g")
        };
        var response = await _amazonEventBridge.PutEventsAsync(
            new PutEventsRequest()
            {
                Entries = new List<PutEventsRequestEntry>()
                {
                    new PutEventsRequestEntry()
                    {
                        Source = "ExampleSource",
                        Detail = JsonSerializer.Serialize(eventDetail),
                        DetailType = "ExampleType"
                    }
                }
```

```
            });

        return response.FailedEntryCount == 0;
    }
```

- Para obter detalhes da API, consulte PutEventsa Referência AWS SDK for .NET da API.

C++

SDK para C++

> **Note**
>
> Tem mais sobre 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/events/EventBridgeClient.h>
#include <aws/events/model/PutEventsRequest.h>
#include <aws/events/model/PutEventsResult.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Enviar o evento.

```
        Aws::CloudWatchEvents::EventBridgeClient cwe;

        Aws::CloudWatchEvents::Model::PutEventsRequestEntry event_entry;
        event_entry.SetDetail(MakeDetails(event_key, event_value));
        event_entry.SetDetailType("sampleSubmitted");
        event_entry.AddResources(resource_arn);
        event_entry.SetSource("aws-sdk-cpp-cloudwatch-example");

        Aws::CloudWatchEvents::Model::PutEventsRequest request;
        request.AddEntries(event_entry);
```

```
        auto outcome = cwe.PutEvents(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to post CloudWatch event: " <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully posted CloudWatch event" << std::endl;
        }
```

- Para obter detalhes da API, consulte PutEventsa Referência AWS SDK for C++ da API.

CLI

AWS CLI

Para enviar um evento personalizado para CloudWatch Eventos

Este exemplo envia um evento personalizado para CloudWatch Eventos. O evento está contido no arquivo putevents.json:

```
aws events put-events --entries file://putevents.json
```

Veja a seguir o conteúdo do arquivo putevent.json:

```
[
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
  },
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value3\", \"key2\": \"value4\" }",
    "Resources": [
      "resource1",
```

```
      "resource2"
    ],
    "DetailType": "myDetailType"
   }
]
```

- Para obter detalhes da API, consulte PutEventsna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
   public static void triggerCustomRule(EventBridgeClient eventBrClient, String
email) {
       String json = "{" +
               "\"UserEmail\": \"" + email + "\"," +
               "\"Message\": \"This event was generated by example code.\"," +
               "\"UtcTime\": \"Now.\"" +
               "}";

       PutEventsRequestEntry entry = PutEventsRequestEntry.builder()
               .source("ExampleSource")
               .detail(json)
               .detailType("ExampleType")
               .build();

       PutEventsRequest eventsRequest = PutEventsRequest.builder()
               .entries(entry)
               .build();

       eventBrClient.putEvents(eventsRequest);
   }
```

- Para obter detalhes da API, consulte PutEventsa Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre 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.

```
import {
  EventBridgeClient,
  PutEventsCommand,
} from "@aws-sdk/client-eventbridge";

export const putEvents = async (
  source = "eventbridge.integration.test",
  detailType = "greeting",
  resources = [],
) => {
  const client = new EventBridgeClient({});

  const response = await client.send(
    new PutEventsCommand({
      Entries: [
        {
          Detail: JSON.stringify({ greeting: "Hello there." }),
          DetailType: detailType,
          Resources: resources,
          Source: source,
        },
      ],
    }),
  );

  console.log("PutEvents response:");
  console.log(response);
  // PutEvents response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
```

```
  //     requestId: '3d0df73d-dcea-4a23-ae0d-f5556a3ac109',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   Entries: [ { EventId: '51620841-5af4-6402-d9bc-b77734991eb5' } ],
  //   FailedEntryCount: 0
  // }

  return response;
};
```

- Para obter detalhes da API, consulte PutEventsa Referência AWS SDK for JavaScript da API.

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatchEvents service object
var ebevents = new AWS.EventBridge({ apiVersion: "2015-10-07" });

var params = {
  Entries: [
    {
      Detail: '{ "key1": "value1", "key2": "value2" }',
      DetailType: "appRequestSubmitted",
      Resources: ["RESOURCE_ARN"],
      Source: "com.company.app",
    },
  ],
};
```

```
ebevents.putEvents(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.Entries);
  }
});
```

- Para obter detalhes da API, consulte PutEventsa Referência AWS SDK for JavaScript da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun triggerCustomRule(email: String) {
    val json = "{" +
        "\"UserEmail\": \"" + email + "\"," +
        "\"Message\": \"This event was generated by example code.\"" +
        "\"UtcTime\": \"Now.\"" +
        "}"

    val entry = PutEventsRequestEntry {
        source = "ExampleSource"
        detail = json
        detailType = "ExampleType"
    }

    val eventsRequest = PutEventsRequest {
        this.entries = listOf(entry)
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
```

```
        eventBrClient.putEvents(eventsRequest)
    }
}
```

- Para obter detalhes da API, consulte a PutEventsreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`PutRule`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `PutRule`.

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:

- Criar e acionar uma regra
- Conceitos básicos de regras e destinos

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Crie uma regra que seja acionada quando um objeto é adicionado a um bucket do Amazon Simple Storage Service.

```
    /// <summary>
    /// Create a new event rule that triggers when an Amazon S3 object is created
in a bucket.
    /// </summary>
    /// <param name="roleArn">The ARN of the role.</param>
```

```
    /// <param name="ruleName">The name to give the rule.</param>
    /// <param name="bucketName">The name of the bucket to trigger the event.</
param>
    /// <returns>The ARN of the new rule.</returns>
    public async Task<string> PutS3UploadRule(string roleArn, string ruleName,
 string bucketName)
    {
        string eventPattern = "{" +
                                "\"source\": [\"aws.s3\"]," +
                                    "\"detail-type\": [\"Object Created\"]," +
                                    "\"detail\": {" +
                                        "\"bucket\": {" +
                                            "\"name\": [\"" + bucketName + "\"]"
 +
                                        "}" +
                                    "}" +
                                "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Example S3 upload rule for EventBridge",
                RoleArn = roleArn,
                EventPattern = eventPattern
            });

        return response.RuleArn;
    }
```

Crie uma regra que utilize um padrão personalizado.

```
    /// <summary>
    /// Update a rule to use a custom defined event pattern.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <returns>The ARN of the updated rule.</returns>
    public async Task<string> UpdateCustomEventPattern(string ruleName)
    {
        string customEventsPattern = "{" +
                                     "\"source\": [\"ExampleSource\"]," +
                                     "\"detail-type\": [\"ExampleType\"]" +
```

```
                                    "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Custom test rule",
                EventPattern = customEventsPattern
            });

        return response.RuleArn;
    }
```

- Para obter detalhes da API, consulte PutRulea Referência AWS SDK for .NET da API.

C++

SDK para C++

> **Note**
>
> Tem mais sobre 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/events/EventBridgeClient.h>
#include <aws/events/model/PutRuleRequest.h>
#include <aws/events/model/PutRuleResult.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Crie a regra.

```
        Aws::CloudWatchEvents::EventBridgeClient cwe;
        Aws::CloudWatchEvents::Model::PutRuleRequest request;
        request.SetName(rule_name);
```

```
        request.SetRoleArn(role_arn);
        request.SetScheduleExpression("rate(5 minutes)");
        request.SetState(Aws::CloudWatchEvents::Model::RuleState::ENABLED);

        auto outcome = cwe.PutRule(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch events rule " <<
                rule_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch events rule " <<
                rule_name << " with resulting Arn " <<
                outcome.GetResult().GetRuleArn() << std::endl;
        }
```

- Para obter detalhes da API, consulte PutRulea Referência AWS SDK for C++ da API.

CLI

AWS CLI

Para criar regras de CloudWatch eventos

Este exemplo cria uma regra que é acionada todos os dias, às 9h UTC. Se você usar put-targets para adicionar uma função do Lambda como destino dessa regra, poderá executar a função do Lambda todos os dias no horário especificado:

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9
 * * ? *)"
```

Este exemplo cria uma regra que é acionada quando qualquer instância do EC2 na região muda de estado:

```
aws events put-rule --name "EC2InstanceStateChanges" --event-pattern "{\"source
\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance State-change Notification\"]}"
 --role-arn "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

Este exemplo cria uma regra que é acionada quando qualquer instância do EC2 na região é interrompida ou encerrada:

```
aws events put-rule --name "EC2InstanceStateChangeStopOrTerminate" --event-
pattern "{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance State-change
 Notification\"],\"detail\":{\"state\":[\"stopped\",\"terminated\"]}}" --role-arn
 "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

- Para obter detalhes da API, consulte PutRulena Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Criar uma regra agendada

```
   public static void createEBRule(EventBridgeClient eventBrClient, String
ruleName, String cronExpression) {
       try {
           PutRuleRequest ruleRequest = PutRuleRequest.builder()
                   .name(ruleName)
                   .eventBusName("default")
                   .scheduleExpression(cronExpression)
                   .state("ENABLED")
                   .description("A test rule that runs on a schedule created by
the Java API")
                   .build();

           PutRuleResponse ruleResponse = eventBrClient.putRule(ruleRequest);
           System.out.println("The ARN of the new rule is " +
ruleResponse.ruleArn());

       } catch (EventBridgeException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
```

```
        }
    }
```

Crie uma regra que seja acionada quando um objeto é adicionado a um bucket do Amazon Simple Storage Service.

```
    // Create a new event rule that triggers when an Amazon S3 object is created
in
    // a bucket.
    public static void addEventRule(EventBridgeClient eventBrClient, String
roleArn, String bucketName,
            String eventRuleName) {
        String pattern = "{\n" +
                "  \"source\": [\"aws.s3\"],\n" +
                "  \"detail-type\": [\"Object Created\"],\n" +
                "  \"detail\": {\n" +
                "    \"bucket\": {\n" +
                "      \"name\": [\"" + bucketName + "\"]\n" +
                "    }\n" +
                "  }\n" +
                "}";

        try {
            PutRuleRequest ruleRequest = PutRuleRequest.builder()
                    .description("Created by using the AWS SDK for Java v2")
                    .name(eventRuleName)
                    .eventPattern(pattern)
                    .roleArn(roleArn)
                    .build();

            PutRuleResponse ruleResponse = eventBrClient.putRule(ruleRequest);
            System.out.println("The ARN of the new rule is " +
ruleResponse.ruleArn());

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte PutRulea Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre 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.

```
import { EventBridgeClient, PutRuleCommand } from "@aws-sdk/client-eventbridge";

export const putRule = async (
  ruleName = "some-rule",
  source = "some-source",
) => {
  const client = new EventBridgeClient({});

  const response = await client.send(
    new PutRuleCommand({
      Name: ruleName,
      EventPattern: JSON.stringify({ source: [source] }),
      State: "ENABLED",
      EventBusName: "default",
    }),
  );

  console.log("PutRule response:");
  console.log(response);
  // PutRule response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: 'd7292ced-1544-421b-842f-596326bc7072',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   RuleArn: 'arn:aws:events:us-east-1:xxxxxxxxxxxx:rule/
EventBridgeTestRule-1696280037720'
```

```
  // }
  return response;
};
```

- Para obter detalhes da API, consulte PutRulea Referência AWS SDK for JavaScript da API.

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatchEvents service object
var ebevents = new AWS.EventBridge({ apiVersion: "2015-10-07" });

var params = {
  Name: "DEMO_EVENT",
  RoleArn: "IAM_ROLE_ARN",
  ScheduleExpression: "rate(5 minutes)",
  State: "ENABLED",
};

ebevents.putRule(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.RuleArn);
  }
});
```

- Para obter detalhes da API, consulte PutRulea Referência AWS SDK for JavaScript da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Criar uma regra agendada

```
suspend fun createScRule(ruleName: String?, cronExpression: String?) {
    val ruleRequest = PutRuleRequest {
        name = ruleName
        eventBusName = "default"
        scheduleExpression = cronExpression
        state = RuleState.Enabled
        description = "A test rule that runs on a schedule created by the Kotlin
 API"
    }

    EventBridgeClient { region = "us-west-2" }.use { eventBrClient ->
        val ruleResponse = eventBrClient.putRule(ruleRequest)
        println("The ARN of the new rule is ${ruleResponse.ruleArn}")
    }
}
```

Crie uma regra que seja acionada quando um objeto é adicionado a um bucket do Amazon Simple Storage Service.

```
// Create a new event rule that triggers when an Amazon S3 object is created in a
 bucket.
suspend fun addEventRule(roleArnVal: String?, bucketName: String, eventRuleName:
 String?) {
    val pattern = """{
        "source": ["aws.s3"],
        "detail-type": ["Object Created"],
        "detail": {
        "bucket": {
            "name": ["$bucketName"]
```

```
            }
        }
    }"""

    val ruleRequest = PutRuleRequest {
        description = "Created by using the AWS SDK for Kotlin"
        name = eventRuleName
        eventPattern = pattern
        roleArn = roleArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val ruleResponse = eventBrClient.putRule(ruleRequest)
        println("The ARN of the new rule is ${ruleResponse.ruleArn}")
    }
}
```

- Para obter detalhes da API, consulte a PutRulereferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`PutTargets`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `PutTargets`.

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:

- Conceitos básicos de regras e destinos

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Adicione um tópico do Amazon SNS como um destino para uma regra.

```
    /// <summary>
    /// Add an Amazon SNS target topic to a rule.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <param name="targetArn">The ARN of the Amazon SNS target.</param>
    /// <param name="eventBusArn">The optional event bus name, uses default if
empty.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> AddSnsTargetToRule(string ruleName, string
targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        // Create the list of targets and add a new target.
        var targets = new List<Target>
        {
            new Target()
            {
                Arn = targetArn,
                Id = targetID
            }
        };

        // Add the targets to the rule.
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
```

```
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
            });
        }

        return targetID;
    }
```

Adicione um transformador de entrada a um destino para uma regra.

```
    /// <summary>
    /// Update an Amazon S3 object created rule with a transform on the target.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <param name="eventBusArn">Optional event bus ARN. If empty, uses the
default event bus.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> UpdateS3UploadRuleTargetWithTransform(string
ruleName, string targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        var targets = new List<Target>
        {
            new Target()
            {
                Id = targetID,
                Arn = targetArn,
                InputTransformer = new InputTransformer()
                {
                    InputPathsMap = new Dictionary<string, string>()
                    {
                        {"bucket", "$.detail.bucket.name"},
                        {"time", "$.time"}
                    },
```

```
                    InputTemplate = "\"Notification: an object was uploaded to
bucket <bucket> at <time>.\""
                }
            }
        };
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
            });
        }
        return targetID;
    }
```

- Para obter detalhes da API, consulte PutTargetsa Referência AWS SDK for .NET da API.

C++

SDK para C++

> **Note**
>
> Tem mais sobre 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/events/EventBridgeClient.h>
#include <aws/events/model/PutTargetsRequest.h>
```

```
#include <aws/events/model/PutTargetsResult.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Adicione o destino.

```
        Aws::CloudWatchEvents::EventBridgeClient cwe;

        Aws::CloudWatchEvents::Model::Target target;
        target.SetArn(lambda_arn);
        target.SetId(target_id);

        Aws::CloudWatchEvents::Model::PutTargetsRequest request;
        request.SetRule(rule_name);
        request.AddTargets(target);

        auto putTargetsOutcome = cwe.PutTargets(request);
        if (!putTargetsOutcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch events target for rule "
                << rule_name << ": " <<
                putTargetsOutcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout <<
                "Successfully created CloudWatch events target for rule "
                << rule_name << std::endl;
        }
```

- Para obter detalhes da API, consulte PutTargetsa Referência AWS SDK for C++ da API.

CLI

AWS CLI

Para adicionar alvos às regras de CloudWatch eventos

Este exemplo adiciona uma função do Lambda como o destino de uma regra:

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="1","Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

Este exemplo define um fluxo do Amazon Kinesis como o destino. Desta forma, os eventos capturados pela regra são retransmitidos para o fluxo:

```
aws events put-targets --rule EC2InstanceStateChanges --targets
 "Id"="1","Arn"="arn:aws:kinesis:us-east-1:123456789012:stream/
MyStream","RoleArn"="arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

Este exemplo define dois fluxos do Amazon Kinesis como destinos de uma regra:

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="Target1","Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/
MyStream1","RoleArn"="arn:aws:iam::379642911888:role/ MyRoleToAccessLambda"
  "Id"="Target2"," Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/
MyStream2","RoleArn"="arn:aws:iam::379642911888:role/MyRoleToAccessLambda"
```

- Para obter detalhes da API, consulte PutTargetsna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Adicione um tópico do Amazon SNS como um destino para uma regra.

```
    // Add a rule which triggers an SNS target when a file is uploaded to an S3
    // bucket.
    public static void addSnsEventRule(EventBridgeClient eventBrClient, String
 ruleName, String topicArn,
            String topicName, String eventRuleName, String bucketName) {
        String targetID = java.util.UUID.randomUUID().toString();
        Target myTarget = Target.builder()
```

```
                .id(targetID)
                .arn(topicArn)
                .build();

        List<Target> targets = new ArrayList<>();
        targets.add(myTarget);
        PutTargetsRequest request = PutTargetsRequest.builder()
                .eventBusName(null)
                .targets(targets)
                .rule(ruleName)
                .build();

        eventBrClient.putTargets(request);
        System.out.println("Added event rule " + eventRuleName + " with Amazon
SNS target " + topicName + " for bucket "
                + bucketName + ".");
    }
```

Adicione um transformador de entrada a um destino para uma regra.

```
    public static void updateCustomRuleTargetWithTransform(EventBridgeClient
eventBrClient, String topicArn,
            String ruleName) {
        String targetId = java.util.UUID.randomUUID().toString();
        InputTransformer inputTransformer = InputTransformer.builder()
                .inputTemplate("\"Notification: sample event was received.\"")
                .build();

        Target target = Target.builder()
                .id(targetId)
                .arn(topicArn)
                .inputTransformer(inputTransformer)
                .build();

        try {
            PutTargetsRequest targetsRequest = PutTargetsRequest.builder()
                    .rule(ruleName)
                    .targets(target)
                    .eventBusName(null)
                    .build();

            eventBrClient.putTargets(targetsRequest);
```

```
        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Para obter detalhes da API, consulte PutTargetsa Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre 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.

```
import {
  EventBridgeClient,
  PutTargetsCommand,
} from "@aws-sdk/client-eventbridge";

export const putTarget = async (
  existingRuleName = "some-rule",
  targetArn = "arn:aws:lambda:us-east-1:000000000000:function:test-func",
  uniqueId = Date.now().toString(),
) => {
  const client = new EventBridgeClient({});
  const response = await client.send(
    new PutTargetsCommand({
      Rule: existingRuleName,
      Targets: [
        {
          Arn: targetArn,
          Id: uniqueId,
        },
      ],
    }),
```

```
  );

  console.log("PutTargets response:");
  console.log(response);
  // PutTargets response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: 'f5b23b9a-2c17-45c1-ad5c-f926c3692e3d',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   FailedEntries: [],
  //   FailedEntryCount: 0
  // }

  return response;
};
```

- Para obter detalhes da API, consulte PutTargetsa Referência AWS SDK for JavaScript da API.

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatchEvents service object
var ebevents = new AWS.EventBridge({ apiVersion: "2015-10-07" });

var params = {
  Rule: "DEMO_EVENT",
```

```
  Targets: [
    {
      Arn: "LAMBDA_FUNCTION_ARN",
      Id: "myEventBridgeTarget",
    },
  ],
};

ebevents.putTargets(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Para obter detalhes da API, consulte PutTargetsa Referência AWS SDK for JavaScript da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
// Add a rule that triggers an SNS target when a file is uploaded to an S3
 bucket.
suspend fun addSnsEventRule(ruleName: String?, topicArn: String?, topicName:
 String, eventRuleName: String, bucketName: String) {
    val targetID = UUID.randomUUID().toString()
    val myTarget = Target {
        id = targetID
        arn = topicArn
    }

    val targetsOb = mutableListOf<Target>()
```

```
    targetsOb.add(myTarget)

    val request = PutTargetsRequest {
        eventBusName = null
        targets = targetsOb
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(request)
        println("Added event rule $eventRuleName with Amazon SNS target
 $topicName for bucket $bucketName.")
    }
}
```

Adicione um transformador de entrada a um destino para uma regra.

```
suspend fun updateCustomRuleTargetWithTransform(topicArn: String?, ruleName:
 String?) {
    val targetId = UUID.randomUUID().toString()

    val inputTransformerOb = InputTransformer {
        inputTemplate = "\"Notification: sample event was received.\""
    }

    val target = Target {
        id = targetId
        arn = topicArn
        inputTransformer = inputTransformerOb
    }

    val targetsRequest = PutTargetsRequest {
        rule = ruleName
        targets = listOf(target)
        eventBusName = null
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(targetsRequest)
    }
}
```

- Para obter detalhes da API, consulte a PutTargetsreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`RemoveTargets`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `RemoveTargets`.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Remover todos os destinos de uma regra usando o nome da regra.

```
    /// <summary>
    /// Delete an event rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the event rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> RemoveAllTargetsFromRule(string ruleName)
    {
        var targetIds = new List<string>();
        var request = new ListTargetsByRuleRequest()
        {
            Rule = ruleName
        };
        ListTargetsByRuleResponse targetsResponse;
        do
        {
            targetsResponse = await
_amazonEventBridge.ListTargetsByRuleAsync(request);
            targetIds.AddRange(targetsResponse.Targets.Select(t => t.Id));
            request.NextToken = targetsResponse.NextToken;
```

```
        } while (targetsResponse.NextToken is not null);

        var removeResponse = await _amazonEventBridge.RemoveTargetsAsync(
            new RemoveTargetsRequest()
            {
                Rule = ruleName,
                Ids = targetIds
            });

        if (removeResponse.FailedEntryCount > 0)
        {
            removeResponse.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to remove target {e.TargetId}: {e.ErrorMessage},
code {e.ErrorCode}");
            });
        }

        return removeResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Para obter detalhes da API, consulte RemoveTargetsa Referência AWS SDK for .NET da API.

CLI

AWS CLI

Como remover um destino de um evento

Este exemplo remove o stream do Amazon Kinesis chamado MyStream 1 de ser um alvo da regra. DailyLambdaFunction Quando DailyLambdaFunction foi criado, esse fluxo foi definido como um destino com um ID de Target1:

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- Para obter detalhes da API, consulte RemoveTargetsna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Remover todos os destinos de uma regra usando o nome da regra.

```
   public static void deleteTargetsFromRule(EventBridgeClient eventBrClient,
String eventRuleName) {
       // First, get all targets that will be deleted.
       ListTargetsByRuleRequest request = ListTargetsByRuleRequest.builder()
               .rule(eventRuleName)
               .build();

       ListTargetsByRuleResponse response =
eventBrClient.listTargetsByRule(request);
       List<Target> allTargets = response.targets();

       // Get all targets and delete them.
       for (Target myTarget : allTargets) {
           RemoveTargetsRequest removeTargetsRequest =
RemoveTargetsRequest.builder()
                   .rule(eventRuleName)
                   .ids(myTarget.id())
                   .build();

           eventBrClient.removeTargets(removeTargetsRequest);
           System.out.println("Successfully removed the target");
       }
   }
```

- Para obter detalhes da API, consulte RemoveTargetsa Referência AWS SDK for Java 2.x da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun deleteTargetsFromRule(eventRuleName: String?) {
    // First, get all targets that will be deleted.
    val request = ListTargetsByRuleRequest {
        rule = eventRuleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(request)
        val allTargets = response.targets

        // Get all targets and delete them.
        if (allTargets != null) {
            for (myTarget in allTargets) {
                val removeTargetsRequest = RemoveTargetsRequest {
                    rule = eventRuleName
                    ids = listOf(myTarget.id.toString())
                }
                eventBrClient.removeTargets(removeTargetsRequest)
                println("Successfully removed the target")
            }
        }
    }
}
```

- Para obter detalhes da API, consulte a RemoveTargetsreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cenários para EventBridge usar AWS SDKs

Os exemplos de código a seguir mostram como implementar cenários comuns EventBridge com AWS SDKs. Esses cenários mostram como realizar tarefas específicas chamando várias funções internas EventBridge. Cada cenário inclui um link para GitHub, onde você pode encontrar instruções sobre como configurar e executar o código.

Exemplos

- Crie e acione uma regra na Amazon EventBridge usando um AWS SDK
- Comece a usar EventBridge regras e metas usando um AWS SDK

Crie e acione uma regra na Amazon EventBridge usando um AWS SDK

O exemplo de código a seguir mostra como criar e acionar uma regra na Amazon EventBridge.

Ruby

SDK para Ruby

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Chame as funções na ordem correta.

```
require "aws-sdk-sns"
require "aws-sdk-iam"
require "aws-sdk-cloudwatchevents"
require "aws-sdk-ec2"
require "aws-sdk-cloudwatch"
require "aws-sdk-cloudwatchlogs"
require "securerandom"
```

Verifica se o tópico do Amazon Simple Notification Service (Amazon SNS) existe dentre aqueles fornecidos para essa função.

```
# Checks whether the specified Amazon SNS
# topic exists among those provided to this function.
# This is a helper function that is called by the topic_exists? function.
#
# @param topics [Array] An array of Aws::SNS::Types::Topic objects.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   sns_client = Aws::SNS::Client.new(region: 'us-east-1')
#   response = sns_client.list_topics
#   if topic_found?(
#     response.topics,
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
#     puts 'Topic found.'
#   end

def topic_found?(topics, topic_arn)
  topics.each do |topic|
    return true if topic.topic_arn == topic_arn
  end
  return false
end
```

Verifica se o tópico especificado existe dentre aqueles disponíveis para o chamador no Amazon SNS.

```
# Checks whether the specified topic exists among those available to the
# caller in Amazon SNS.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   exit 1 unless topic_exists?(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def topic_exists?(sns_client, topic_arn)
  puts "Searching for topic with ARN '#{topic_arn}'..."
  response = sns_client.list_topics
  if response.topics.count.positive?
```

```
    if topic_found?(response.topics, topic_arn)
      puts "Topic found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.topics.count.positive?
        if topic_found?(response.topics, topic_arn)
          puts "Topic found."
          return true
        end
      end
    end
  end
  puts "Topic not found."
  return false
rescue StandardError => e
  puts "Topic not found: #{e.message}"
  return false
end
```

Crie um tópico no Amazon SNS e, em seguida, assine um endereço de e-mail para receber notificações sobre esse tópico.

```
# Creates a topic in Amazon SNS
# and then subscribes an email address to receive notifications to that topic.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_name [String] The name of the topic to create.
# @param email_address [String] The email address of the recipient to notify.
# @return [String] The ARN of the topic that was created.
# @example
#   puts create_topic(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-topic',
#     'mary@example.com'
#   )
def create_topic(sns_client, topic_name, email_address)
  puts "Creating the topic named '#{topic_name}'..."
  topic_response = sns_client.create_topic(name: topic_name)
  puts "Topic created with ARN '#{topic_response.topic_arn}'."
  subscription_response = sns_client.subscribe(
```

```
    topic_arn: topic_response.topic_arn,
    protocol: "email",
    endpoint: email_address,
    return_subscription_arn: true
  )
  puts "Subscription created with ARN " \
    "'#{subscription_response.subscription_arn}'. Have the owner of the " \
    "email address '#{email_address}' check their inbox in a few minutes " \
    "and confirm the subscription to start receiving notification emails."
  return topic_response.topic_arn
rescue StandardError => e
  puts "Error creating or subscribing to topic: #{e.message}"
  return "Error"
end
```

Verifique se a função especificada AWS Identity and Access Management (IAM) existe entre as fornecidas para essa função.

```
# Checks whether the specified AWS Identity and Access Management (IAM)
# role exists among those provided to this function.
# This is a helper function that is called by the role_exists? function.
#
# @param roles [Array] An array of Aws::IAM::Role objects.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-east-1')
#   response = iam_client.list_roles
#   if role_found?(
#     response.roles,
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
#     puts 'Role found.'
#   end
def role_found?(roles, role_arn)
  roles.each do |role|
    return true if role.arn == role_arn
  end
  return false
end
```

Verificar se o perfil especificado existe dentre aqueles disponíveis para o chamador no IAM.

```
# Checks whether the specified role exists among those available to the
# caller in AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   exit 1 unless role_exists?(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
def role_exists?(iam_client, role_arn)
  puts "Searching for role with ARN '#{role_arn}'..."
  response = iam_client.list_roles
  if response.roles.count.positive?
    if role_found?(response.roles, role_arn)
      puts "Role found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.roles.count.positive?
        if role_found?(response.roles, role_arn)
          puts "Role found."
          return true
        end
      end
    end
  end
  puts "Role not found."
  return false
rescue StandardError => e
  puts "Role not found: #{e.message}"
  return false
end
```

Criar um perfil do IAM.

```
# Creates a role in AWS Identity and Access Management (IAM).
# This role is used by a rule in Amazon EventBridge to allow
```

```
# that rule to operate within the caller's account.
# This role is designed to be used specifically by this code example.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to create.
# @return [String] The ARN of the role that was created.
# @example
#   puts create_role(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def create_role(iam_client, role_name)
  puts "Creating the role named '#{role_name}'..."
  response = iam_client.create_role(
    assume_role_policy_document: {
      'Version': "2012-10-17",
      'Statement': [
        {
          'Sid': "",
          'Effect': "Allow",
          'Principal': {
            'Service': "events.amazonaws.com"
          },
          'Action': "sts:AssumeRole"
        }
      ]
    }.to_json,
    path: "/",
    role_name: role_name
  )
  puts "Role created with ARN '#{response.role.arn}'."
  puts "Adding access policy to role..."
  iam_client.put_role_policy(
    policy_document: {
      'Version': "2012-10-17",
      'Statement': [
        {
          'Sid': "CloudWatchEventsFullAccess",
          'Effect': "Allow",
          'Resource': "*",
          'Action': "events:*"
        },
        {
          'Sid': "IAMPassRoleForCloudWatchEvents",
```

```
          'Effect': "Allow",
          'Resource': "arn:aws:iam::*:role/AWS_Events_Invoke_Targets",
          'Action': "iam:PassRole"
        }
      ]
    }.to_json,
    policy_name: "CloudWatchEventsPolicy",
    role_name: role_name
  )
  puts "Access policy added to role."
  return response.role.arn
rescue StandardError => e
  puts "Error creating role or adding policy to it: #{e.message}"
  puts "If the role was created, you must add the access policy " \
    "to the role yourself, or delete the role yourself and try again."
  return "Error"
end
```

Verifica se a EventBridge regra especificada existe entre as fornecidas para essa função.

```
# Checks whether the specified Amazon EventBridge rule exists among
# those provided to this function.
# This is a helper function that is called by the rule_exists? function.
#
# @param rules [Array] An array of Aws::CloudWatchEvents::Types::Rule objects.
# @param rule_arn [String] The name of the rule to find.
# @return [Boolean] true if the name of the rule was found; otherwise, false.
# @example
#   cloudwatchevents_client = Aws::CloudWatch::Client.new(region: 'us-east-1')
#   response = cloudwatchevents_client.list_rules
#   if rule_found?(response.rules, 'aws-doc-sdk-examples-ec2-state-change')
#     puts 'Rule found.'
#   end
def rule_found?(rules, rule_name)
  rules.each do |rule|
    return true if rule.name == rule_name
  end
  return false
end
```

Verifica se a regra especificada existe entre as disponíveis para o chamador. EventBridge

```
# Checks whether the specified rule exists among those available to the
# caller in Amazon EventBridge.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized Amazon EventBridge client.
# @param rule_name [String] The name of the rule to find.
# @return [Boolean] true if the rule name was found; otherwise, false.
# @example
#   exit 1 unless rule_exists?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1')
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def rule_exists?(cloudwatchevents_client, rule_name)
  puts "Searching for rule with name '#{rule_name}'..."
  response = cloudwatchevents_client.list_rules
  if response.rules.count.positive?
    if rule_found?(response.rules, rule_name)
      puts "Rule found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.rules.count.positive?
        if rule_found?(response.rules, rule_name)
          puts "Rule found."
          return true
        end
      end
    end
  end
  puts "Rule not found."
  return false
rescue StandardError => e
  puts "Rule not found: #{e.message}"
  return false
end
```

Crie uma regra em EventBridge.

```
# Creates a rule in Amazon EventBridge.
# This rule is triggered whenever an available instance in
# Amazon EC2 changes to the specified state.
```

```
# This rule is designed to be used specifically by this code example.
#
# Prerequisites:
#
# - A role in AWS Identity and Access Management (IAM) that is designed
#   to be used specifically by this code example.
# - A topic in Amazon SNS.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized Amazon EventBridge client.
# @param rule_name [String] The name of the rule to create.
# @param rule_description [String] Some description for this rule.
# @param instance_state [String] The state that available instances in
#   Amazon EC2 must change to, to
#   trigger this rule.
# @param role_arn [String] The Amazon Resource Name (ARN) of the IAM role.
# @param target_id [String] Some identifying string for the rule's target.
# @param topic_arn [String] The ARN of the Amazon SNS topic.
# @return [Boolean] true if the rule was created; otherwise, false.
# @example
#   exit 1 unless rule_created?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     'Triggers when any available EC2 instance starts.',
#     'running',
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change',
#     'sns-topic',
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def rule_created?(
  cloudwatchevents_client,
  rule_name,
  rule_description,
  instance_state,
  role_arn,
  target_id,
  topic_arn
)
  puts "Creating rule with name '#{rule_name}'..."
  put_rule_response = cloudwatchevents_client.put_rule(
    name: rule_name,
    description: rule_description,
    event_pattern: {
      'source': [
```

```
        "aws.ec2"
      ],
      'detail-type': [
        "EC2 Instance State-change Notification"
      ],
      'detail': {
        'state': [
          instance_state
        ]
      }
    }.to_json,
    state: "ENABLED",
    role_arn: role_arn
  )
  puts "Rule created with ARN '#{put_rule_response.rule_arn}'."

  put_targets_response = cloudwatchevents_client.put_targets(
    rule: rule_name,
    targets: [
      {
        id: target_id,
        arn: topic_arn
      }
    ]
  )
  if put_targets_response.key?(:failed_entry_count) &&
      put_targets_response.failed_entry_count > 0
    puts "Error(s) adding target to rule:"
    put_targets_response.failed_entries.each do |failure|
      puts failure.error_message
    end
    return false
  else
    return true
  end
rescue StandardError => e
  puts "Error creating rule or adding target to rule: #{e.message}"
  puts "If the rule was created, you must add the target " \
    "to the rule yourself, or delete the rule yourself and try again."
  return false
end
```

Verifique se o grupo de registros especificado existe entre aqueles disponíveis para o chamador no Amazon CloudWatch Logs.

```
# Checks to see whether the specified log group exists among those available
# to the caller in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to find.
# @return [Boolean] true if the log group name was found; otherwise, false.
# @example
#   exit 1 unless log_group_exists?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_exists?(cloudwatchlogs_client, log_group_name)
  puts "Searching for log group with name '#{log_group_name}'..."
  response = cloudwatchlogs_client.describe_log_groups(
    log_group_name_prefix: log_group_name
  )
  if response.log_groups.count.positive?
    response.log_groups.each do |log_group|
      if log_group.log_group_name == log_group_name
        puts "Log group found."
        return true
      end
    end
  end
  puts "Log group not found."
  return false
rescue StandardError => e
  puts "Log group not found: #{e.message}"
  return false
end
```

Crie um grupo de CloudWatch registros em Registros.

```
# Creates a log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to create.
```

```
# @return [Boolean] true if the log group name was created; otherwise, false.
# @example
#   exit 1 unless log_group_created?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_created?(cloudwatchlogs_client, log_group_name)
  puts "Attempting to create log group with the name '#{log_group_name}'..."
  cloudwatchlogs_client.create_log_group(log_group_name: log_group_name)
  puts "Log group created."
  return true
rescue StandardError => e
  puts "Error creating log group: #{e.message}"
  return false
end
```

Grave um evento em um stream de CloudWatch registros em Logs.

```
# Writes an event to a log stream in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
# - A log stream within the log group.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @param log_stream_name [String] The name of the log stream within
#   the log group.
# @param message [String] The message to write to the log stream.
# @param sequence_token [String] If available, the sequence token from the
#   message that was written immediately before this message. This sequence
#   token is returned by Amazon CloudWatch Logs whenever you programmatically
#   write a message to the log stream.
# @return [String] The sequence token that is returned by
#   Amazon CloudWatch Logs after successfully writing the message to the
#   log stream.
# @example
#   puts log_event(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
```

```
#     '2020/11/19/53f985be-199f-408e-9a45-fc242df41fEX',
#     "Instance 'i-033c48ef067af3dEX' restarted.",
#     '495426724868310740095796045676567882148068632824696073EX'
#   )
def log_event(
  cloudwatchlogs_client,
  log_group_name,
  log_stream_name,
  message,
  sequence_token
)
  puts "Attempting to log '#{message}' to log stream '#{log_stream_name}'..."
  event = {
    log_group_name: log_group_name,
    log_stream_name: log_stream_name,
    log_events: [
      {
        timestamp: (Time.now.utc.to_f.round(3) * 1_000).to_i,
        message: message
      }
    ]
  }
  unless sequence_token.empty?
    event[:sequence_token] = sequence_token
  end

  response = cloudwatchlogs_client.put_log_events(event)
  puts "Message logged."
  return response.next_sequence_token
rescue StandardError => e
  puts "Message not logged: #{e.message}"
end
```

Reinicie uma instância do Amazon Elastic Compute Cloud (Amazon EC2) e adicione informações sobre a atividade relacionada a um stream de log no Logs. CloudWatch

```

# Restarts an Amazon EC2 instance
# and adds information about the related activity to a log stream
# in Amazon CloudWatch Logs.
#
# Prerequisites:
```

```
#
# - The Amazon EC2 instance to restart.
# - The log group in Amazon CloudWatch Logs to add related activity
#   information to.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client]
#   An initialized Amazon CloudWatch Logs client.
# @param instance_id [String] The ID of the instance.
# @param log_group_name [String] The name of the log group.
# @return [Boolean] true if the instance was restarted and the information
#   was written to the log stream; otherwise, false.
# @example
#   exit 1 unless instance_restarted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX',
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  log_stream_name = "#{Time.now.year}/#{Time.now.month}/#{Time.now.day}/" \
    "#{SecureRandom.uuid}"
  cloudwatchlogs_client.create_log_stream(
    log_group_name: log_group_name,
    log_stream_name: log_stream_name
  )
  sequence_token = ""

  puts "Attempting to stop the instance with the ID '#{instance_id}'. " \
    "This might take a few minutes..."
  ec2_client.stop_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
  puts "Instance stopped."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' stopped.",
    sequence_token
```

```
  )

  puts "Attempting to restart the instance. This might take a few minutes..."
  ec2_client.start_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
  puts "Instance restarted."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' restarted.",
    sequence_token
  )

  return true
rescue StandardError => e
  puts "Error creating log stream or stopping or restarting the instance: " \
    "#{e.message}"
  log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Error stopping or starting instance '#{instance_id}': #{e.message}",
    sequence_token
  )
  return false
end
```

Exibir informações sobre a atividade de uma regra em EventBridge.

```
# Displays information about activity for a rule in Amazon EventBridge.
#
# Prerequisites:
#
# - A rule in Amazon EventBridge.
#
# @param cloudwatch_client [Amazon::CloudWatch::Client] An initialized
#   Amazon CloudWatch client.
# @param rule_name [String] The name of the rule.
# @param start_time [Time] The timestamp that determines the first datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param end_time [Time] The timestamp that determines the last datapoint
```

```
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param period [Integer] The interval, in seconds, to check for activity.
# @example
#   display_rule_activity(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     Time.now - 600, # Start checking from 10 minutes ago.
#     Time.now, # Check up until now.
#     60 # Check every minute during those 10 minutes.
#   )
def display_rule_activity(
  cloudwatch_client,
  rule_name,
  start_time,
  end_time,
  period
)
  puts "Attempting to display rule activity..."
  response = cloudwatch_client.get_metric_statistics(
    namespace: "AWS/Events",
    metric_name: "Invocations",
    dimensions: [
      {
        name: "RuleName",
        value: rule_name
      }
    ],
    start_time: start_time,
    end_time: end_time,
    period: period,
    statistics: ["Sum"],
    unit: "Count"
  )

  if response.key?(:datapoints) && response.datapoints.count.positive?
    puts "The event rule '#{rule_name}' was triggered:"
    response.datapoints.each do |datapoint|
      puts "  #{datapoint.sum} time(s) at #{datapoint.timestamp}"
    end
  else
    puts "The event rule '#{rule_name}' was not triggered during the " \
      "specified time period."
  end
rescue StandardError => e
```

```
  puts "Error getting information about event rule activity: #{e.message}"
end
```

Exibir informações de registro de todos os fluxos de registros em um grupo de CloudWatch registros de registros.

```
# Displays log information for all of the log streams in a log group in
# Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Amazon::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @example
#   display_log_data(
#     Amazon::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def display_log_data(cloudwatchlogs_client, log_group_name)
  puts "Attempting to display log stream data for the log group " \
    "named '#{log_group_name}'..."
  describe_log_streams_response = cloudwatchlogs_client.describe_log_streams(
    log_group_name: log_group_name,
    order_by: "LastEventTime",
    descending: true
  )
  if describe_log_streams_response.key?(:log_streams) &&
      describe_log_streams_response.log_streams.count.positive?
    describe_log_streams_response.log_streams.each do |log_stream|
      get_log_events_response = cloudwatchlogs_client.get_log_events(
        log_group_name: log_group_name,
        log_stream_name: log_stream.log_stream_name
      )
      puts "\nLog messages for '#{log_stream.log_stream_name}':"
      puts "-" * (log_stream.log_stream_name.length + 20)
      if get_log_events_response.key?(:events) &&
          get_log_events_response.events.count.positive?
        get_log_events_response.events.each do |event|
          puts event.message
```

```
        end
      else
        puts "No log messages for this log stream."
      end
    end
  end
rescue StandardError => e
  puts "Error getting information about the log streams or their messages: " \
    "#{e.message}"
end
```

Exiba um lembrete para o chamador limpar manualmente todos AWS os recursos associados dos quais ele não precisa mais.

```
# Displays a reminder to the caller to manually clean up any associated
# AWS resources that they no longer need.
#
# @param topic_name [String] The name of the Amazon SNS topic.
# @param role_name [String] The name of the IAM role.
# @param rule_name [String] The name of the Amazon EventBridge rule.
# @param log_group_name [String] The name of the Amazon CloudWatch Logs log
 group.
# @param instance_id [String] The ID of the Amazon EC2 instance.
# @example
#   manual_cleanup_notice(
#     'aws-doc-sdk-examples-topic',
#     'aws-doc-sdk-examples-cloudwatch-events-rule-role',
#     'aws-doc-sdk-examples-ec2-state-change',
#     'aws-doc-sdk-examples-cloudwatch-log',
#     'i-033c48ef067af3dEX'
#   )
def manual_cleanup_notice(
  topic_name, role_name, rule_name, log_group_name, instance_id
)
  puts "-" * 10
  puts "Some of the following AWS resources might still exist in your account."
  puts "If you no longer want to use this code example, then to clean up"
  puts "your AWS account and avoid unexpected costs, you might want to"
  puts "manually delete any of the following resources if they exist:"
  puts "- The Amazon SNS topic named '#{topic_name}'."
  puts "- The IAM role named '#{role_name}'."
```

```
  puts "- The Amazon EventBridge rule named '#{rule_name}'."
  puts "- The Amazon CloudWatch Logs log group named '#{log_group_name}'."
  puts "- The Amazon EC2 instance with the ID '#{instance_id}'."
end

# Example usage:
def run_me
  # Properties for the Amazon SNS topic.
  topic_name = "aws-doc-sdk-examples-topic"
  email_address = "mary@example.com"
  # Properties for the IAM role.
  role_name = "aws-doc-sdk-examples-cloudwatch-events-rule-role"
  # Properties for the Amazon EventBridge rule.
  rule_name = "aws-doc-sdk-examples-ec2-state-change"
  rule_description = "Triggers when any available EC2 instance starts."
  instance_state = "running"
  target_id = "sns-topic"
  # Properties for the Amazon EC2 instance.
  instance_id = "i-033c48ef067af3dEX"
  # Properties for displaying the event rule's activity.
  start_time = Time.now - 600 # Go back over the past 10 minutes
                              # (10 minutes * 60 seconds = 600 seconds).
  end_time = Time.now
  period = 60 # Look back every 60 seconds over the past 10 minutes.
  # Properties for the Amazon CloudWatch Logs log group.
  log_group_name = "aws-doc-sdk-examples-cloudwatch-log"
  # AWS service clients for this code example.
  region = "us-east-1"
  sts_client = Aws::STS::Client.new(region: region)
  sns_client = Aws::SNS::Client.new(region: region)
  iam_client = Aws::IAM::Client.new(region: region)
  cloudwatchevents_client = Aws::CloudWatchEvents::Client.new(region: region)
  ec2_client = Aws::EC2::Client.new(region: region)
  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
  cloudwatchlogs_client = Aws::CloudWatchLogs::Client.new(region: region)

  # Get the caller's account ID for use in forming
  # Amazon Resource Names (ARNs) that this code relies on later.
  account_id = sts_client.get_caller_identity.account

  # If the Amazon SNS topic doesn't exist, create it.
  topic_arn = "arn:aws:sns:#{region}:#{account_id}:#{topic_name}"
  unless topic_exists?(sns_client, topic_arn)
    topic_arn = create_topic(sns_client, topic_name, email_address)
```

```
  if topic_arn == "Error"
    puts "Could not create the Amazon SNS topic correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
    exit 1
  end
end

# If the IAM role doesn't exist, create it.
role_arn = "arn:aws:iam::#{account_id}:role/#{role_name}"
unless role_exists?(iam_client, role_arn)
  role_arn = create_role(iam_client, role_name)
  if role_arn == "Error"
    puts "Could not create the IAM role correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon EventBridge rule doesn't exist, create it.
unless rule_exists?(cloudwatchevents_client, rule_name)
  unless rule_created?(
    cloudwatchevents_client,
    rule_name,
    rule_description,
    instance_state,
    role_arn,
    target_id,
    topic_arn
  )
    puts "Could not create the Amazon EventBridge rule correctly. " \
      "Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon CloudWatch Logs log group doesn't exist, create it.
unless log_group_exists?(cloudwatchlogs_client, log_group_name)
  unless log_group_created?(cloudwatchlogs_client, log_group_name)
    puts "Could not create the Amazon CloudWatch Logs log group " \
```

```
      "correctly. Program stopped."
      manual_cleanup_notice(
        topic_name, role_name, rule_name, log_group_name, instance_id
      )
    end
  end

  # Restart the Amazon EC2 instance, which triggers the rule.
  unless instance_restarted?(
    ec2_client,
    cloudwatchlogs_client,
    instance_id,
    log_group_name
  )
    puts "Could not restart the instance to trigger the rule. " \
      "Continuing anyway to show information about the rule and logs..."
  end

  # Display how many times the rule was triggered over the past 10 minutes.
  display_rule_activity(
    cloudwatch_client,
    rule_name,
    start_time,
    end_time,
    period
  )

  # Display related log data in Amazon CloudWatch Logs.
  display_log_data(cloudwatchlogs_client, log_group_name)

  # Reminder the caller to clean up any AWS resources that are used
  # by this code example and are no longer needed.
  manual_cleanup_notice(
    topic_name, role_name, rule_name, log_group_name, instance_id
  )
end

run_me if $PROGRAM_NAME == __FILE__
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Ruby .
  - PutEvents

- PutRule

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Comece a usar EventBridge regras e metas usando um AWS SDK

Os exemplos de código a seguir mostram como:

- Crie uma regra e adicione um destino a ela.
- Habilitar e desabilitar regras.
- Listar e atualizar regras e destinos.
- Enviar eventos e, em seguida, limpar os recursos.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Execute um cenário interativo em um prompt de comando.

```
public class EventBridgeScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    This .NET example performs the following tasks with Amazon EventBridge:
    - Create a rule.
    - Add a target to a rule.
    - Enable and disable rules.
    - List rules and targets.
    - Update rules and targets.
```

```
    - Send events.
    - Delete the rule.
    */

    private static ILogger logger = null!;
    private static EventBridgeWrapper _eventBridgeWrapper = null!;
    private static IConfiguration _configuration = null!;

    private static IAmazonIdentityManagementService? _iamClient = null!;
    private static IAmazonSimpleNotificationService? _snsClient = null!;
    private static IAmazonS3 _s3Client = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for Amazon EventBridge.
        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<IAmazonEventBridge>()
            .AddAWSService<IAmazonIdentityManagementService>()
            .AddAWSService<IAmazonS3>()
            .AddAWSService<IAmazonSimpleNotificationService>()
            .AddTransient<EventBridgeWrapper>()
            )
            .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<EventBridgeScenario>();

        ServicesSetup(host);

        string topicArn = "";
```

```
        string roleArn = "";

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the Amazon EventBridge example scenario.");
        Console.WriteLine(new string('-', 80));

        try
        {
            roleArn = await CreateRole();

            await CreateBucketWithEventBridgeEvents();

            await AddEventRule(roleArn);

            await ListEventRules();

            topicArn = await CreateSnsTopic();

            var email = await SubscribeToSnsTopic(topicArn);

            await AddSnsTarget(topicArn);

            await ListTargets();

            await ListRulesForTarget(topicArn);

            await UploadS3File(_s3Client);

            await ChangeRuleState(false);

            await GetRuleState();

            await UpdateSnsEventRule(topicArn);

            await ChangeRuleState(true);

            await UploadS3File(_s3Client);

            await UpdateToCustomRule(topicArn);

            await TriggerCustomRule(email);

            await CleanupResources(topicArn);
        }
```

```
        catch (Exception ex)
        {
            logger.LogError(ex, "There was a problem executing the scenario.");
            await CleanupResources(topicArn);
        }
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("The Amazon EventBridge example scenario is
complete.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Populate the services for use within the console application.
    /// </summary>
    /// <param name="host">The services host.</param>
    private static void ServicesSetup(IHost host)
    {
        _eventBridgeWrapper =
host.Services.GetRequiredService<EventBridgeWrapper>();
        _snsClient =
host.Services.GetRequiredService<IAmazonSimpleNotificationService>();
        _s3Client = host.Services.GetRequiredService<IAmazonS3>();
        _iamClient =
host.Services.GetRequiredService<IAmazonIdentityManagementService>();
    }

    /// <summary>
    /// Create a role to be used by EventBridge.
    /// </summary>
    /// <returns>The role Amazon Resource Name (ARN).</returns>
    public static async Task<string> CreateRole()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Creating a role to use with EventBridge and attaching
managed policy AmazonEventBridgeFullAccess.");
        Console.WriteLine(new string('-', 80));

        var roleName = _configuration["roleName"];

        var assumeRolePolicy = "{" +
                               "\"Version\": \"2012-10-17\"," +
                               "\"Statement\": [{" +
                               "\"Effect\": \"Allow\"," +
                               "\"Principal\": {" +
```

```
                                $"\"Service\": \"events.amazonaws.com\"" +
                                "}," +
                                "\"Action\": \"sts:AssumeRole\"" +
                                "}]" +
                                "}";

        var roleResult = await _iamClient!.CreateRoleAsync(
            new CreateRoleRequest()
            {
                AssumeRolePolicyDocument = assumeRolePolicy,
                Path = "/",
                RoleName = roleName
            });

        await _iamClient.AttachRolePolicyAsync(
            new AttachRolePolicyRequest()
            {
                PolicyArn = "arn:aws:iam::aws:policy/
AmazonEventBridgeFullAccess",
                RoleName = roleName
            });
        // Allow time for the role to be ready.
        Thread.Sleep(10000);
        return roleResult.Role.Arn;
    }

    /// <summary>
    /// Create an Amazon Simple Storage Service (Amazon S3) bucket with
 EventBridge events enabled.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateBucketWithEventBridgeEvents()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Creating an S3 bucket with EventBridge events
 enabled.");

        var testBucketName = _configuration["testBucketName"];

        var bucketExists = await
 Amazon.S3.Util.AmazonS3Util.DoesS3BucketExistV2Async(_s3Client,
            testBucketName);

        if (!bucketExists)
```

```
        {
            await _s3Client.PutBucketAsync(new PutBucketRequest()
            {
                BucketName = testBucketName,
                UseClientRegion = true
            });
        }

        await _s3Client.PutBucketNotificationAsync(new
PutBucketNotificationRequest()
        {
            BucketName = testBucketName,
            EventBridgeConfiguration = new EventBridgeConfiguration()
        });

        Console.WriteLine($"\tAdded bucket {testBucketName} with EventBridge
events enabled.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create and upload a file to an S3 bucket to trigger an event.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task UploadS3File(IAmazonS3 s3Client)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Uploading a file to the test bucket. This will trigger
a subscription email.");

        var testBucketName = _configuration["testBucketName"];

        var fileName = $"example_upload_{DateTime.UtcNow.Ticks}.txt";

        // Create the file if it does not already exist.
        if (!File.Exists(fileName))
        {
            await using StreamWriter sw = File.CreateText(fileName);
            await sw.WriteLineAsync(
                "This is a sample file for testing uploads.");
        }

        await s3Client.PutObjectAsync(new PutObjectRequest()
```

```
        {
            FilePath = fileName,
            BucketName = testBucketName
        });

        Console.WriteLine($"\tPress Enter to continue.");
        Console.ReadLine();

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create an Amazon Simple Notification Service (Amazon SNS) topic to use as
an EventBridge target.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task<string> CreateSnsTopic()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine(
            "Creating an Amazon Simple Notification Service (Amazon SNS) topic
for email subscriptions.");

        var topicName = _configuration["topicName"];

        string topicPolicy = "{" +
                             "\"Version\": \"2012-10-17\"," +
                             "\"Statement\": [{" +
                             "\"Sid\": \"EventBridgePublishTopic\"," +
                             "\"Effect\": \"Allow\"," +
                             "\"Principal\": {" +
                             $"\"Service\": \"events.amazonaws.com\"" +
                             "}," +
                             "\"Resource\": \"*\"," +
                             "\"Action\": \"sns:Publish\"" +
                             "}]" +
                             "}";

        var topicAttributes = new Dictionary<string, string>()
        {
            { "Policy", topicPolicy }
        };
```

```
        var topicResponse = await _snsClient!.CreateTopicAsync(new
CreateTopicRequest()
        {
            Name = topicName,
            Attributes = topicAttributes

        });

        Console.WriteLine($"\tAdded topic {topicName} for email subscriptions.");

        Console.WriteLine(new string('-', 80));

        return topicResponse.TopicArn;
    }

    /// <summary>
    /// Subscribe a user email to an SNS topic.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic.</param>
    /// <returns>The user's email.</returns>
    private static async Task<string> SubscribeToSnsTopic(string topicArn)
    {
        Console.WriteLine(new string('-', 80));


        string email = "";
        while (string.IsNullOrEmpty(email))
        {
            Console.WriteLine("Enter your email to subscribe to the Amazon SNS
topic:");
            email = Console.ReadLine()!;
        }

        var subscriptions = new List<string>();
        var paginatedSubscriptions =
_snsClient!.Paginators.ListSubscriptionsByTopic(
            new ListSubscriptionsByTopicRequest()
            {
                TopicArn = topicArn
            });

        // Get the entire list using the paginator.
        await foreach (var subscription in paginatedSubscriptions.Subscriptions)
        {
```

```
            subscriptions.Add(subscription.Endpoint);
        }

        if (subscriptions.Contains(email))
        {
            Console.WriteLine($"\tYour email is already subscribed.");
            Console.WriteLine(new string('-', 80));
            return email;
        }

        await _snsClient.SubscribeAsync(new SubscribeRequest()
        {
            TopicArn = topicArn,
            Protocol = "email",
            Endpoint = email
        });

        Console.WriteLine($"Use the link in the email you received to confirm
your subscription, then press Enter to continue.");

        Console.ReadLine();

        Console.WriteLine(new string('-', 80));
        return email;
    }

    /// <summary>
    /// Add a rule which triggers when a file is uploaded to an S3 bucket.
    /// </summary>
    /// <param name="roleArn">The ARN of the role used by EventBridge.</param>
    /// <returns>Async task.</returns>
    private static async Task AddEventRule(string roleArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Creating an EventBridge event that sends an email when
an Amazon S3 object is created.");

        var eventRuleName = _configuration["eventRuleName"];
        var testBucketName = _configuration["testBucketName"];

        await _eventBridgeWrapper.PutS3UploadRule(roleArn, eventRuleName,
testBucketName);
        Console.WriteLine($"\tAdded event rule {eventRuleName} for bucket
{testBucketName}.");
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add an SNS target to the rule.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic.</param>
    /// <returns>Async task.</returns>
    private static async Task AddSnsTarget(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Adding a target to the rule to that sends an email
when the rule is triggered.");

        var eventRuleName = _configuration["eventRuleName"];
        var testBucketName = _configuration["testBucketName"];
        var topicName = _configuration["topicName"];
        await _eventBridgeWrapper.AddSnsTargetToRule(eventRuleName, topicArn);
        Console.WriteLine($"\tAdded event rule {eventRuleName} with Amazon SNS
target {topicName} for bucket {testBucketName}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List the event rules on the default event bus.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task ListEventRules()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Current event rules:");

        var rules = await _eventBridgeWrapper.ListAllRulesForEventBus();
        rules.ForEach(r => Console.WriteLine($"\tRule: {r.Name} Description:
{r.Description} State: {r.State}"));

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Update the event target to use a transform.
    /// </summary>
```

```
    /// <param name="topicArn">The SNS topic ARN target to update.</param>
    /// <returns>Async task.</returns>
    private static async Task UpdateSnsEventRule(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Let's update the event target with a transform.");

        var eventRuleName = _configuration["eventRuleName"];
        var testBucketName = _configuration["testBucketName"];

        await
_eventBridgeWrapper.UpdateS3UploadRuleTargetWithTransform(eventRuleName,
topicArn);
        Console.WriteLine($"\tUpdated event rule {eventRuleName} with Amazon SNS
target {topicArn} for bucket {testBucketName}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Update the rule to use a custom event pattern.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task UpdateToCustomRule(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Updating the event pattern to be triggered by a custom
event instead.");

        var eventRuleName = _configuration["eventRuleName"];

        await _eventBridgeWrapper.UpdateCustomEventPattern(eventRuleName);

        Console.WriteLine($"\tUpdated event rule {eventRuleName} to custom
pattern.");
        await
_eventBridgeWrapper.UpdateCustomRuleTargetWithTransform(eventRuleName,
            topicArn);

        Console.WriteLine($"\tUpdated event target {topicArn}.");

        Console.WriteLine(new string('-', 80));
    }
```

```
    /// <summary>
    /// Send rule events for a custom rule using the user's email address.
    /// </summary>
    /// <param name="email">The email address to include.</param>
    /// <returns>Async task.</returns>
    private static async Task TriggerCustomRule(string email)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Sending an event to trigger the rule. This will
trigger a subscription email.");

        await _eventBridgeWrapper.PutCustomEmailEvent(email);

        Console.WriteLine($"\tEvents have been sent. Press Enter to continue.");
        Console.ReadLine();

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List all of the targets for a rule.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task ListTargets()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("List all of the targets for a particular rule.");

        var eventRuleName = _configuration["eventRuleName"];
        var targets = await
_eventBridgeWrapper.ListAllTargetsOnRule(eventRuleName);
        targets.ForEach(t => Console.WriteLine($"\tTarget: {t.Arn} Id: {t.Id}
Input: {t.Input}"));

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List all of the rules for a particular target.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic.</param>
    /// <returns>Async task.</returns>
    private static async Task ListRulesForTarget(string topicArn)
    {
```

```
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("List all of the rules for a particular target.");

        var rules = await _eventBridgeWrapper.ListAllRuleNamesByTarget(topicArn);
        rules.ForEach(r => Console.WriteLine($"\tRule: {r}"));

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Enable or disable a particular rule.
    /// </summary>
    /// <param name="isEnabled">True to enable the rule, otherwise false.</param>
    /// <returns>Async task.</returns>
    private static async Task ChangeRuleState(bool isEnabled)
    {
        Console.WriteLine(new string('-', 80));
        var eventRuleName = _configuration["eventRuleName"];

        if (!isEnabled)
        {
            Console.WriteLine($"Disabling the rule: {eventRuleName}");
            await _eventBridgeWrapper.DisableRuleByName(eventRuleName);
        }
        else
        {
            Console.WriteLine($"Enabling the rule: {eventRuleName}");
            await _eventBridgeWrapper.EnableRuleByName(eventRuleName);
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get the current state of the rule.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetRuleState()
    {
        Console.WriteLine(new string('-', 80));
        var eventRuleName = _configuration["eventRuleName"];

        var state = await
_eventBridgeWrapper.GetRuleStateByRuleName(eventRuleName);
```

```
        Console.WriteLine($"Rule {eventRuleName} is in current state {state}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Clean up the resources from the scenario.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic to clean up.</param>
    /// <returns>Async task.</returns>
    private static async Task CleanupResources(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"Clean up resources.");

        var eventRuleName = _configuration["eventRuleName"];
        if (GetYesNoResponse($"\tDelete all targets and event rule
{eventRuleName}? (y/n)"))
        {
            Console.WriteLine($"\tRemoving all targets from the event rule.");
            await _eventBridgeWrapper.RemoveAllTargetsFromRule(eventRuleName);

            Console.WriteLine($"\tDeleting event rule.");
            await _eventBridgeWrapper.DeleteRuleByName(eventRuleName);
        }

        var topicName = _configuration["topicName"];
        if (GetYesNoResponse($"\tDelete Amazon SNS subscription topic
{topicName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting topic.");
            await _snsClient!.DeleteTopicAsync(new DeleteTopicRequest()
            {
                TopicArn = topicArn
            });
        }

        var bucketName = _configuration["testBucketName"];
        if (GetYesNoResponse($"\tDelete Amazon S3 bucket {bucketName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting bucket.");
            // Delete all objects in the bucket.
            var deleteList = await _s3Client.ListObjectsV2Async(new
ListObjectsV2Request()
```

```
            {
                BucketName = bucketName
            });
            await _s3Client.DeleteObjectsAsync(new DeleteObjectsRequest()
            {
                BucketName = bucketName,
                Objects = deleteList.S3Objects
                    .Select(o => new KeyVersion { Key = o.Key }).ToList()
            });
            // Now delete the bucket.
            await _s3Client.DeleteBucketAsync(new DeleteBucketRequest()
            {
                BucketName = bucketName
            });
        }

        var roleName = _configuration["roleName"];
        if (GetYesNoResponse($"\tDelete role {roleName}? (y/n)"))
        {
            Console.WriteLine($"\tDetaching policy and deleting role.");

            await _iamClient!.DetachRolePolicyAsync(new DetachRolePolicyRequest()
            {
                RoleName = roleName,
                PolicyArn = "arn:aws:iam::aws:policy/
AmazonEventBridgeFullAccess",
            });

            await _iamClient!.DeleteRoleAsync(new DeleteRoleRequest()
            {
                RoleName = roleName
            });
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Helper method to 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;
    }
}
```

Crie uma classe que envolva as EventBridge operações.

```
/// <summary>
/// Wrapper for Amazon EventBridge operations.
/// </summary>
public class EventBridgeWrapper
{
    private readonly IAmazonEventBridge _amazonEventBridge;
    private readonly ILogger<EventBridgeWrapper> _logger;

    /// <summary>
    /// Constructor for the EventBridge wrapper.
    /// </summary>
    /// <param name="amazonEventBridge">The injected EventBridge client.</param>
    /// <param name="logger">The injected logger for the wrapper.</param>
    public EventBridgeWrapper(IAmazonEventBridge amazonEventBridge,
 ILogger<EventBridgeWrapper> logger)

    {
        _amazonEventBridge = amazonEventBridge;
        _logger = logger;
    }

    /// <summary>
    /// Get the state for a rule by the rule name.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="eventBusName">The optional name of the event bus. If empty,
 uses the default event bus.</param>
    /// <returns>The state of the rule.</returns>
```

```
    public async Task<RuleState> GetRuleStateByRuleName(string ruleName, string?
eventBusName = null)
    {
        var ruleResponse = await _amazonEventBridge.DescribeRuleAsync(
            new DescribeRuleRequest()
            {
                Name = ruleName,
                EventBusName = eventBusName
            });
        return ruleResponse.State;
    }

    /// <summary>
    /// Enable a particular rule on an event bus.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableRuleByName(string ruleName)
    {
        var ruleResponse = await _amazonEventBridge.EnableRuleAsync(
            new EnableRuleRequest()
            {
                Name = ruleName
            });
        return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Disable a particular rule on an event bus.
    /// </summary
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableRuleByName(string ruleName)
    {
        var ruleResponse = await _amazonEventBridge.DisableRuleAsync(
            new DisableRuleRequest()
            {
                Name = ruleName
            });
        return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// List the rules on an event bus.
```

```
    /// </summary>
    /// <param name="eventBusArn">The optional ARN of the event bus. If empty,
uses the default event bus.</param>
    /// <returns>The list of rules.</returns>
    public async Task<List<Rule>> ListAllRulesForEventBus(string? eventBusArn =
null)
    {
        var results = new List<Rule>();
        var request = new ListRulesRequest()
        {
            EventBusName = eventBusArn
        };
        // Get all of the pages of rules.
        ListRulesResponse response;
        do
        {
            response = await _amazonEventBridge.ListRulesAsync(request);
            results.AddRange(response.Rules);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }

    /// <summary>
    /// List all of the targets matching a rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>The list of targets.</returns>
    public async Task<List<Target>> ListAllTargetsOnRule(string ruleName)
    {
        var results = new List<Target>();
        var request = new ListTargetsByRuleRequest()
        {
            Rule = ruleName
        };
        ListTargetsByRuleResponse response;
        do
        {
            response = await _amazonEventBridge.ListTargetsByRuleAsync(request);
            results.AddRange(response.Targets);
            request.NextToken = response.NextToken;
```

```
        } while (response.NextToken is not null);

        return results;
    }

    /// <summary>
    /// List names of all rules matching a target.
    /// </summary>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <returns>The list of rule names.</returns>
    public async Task<List<string>> ListAllRuleNamesByTarget(string targetArn)
    {
        var results = new List<string>();
        var request = new ListRuleNamesByTargetRequest()
        {
            TargetArn = targetArn
        };
        ListRuleNamesByTargetResponse response;
        do
        {
            response = await
 _amazonEventBridge.ListRuleNamesByTargetAsync(request);
            results.AddRange(response.RuleNames);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }

    /// <summary>
    /// Create a new event rule that triggers when an Amazon S3 object is created
 in a bucket.
    /// </summary>
    /// <param name="roleArn">The ARN of the role.</param>
    /// <param name="ruleName">The name to give the rule.</param>
    /// <param name="bucketName">The name of the bucket to trigger the event.</
param>
    /// <returns>The ARN of the new rule.</returns>
    public async Task<string> PutS3UploadRule(string roleArn, string ruleName,
 string bucketName)
    {
        string eventPattern = "{" +
                              "\"source\": [\"aws.s3\"]," +
```

```
                                    "\"detail-type\": [\"Object Created\"]," +
                                    "\"detail\": {" +
                                        "\"bucket\": {" +
                                            "\"name\": [\"" + bucketName + "\"]"
+
                                        "}" +
                                    "}" +
                                "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Example S3 upload rule for EventBridge",
                RoleArn = roleArn,
                EventPattern = eventPattern
            });

        return response.RuleArn;
    }

    /// <summary>
    /// Update an Amazon S3 object created rule with a transform on the target.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <param name="eventBusArn">Optional event bus ARN. If empty, uses the
default event bus.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> UpdateS3UploadRuleTargetWithTransform(string
ruleName, string targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        var targets = new List<Target>
        {
            new Target()
            {
                Id = targetID,
                Arn = targetArn,
                InputTransformer = new InputTransformer()
                {
                    InputPathsMap = new Dictionary<string, string>()
                    {
```

```
                        {"bucket", "$.detail.bucket.name"},
                        {"time", "$.time"}
                    },
                    InputTemplate = "\"Notification: an object was uploaded to
bucket <bucket> at <time>.\""
                }
            }
        };
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
            });
        }
        return targetID;
    }

    /// <summary>
    /// Update a custom rule with a transform on the target.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <param name="eventBusArn">Optional event bus ARN. If empty, uses the
default event bus.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> UpdateCustomRuleTargetWithTransform(string
ruleName, string targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        var targets = new List<Target>
        {
            new Target()
            {
```

```
                Id = targetID,
                Arn = targetArn,
                InputTransformer = new InputTransformer()
                {
                    InputTemplate = "\"Notification: sample event was received.
\""
                }
            }
        };
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
 {e.ErrorCode}");
            });
        }
        return targetID;
    }

    /// <summary>
    /// Add an event to the event bus that includes an email, message, and time.
    /// </summary>
    /// <param name="email">The email to use in the event detail of the custom
 event.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutCustomEmailEvent(string email)
    {
        var eventDetail = new
        {
            UserEmail = email,
            Message = "This event was generated by example code.",
            UtcTime = DateTime.UtcNow.ToString("g")
        };
        var response = await _amazonEventBridge.PutEventsAsync(
            new PutEventsRequest()
```

```
            {
                Entries = new List<PutEventsRequestEntry>()
                {
                    new PutEventsRequestEntry()
                    {
                        Source = "ExampleSource",
                        Detail = JsonSerializer.Serialize(eventDetail),
                        DetailType = "ExampleType"
                    }
                }
            });

        return response.FailedEntryCount == 0;
    }

    /// <summary>
    /// Update a rule to use a custom defined event pattern.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <returns>The ARN of the updated rule.</returns>
    public async Task<string> UpdateCustomEventPattern(string ruleName)
    {
        string customEventsPattern = "{" +
                                     "\"source\": [\"ExampleSource\"]," +
                                     "\"detail-type\": [\"ExampleType\"]" +
                                     "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Custom test rule",
                EventPattern = customEventsPattern
            });

        return response.RuleArn;
    }

    /// <summary>
    /// Add an Amazon SNS target topic to a rule.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <param name="targetArn">The ARN of the Amazon SNS target.</param>
```

```
   /// <param name="eventBusArn">The optional event bus name, uses default if
empty.</param>
   /// <returns>The ID of the target.</returns>
   public async Task<string> AddSnsTargetToRule(string ruleName, string
targetArn, string? eventBusArn = null)
   {
       var targetID = Guid.NewGuid().ToString();

       // Create the list of targets and add a new target.
       var targets = new List<Target>
       {
           new Target()
           {
               Arn = targetArn,
               Id = targetID
           }
       };

       // Add the targets to the rule.
       var response = await _amazonEventBridge.PutTargetsAsync(
           new PutTargetsRequest()
           {
               EventBusName = eventBusArn,
               Rule = ruleName,
               Targets = targets,
           });

       if (response.FailedEntryCount > 0)
       {
           response.FailedEntries.ForEach(e =>
           {
               _logger.LogError(
                   $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
           });
       }

       return targetID;
   }

   /// <summary>
   /// Delete an event rule by name.
   /// </summary>
   /// <param name="ruleName">The name of the event rule.</param>
```

```
    /// <returns>True if successful.</returns>
    public async Task<bool> RemoveAllTargetsFromRule(string ruleName)
    {
        var targetIds = new List<string>();
        var request = new ListTargetsByRuleRequest()
        {
            Rule = ruleName
        };
        ListTargetsByRuleResponse targetsResponse;
        do
        {
            targetsResponse = await
_amazonEventBridge.ListTargetsByRuleAsync(request);
            targetIds.AddRange(targetsResponse.Targets.Select(t => t.Id));
            request.NextToken = targetsResponse.NextToken;

        } while (targetsResponse.NextToken is not null);

        var removeResponse = await _amazonEventBridge.RemoveTargetsAsync(
            new RemoveTargetsRequest()
            {
                Rule = ruleName,
                Ids = targetIds
            });

        if (removeResponse.FailedEntryCount > 0)
        {
            removeResponse.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to remove target {e.TargetId}: {e.ErrorMessage},
code {e.ErrorCode}");
            });
        }

        return removeResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Delete an event rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the event rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteRuleByName(string ruleName)
```

```
    {
        var response = await _amazonEventBridge.DeleteRuleAsync(
            new DeleteRuleRequest()
            {
                Name = ruleName
            });

        return response.HttpStatusCode == HttpStatusCode.OK;
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for .NET .
  - DeleteRule
  - DescribeRule
  - DisableRule
  - EnableRule
  - ListRuleNamesByTarget
  - ListRules
  - ListTargetsByRule
  - PutEvents
  - PutRule
  - PutTargets

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
/**
 * 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
 *
 * This Java code example performs the following tasks:
 *
 * This Java V2 example performs the following tasks with Amazon EventBridge:
 *
 * 1. Creates an AWS Identity and Access Management (IAM) role to use with
 * Amazon EventBridge.
 * 2. Amazon Simple Storage Service (Amazon S3) bucket with EventBridge events
 * enabled.
 * 3. Creates a rule that triggers when an object is uploaded to Amazon S3.
 * 4. Lists rules on the event bus.
 * 5. Creates a new Amazon Simple Notification Service (Amazon SNS) topic and
 * lets the user subscribe to it.
 * 6. Adds a target to the rule that sends an email to the specified topic.
 * 7. Creates an EventBridge event that sends an email when an Amazon S3 object
 * is created.
 * 8. Lists Targets.
 * 9. Lists the rules for the same target.
 * 10. Triggers the rule by uploading a file to the Amazon S3 bucket.
 * 11. Disables a specific rule.
 * 12. Checks and print the state of the rule.
 * 13. Adds a transform to the rule to change the text of the email.
 * 14. Enables a specific rule.
 * 15. Triggers the updated rule by uploading a file to the Amazon S3 bucket.
 * 16. Updates the rule to be a custom rule pattern.
 * 17. Sending an event to trigger the rule.
 * 18. Cleans up resources.
 *
 */
public class EventbridgeMVP {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws InterruptedException,
 IOException {
        final String usage = """

                Usage:
```

```
                    <roleName> <bucketName> <topicName> <eventRuleName>

                Where:
                    roleName - The name of the role to create.
                    bucketName - The Amazon Simple Storage Service (Amazon S3)
bucket name to create.
                    topicName - The name of the Amazon Simple Notification
Service (Amazon SNS) topic to create.
                    eventRuleName - The Amazon EventBridge rule name to create.
                """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
        }

        String polJSON = "{" +
                "\"Version\": \"2012-10-17\"," +
                "\"Statement\": [{" +
                "\"Effect\": \"Allow\"," +
                "\"Principal\": {" +
                "\"Service\": \"events.amazonaws.com\"" +
                "}," +
                "\"Action\": \"sts:AssumeRole\"" +
                "}]" +
                "}";

        Scanner sc = new Scanner(System.in);
        String roleName = args[0];
        String bucketName = args[1];
        String topicName = args[2];
        String eventRuleName = args[3];

        Region region = Region.US_EAST_1;
        EventBridgeClient eventBrClient = EventBridgeClient.builder()
                .region(region)
                .build();

        S3Client s3Client = S3Client.builder()
                .region(region)
                .build();

        Region regionGl = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
```

```
                .region(regionGl)
                .build();

        SnsClient snsClient = SnsClient.builder()
                .region(region)
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon EventBridge example
scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out
                .println("1. Create an AWS Identity and Access Management (IAM)
role to use with Amazon EventBridge.");
        String roleArn = createIAMRole(iam, roleName, polJSON);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Create an S3 bucket with EventBridge events
enabled.");
        if (checkBucket(s3Client, bucketName)) {
            System.out.println("Bucket " + bucketName + " already exists. Ending
this scenario.");
            System.exit(1);
        }

        createBucket(s3Client, bucketName);
        Thread.sleep(3000);
        setBucketNotification(s3Client, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Create a rule that triggers when an object is
uploaded to Amazon S3.");
        Thread.sleep(10000);
        addEventRule(eventBrClient, roleArn, bucketName, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. List rules on the event bus.");
        listRules(eventBrClient);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("5. Create a new SNS topic for testing and let the
user subscribe to the topic.");
        String topicArn = createSnsTopic(snsClient, topicName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. Add a target to the rule that sends an email to
the specified topic.");
        System.out.println("Enter your email to subscribe to the Amazon SNS
topic:");
        String email = sc.nextLine();
        subEmail(snsClient, topicArn, email);
        System.out.println(
                "Use the link in the email you received to confirm your
subscription. Then, press Enter to continue.");
        sc.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Create an EventBridge event that sends an email
when an Amazon S3 object is created.");
        addSnsEventRule(eventBrClient, eventRuleName, topicArn, topicName,
eventRuleName, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 8. List Targets.");
        listTargets(eventBrClient, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 9. List the rules for the same target.");
        listTargetRules(eventBrClient, topicArn);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 10. Trigger the rule by uploading a file to the S3
bucket.");
        System.out.println("Press Enter to continue.");
        sc.nextLine();
        uploadTextFiletoS3(s3Client, bucketName);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("11. Disable a specific rule.");
        changeRuleState(eventBrClient, eventRuleName, false);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("12. Check and print the state of the rule.");
        checkRule(eventBrClient, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Add a transform to the rule to change the text of
the email.");
        updateSnsEventRule(eventBrClient, topicArn, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Enable a specific rule.");
        changeRuleState(eventBrClient, eventRuleName, true);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 15. Trigger the updated rule by uploading a file to
the S3 bucket.");
        System.out.println("Press Enter to continue.");
        sc.nextLine();
        uploadTextFiletoS3(s3Client, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 16. Update the rule to be a custom rule pattern.");
        updateToCustomRule(eventBrClient, eventRuleName);
        System.out.println("Updated event rule " + eventRuleName + " to use a
custom pattern.");
        updateCustomRuleTargetWithTransform(eventBrClient, topicArn,
eventRuleName);
        System.out.println("Updated event target " + topicArn + ".");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("17. Sending an event to trigger the rule. This will
trigger a subscription email.");
        triggerCustomRule(eventBrClient, email);
```

```
        System.out.println("Events have been sent. Press Enter to continue.");
        sc.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("18. Clean up resources.");
        System.out.println("Do you want to clean up resources (y/n)");
        String ans = sc.nextLine();
        if (ans.compareTo("y") == 0) {
            cleanupResources(eventBrClient, snsClient, s3Client, iam, topicArn,
eventRuleName, bucketName, roleName);
        } else {
            System.out.println("The resources will not be cleaned up. ");
        }
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Amazon EventBridge example scenario has
successfully completed.");
        System.out.println(DASHES);
    }

    public static void cleanupResources(EventBridgeClient eventBrClient,
SnsClient snsClient, S3Client s3Client,
            IamClient iam, String topicArn, String eventRuleName, String
bucketName, String roleName) {
        System.out.println("Removing all targets from the event rule.");
        deleteTargetsFromRule(eventBrClient, eventRuleName);
        deleteRuleByName(eventBrClient, eventRuleName);
        deleteSNSTopic(snsClient, topicArn);
        deleteS3Bucket(s3Client, bucketName);
        deleteRole(iam, roleName);
    }

    public static void deleteRole(IamClient iam, String roleName) {
        String policyArn = "arn:aws:iam::aws:policy/AmazonEventBridgeFullAccess";
        DetachRolePolicyRequest policyRequest = DetachRolePolicyRequest.builder()
                .policyArn(policyArn)
                .roleName(roleName)
                .build();

        iam.detachRolePolicy(policyRequest);
        System.out.println("Successfully detached policy " + policyArn + " from
role " + roleName);
```

```
        // Delete the role.
        DeleteRoleRequest roleRequest = DeleteRoleRequest.builder()
                .roleName(roleName)
                .build();

        iam.deleteRole(roleRequest);
        System.out.println("*** Successfully deleted " + roleName);
    }

    public static void deleteS3Bucket(S3Client s3Client, String bucketName) {
        // Remove all the objects from the S3 bucket.
        ListObjectsRequest listObjects = ListObjectsRequest.builder()
                .bucket(bucketName)
                .build();

        ListObjectsResponse res = s3Client.listObjects(listObjects);
        List<S3Object> objects = res.contents();
        ArrayList<ObjectIdentifier> toDelete = new ArrayList<>();

        for (S3Object myValue : objects) {
            toDelete.add(ObjectIdentifier.builder()
                    .key(myValue.key())
                    .build());
        }

        DeleteObjectsRequest dor = DeleteObjectsRequest.builder()
                .bucket(bucketName)
                .delete(Delete.builder()
                        .objects(toDelete).build())
                .build();

        s3Client.deleteObjects(dor);

        // Delete the S3 bucket.
        DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
                .bucket(bucketName)
                .build();

        s3Client.deleteBucket(deleteBucketRequest);
        System.out.println("You have deleted the bucket and the objects");
    }

    // Delete the SNS topic.
```

```
   public static void deleteSNSTopic(SnsClient snsClient, String topicArn) {
       try {
           DeleteTopicRequest request = DeleteTopicRequest.builder()
                   .topicArn(topicArn)
                   .build();

           DeleteTopicResponse result = snsClient.deleteTopic(request);
           System.out.println("\n\nStatus was " +
result.sdkHttpResponse().statusCode());

       } catch (SnsException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }

   public static void deleteRuleByName(EventBridgeClient eventBrClient, String
ruleName) {
       DeleteRuleRequest ruleRequest = DeleteRuleRequest.builder()
               .name(ruleName)
               .build();

       eventBrClient.deleteRule(ruleRequest);
       System.out.println("Successfully deleted the rule");
   }

   public static void deleteTargetsFromRule(EventBridgeClient eventBrClient,
String eventRuleName) {
       // First, get all targets that will be deleted.
       ListTargetsByRuleRequest request = ListTargetsByRuleRequest.builder()
               .rule(eventRuleName)
               .build();

       ListTargetsByRuleResponse response =
eventBrClient.listTargetsByRule(request);
       List<Target> allTargets = response.targets();

       // Get all targets and delete them.
       for (Target myTarget : allTargets) {
           RemoveTargetsRequest removeTargetsRequest =
RemoveTargetsRequest.builder()
                   .rule(eventRuleName)
                   .ids(myTarget.id())
                   .build();
```

```
            eventBrClient.removeTargets(removeTargetsRequest);
            System.out.println("Successfully removed the target");
        }
    }

    public static void triggerCustomRule(EventBridgeClient eventBrClient, String
email) {
        String json = "{" +
                "\"UserEmail\": \"" + email + "\"," +
                "\"Message\": \"This event was generated by example code.\"," +
                "\"UtcTime\": \"Now.\"" +
                "}";

        PutEventsRequestEntry entry = PutEventsRequestEntry.builder()
                .source("ExampleSource")
                .detail(json)
                .detailType("ExampleType")
                .build();

        PutEventsRequest eventsRequest = PutEventsRequest.builder()
                .entries(entry)
                .build();

        eventBrClient.putEvents(eventsRequest);
    }

    public static void updateCustomRuleTargetWithTransform(EventBridgeClient
eventBrClient, String topicArn,
            String ruleName) {
        String targetId = java.util.UUID.randomUUID().toString();
        InputTransformer inputTransformer = InputTransformer.builder()
                .inputTemplate("\"Notification: sample event was received.\"")
                .build();

        Target target = Target.builder()
                .id(targetId)
                .arn(topicArn)
                .inputTransformer(inputTransformer)
                .build();

        try {
            PutTargetsRequest targetsRequest = PutTargetsRequest.builder()
                    .rule(ruleName)
```

```
                    .targets(target)
                    .eventBusName(null)
                    .build();

            eventBrClient.putTargets(targetsRequest);
        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void updateToCustomRule(EventBridgeClient eventBrClient, String
ruleName) {
        String customEventsPattern = "{" +
                "\"source\": [\"ExampleSource\"]," +
                "\"detail-type\": [\"ExampleType\"]" +
                "}";

        PutRuleRequest request = PutRuleRequest.builder()
                .name(ruleName)
                .description("Custom test rule")
                .eventPattern(customEventsPattern)
                .build();

        eventBrClient.putRule(request);
    }

    // Update an Amazon S3 object created rule with a transform on the target.
    public static void updateSnsEventRule(EventBridgeClient eventBrClient, String
topicArn, String ruleName) {
        String targetId = java.util.UUID.randomUUID().toString();
        Map<String, String> myMap = new HashMap<>();
        myMap.put("bucket", "$.detail.bucket.name");
        myMap.put("time", "$.time");

        InputTransformer inputTransformer = InputTransformer.builder()
                .inputTemplate("\"Notification: an object was uploaded to bucket
<bucket> at <time>.\"")
                .inputPathsMap(myMap)
                .build();

        Target target = Target.builder()
                .id(targetId)
                .arn(topicArn)
```

```
                .inputTransformer(inputTransformer)
                .build();

        try {
            PutTargetsRequest targetsRequest = PutTargetsRequest.builder()
                    .rule(ruleName)
                    .targets(target)
                    .eventBusName(null)
                    .build();

            eventBrClient.putTargets(targetsRequest);

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void checkRule(EventBridgeClient eventBrClient, String
eventRuleName) {
        try {
            DescribeRuleRequest ruleRequest = DescribeRuleRequest.builder()
                    .name(eventRuleName)
                    .build();

            DescribeRuleResponse response =
eventBrClient.describeRule(ruleRequest);
            System.out.println("The state of the rule is " +
response.stateAsString());

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void changeRuleState(EventBridgeClient eventBrClient, String
eventRuleName, Boolean isEnabled) {
        try {
            if (!isEnabled) {
                System.out.println("Disabling the rule: " + eventRuleName);
                DisableRuleRequest ruleRequest = DisableRuleRequest.builder()
                        .name(eventRuleName)
                        .build();
```

```
                eventBrClient.disableRule(ruleRequest);
            } else {
                System.out.println("Enabling the rule: " + eventRuleName);
                EnableRuleRequest ruleRequest = EnableRuleRequest.builder()
                        .name(eventRuleName)
                        .build();
                eventBrClient.enableRule(ruleRequest);
            }

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // Create and upload a file to an S3 bucket to trigger an event.
    public static void uploadTextFiletoS3(S3Client s3Client, String bucketName)
throws IOException {
        // Create a unique file name.
        String fileSuffix = new SimpleDateFormat("yyyyMMddHHmmss").format(new
Date());
        String fileName = "TextFile" + fileSuffix + ".txt";

        File myFile = new File(fileName);
        FileWriter fw = new FileWriter(myFile.getAbsoluteFile());
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write("This is a sample file for testing uploads.");
        bw.close();

        try {
            PutObjectRequest putOb = PutObjectRequest.builder()
                    .bucket(bucketName)
                    .key(fileName)
                    .build();

            s3Client.putObject(putOb, RequestBody.fromFile(myFile));

        } catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

```
   public static void listTargetRules(EventBridgeClient eventBrClient, String
topicArn) {
       ListRuleNamesByTargetRequest ruleNamesByTargetRequest =
ListRuleNamesByTargetRequest.builder()
               .targetArn(topicArn)
               .build();

       ListRuleNamesByTargetResponse response =
eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest);
       List<String> rules = response.ruleNames();
       for (String rule : rules) {
           System.out.println("The rule name is " + rule);
       }
   }

   public static void listTargets(EventBridgeClient eventBrClient, String
ruleName) {
       ListTargetsByRuleRequest ruleRequest = ListTargetsByRuleRequest.builder()
               .rule(ruleName)
               .build();

       ListTargetsByRuleResponse res =
eventBrClient.listTargetsByRule(ruleRequest);
       List<Target> targetsList = res.targets();
       for (Target target: targetsList) {
           System.out.println("Target ARN: "+target.arn());
       }
   }

   // Add a rule which triggers an SNS target when a file is uploaded to an S3
   // bucket.
   public static void addSnsEventRule(EventBridgeClient eventBrClient, String
ruleName, String topicArn,
           String topicName, String eventRuleName, String bucketName) {
       String targetID = java.util.UUID.randomUUID().toString();
       Target myTarget = Target.builder()
               .id(targetID)
               .arn(topicArn)
               .build();

       List<Target> targets = new ArrayList<>();
       targets.add(myTarget);
       PutTargetsRequest request = PutTargetsRequest.builder()
               .eventBusName(null)
```

```
                .targets(targets)
                .rule(ruleName)
                .build();

        eventBrClient.putTargets(request);
        System.out.println("Added event rule " + eventRuleName + " with Amazon
SNS target " + topicName + " for bucket "
                + bucketName + ".");
    }

    public static void subEmail(SnsClient snsClient, String topicArn, String
email) {
        try {
            SubscribeRequest request = SubscribeRequest.builder()
                    .protocol("email")
                    .endpoint(email)
                    .returnSubscriptionArn(true)
                    .topicArn(topicArn)
                    .build();

            SubscribeResponse result = snsClient.subscribe(request);
            System.out.println("Subscription ARN: " + result.subscriptionArn() +
"\n\n Status is "
                    + result.sdkHttpResponse().statusCode());

        } catch (SnsException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void listRules(EventBridgeClient eventBrClient) {
        try {
            ListRulesRequest rulesRequest = ListRulesRequest.builder()
                    .eventBusName("default")
                    .limit(10)
                    .build();

            ListRulesResponse response = eventBrClient.listRules(rulesRequest);
            List<Rule> rules = response.rules();
            for (Rule rule : rules) {
                System.out.println("The rule name is : " + rule.name());
                System.out.println("The rule description is : " +
rule.description());
```

```
                System.out.println("The rule state is : " +
rule.stateAsString());
            }

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static String createSnsTopic(SnsClient snsClient, String topicName) {
        String topicPolicy = "{" +
                "\"Version\": \"2012-10-17\"," +
                "\"Statement\": [{" +
                "\"Sid\": \"EventBridgePublishTopic\"," +
                "\"Effect\": \"Allow\"," +
                "\"Principal\": {" +
                "\"Service\": \"events.amazonaws.com\"" +
                "}," +
                "\"Resource\": \"*\"," +
                "\"Action\": \"sns:Publish\"" +
                "}]" +
                "}";

        Map<String, String> topicAttributes = new HashMap<>();
        topicAttributes.put("Policy", topicPolicy);
        CreateTopicRequest topicRequest = CreateTopicRequest.builder()
                .name(topicName)
                .attributes(topicAttributes)
                .build();

        CreateTopicResponse response = snsClient.createTopic(topicRequest);
        System.out.println("Added topic " + topicName + " for email
subscriptions.");
        return response.topicArn();
    }

    // Create a new event rule that triggers when an Amazon S3 object is created
in
    // a bucket.
    public static void addEventRule(EventBridgeClient eventBrClient, String
roleArn, String bucketName,
            String eventRuleName) {
        String pattern = "{\n" +
```

```
                "  \"source\": [\"aws.s3\"],\n" +
                "  \"detail-type\": [\"Object Created\"],\n" +
                "  \"detail\": {\n" +
                "    \"bucket\": {\n" +
                "      \"name\": [\"" + bucketName + "\"]\n" +
                "    }\n" +
                "  }\n" +
                "}";

        try {
            PutRuleRequest ruleRequest = PutRuleRequest.builder()
                    .description("Created by using the AWS SDK for Java v2")
                    .name(eventRuleName)
                    .eventPattern(pattern)
                    .roleArn(roleArn)
                    .build();

            PutRuleResponse ruleResponse = eventBrClient.putRule(ruleRequest);
            System.out.println("The ARN of the new rule is " +
ruleResponse.ruleArn());

        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // Determine if the S3 bucket exists.
    public static Boolean checkBucket(S3Client s3Client, String bucketName) {
        try {
            HeadBucketRequest headBucketRequest = HeadBucketRequest.builder()
                    .bucket(bucketName)
                    .build();

            s3Client.headBucket(headBucketRequest);
            return true;
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
        return false;
    }

    // Set the S3 bucket notification configuration.
```

```
   public static void setBucketNotification(S3Client s3Client, String
bucketName) {
       try {
           EventBridgeConfiguration eventBridgeConfiguration =
EventBridgeConfiguration.builder()
                   .build();

           NotificationConfiguration configuration =
NotificationConfiguration.builder()
                   .eventBridgeConfiguration(eventBridgeConfiguration)
                   .build();

           PutBucketNotificationConfigurationRequest configurationRequest =
PutBucketNotificationConfigurationRequest
                   .builder()
                   .bucket(bucketName)
                   .notificationConfiguration(configuration)
                   .skipDestinationValidation(true)
                   .build();

           s3Client.putBucketNotificationConfiguration(configurationRequest);
           System.out.println("Added bucket " + bucketName + " with EventBridge
events enabled.");

       } catch (S3Exception e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }

   public static void createBucket(S3Client s3Client, String bucketName) {
       try {
           S3Waiter s3Waiter = s3Client.waiter();
           CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
                   .bucket(bucketName)
                   .build();

           s3Client.createBucket(bucketRequest);
           HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
                   .bucket(bucketName)
                   .build();

           // Wait until the bucket is created and print out the response.
```

```
            WaiterResponse<HeadBucketResponse> waiterResponse =
 s3Waiter.waitUntilBucketExists(bucketRequestWait);
            waiterResponse.matched().response().ifPresent(System.out::println);
            System.out.println(bucketName + " is ready");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static String createIAMRole(IamClient iam, String rolename, String
 polJSON) {
        try {
            CreateRoleRequest request = CreateRoleRequest.builder()
                    .roleName(rolename)
                    .assumeRolePolicyDocument(polJSON)
                    .description("Created using the AWS SDK for Java")
                    .build();

            CreateRoleResponse response = iam.createRole(request);
            AttachRolePolicyRequest rolePolicyRequest =
 AttachRolePolicyRequest.builder()
                    .roleName(rolename)
                    .policyArn("arn:aws:iam::aws:policy/
AmazonEventBridgeFullAccess")
                    .build();

            iam.attachRolePolicy(rolePolicyRequest);
            return response.role().arn();

        } catch (IamException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for Java 2.x .
  - DeleteRule

- DescribeRule
- DisableRule
- EnableRule
- ListRuleNamesByTarget
- ListRules
- ListTargetsByRule
- PutEvents
- PutRule
- PutTargets

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
/*
 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

 This Kotlin example performs the following tasks with Amazon EventBridge:

 1. Creates an AWS Identity and Access Management (IAM) role to use with Amazon
 EventBridge.
 2. Creates an Amazon Simple Storage Service (Amazon S3) bucket with EventBridge
 events enabled.
 3. Creates a rule that triggers when an object is uploaded to Amazon S3.
 4. Lists rules on the event bus.
 5. Creates a new Amazon Simple Notification Service (Amazon SNS) topic and lets
 the user subscribe to it.
 6. Adds a target to the rule that sends an email to the specified topic.
```

```
 7. Creates an EventBridge event that sends an email when an Amazon S3 object is
 created.
 8. Lists targets.
 9. Lists the rules for the same target.
 10. Triggers the rule by uploading a file to the S3 bucket.
 11. Disables a specific rule.
 12. Checks and prints the state of the rule.
 13. Adds a transform to the rule to change the text of the email.
 14. Enables a specific rule.
 15. Triggers the updated rule by uploading a file to the S3 bucket.
 16. Updates the rule to a custom rule pattern.
 17. Sends an event to trigger the rule.
 18. Cleans up resources.
*/
val DASHES: String = String(CharArray(80)).replace("\u0000", "-")
suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <roleName> <bucketName> <topicName> <eventRuleName>

    Where:
        roleName - The name of the role to create.
        bucketName - The Amazon Simple Storage Service (Amazon S3) bucket name to
 create.
        topicName - The name of the Amazon Simple Notification Service (Amazon
 SNS) topic to create.
        eventRuleName - The Amazon EventBridge rule name to create.
    """
    val polJSON = "{" +
        "\"Version\": \"2012-10-17\"," +
        "\"Statement\": [{" +
        "\"Effect\": \"Allow\"," +
        "\"Principal\": {" +
        "\"Service\": \"events.amazonaws.com\"" +
        "}," +
        "\"Action\": \"sts:AssumeRole\"" +
        "}]" +
        "}"

    if (args.size != 4) {
        println(usage)
        exitProcess(1)
    }
```

```
    val sc = Scanner(System.`in`)
    val roleName = args[0]
    val bucketName = args[1]
    val topicName = args[2]
    val eventRuleName = args[3]

    println(DASHES)
    println("Welcome to the Amazon EventBridge example scenario.")
    println(DASHES)

    println(DASHES)
    println("1. Create an AWS Identity and Access Management (IAM) role to use
with Amazon EventBridge.")
    val roleArn = createIAMRole(roleName, polJSON)
    println(DASHES)

    println(DASHES)
    println("2. Create an S3 bucket with EventBridge events enabled.")
    if (checkBucket(bucketName)) {
        println("$bucketName already exists. Ending this scenario.")
        exitProcess(1)
    }

    createBucket(bucketName)
    delay(3000)
    setBucketNotification(bucketName)
    println(DASHES)

    println(DASHES)
    println("3. Create a rule that triggers when an object is uploaded to Amazon
S3.")
    delay(10000)
    addEventRule(roleArn, bucketName, eventRuleName)
    println(DASHES)

    println(DASHES)
    println("4. List rules on the event bus.")
    listRules()
    println(DASHES)

    println(DASHES)
    println("5. Create a new SNS topic for testing and let the user subscribe to
the topic.")
    val topicArn = createSnsTopic(topicName)
```

```
    println(DASHES)

    println(DASHES)
    println("6. Add a target to the rule that sends an email to the specified
topic.")
    println("Enter your email to subscribe to the Amazon SNS topic:")
    val email = sc.nextLine()
    subEmail(topicArn, email)
    println("Use the link in the email you received to confirm your subscription.
Then press Enter to continue.")
    sc.nextLine()
    println(DASHES)

    println(DASHES)
    println("7. Create an EventBridge event that sends an email when an Amazon S3
object is created.")
    addSnsEventRule(eventRuleName, topicArn, topicName, eventRuleName,
bucketName)
    println(DASHES)

    println(DASHES)
    println("8. List targets.")
    listTargets(eventRuleName)
    println(DASHES)

    println(DASHES)
    println(" 9. List the rules for the same target.")
    listTargetRules(topicArn)
    println(DASHES)

    println(DASHES)
    println("10. Trigger the rule by uploading a file to the S3 bucket.")
    println("Press Enter to continue.")
    sc.nextLine()
    uploadTextFiletoS3(bucketName)
    println(DASHES)

    println(DASHES)
    println("11. Disable a specific rule.")
    changeRuleState(eventRuleName, false)
    println(DASHES)

    println(DASHES)
    println("12. Check and print the state of the rule.")
```

```
    checkRule(eventRuleName)
    println(DASHES)

    println(DASHES)
    println("13. Add a transform to the rule to change the text of the email.")
    updateSnsEventRule(topicArn, eventRuleName)
    println(DASHES)

    println(DASHES)
    println("14. Enable a specific rule.")
    changeRuleState(eventRuleName, true)
    println(DASHES)

    println(DASHES)
    println("15. Trigger the updated rule by uploading a file to the S3 bucket.")
    println("Press Enter to continue.")
    sc.nextLine()
    uploadTextFiletoS3(bucketName)
    println(DASHES)

    println(DASHES)
    println("16. Update the rule to a custom rule pattern.")
    updateToCustomRule(eventRuleName)
    println("Updated event rule $eventRuleName to use a custom pattern.")
    updateCustomRuleTargetWithTransform(topicArn, eventRuleName)
    println("Updated event target $topicArn.")
    println(DASHES)

    println(DASHES)
    println("17. Send an event to trigger the rule. This will trigger a
subscription email.")
    triggerCustomRule(email)
    println("Events have been sent. Press Enter to continue.")
    sc.nextLine()
    println(DASHES)

    println(DASHES)
    println("18. Clean up resources.")
    println("Do you want to clean up resources (y/n)")
    val ans = sc.nextLine()
    if (ans.compareTo("y") == 0) {
        cleanupResources(topicArn, eventRuleName, bucketName, roleName)
    } else {
        println("The resources will not be cleaned up. ")
```

```
    }
    println(DASHES)

    println(DASHES)
    println("The Amazon EventBridge example scenario has successfully
 completed.")
    println(DASHES)
}

suspend fun cleanupResources(topicArn: String?, eventRuleName: String?,
 bucketName: String?, roleName: String?) {
    println("Removing all targets from the event rule.")
    deleteTargetsFromRule(eventRuleName)
    deleteRuleByName(eventRuleName)
    deleteSNSTopic(topicArn)
    deleteS3Bucket(bucketName)
    deleteRole(roleName)
}

suspend fun deleteRole(roleNameVal: String?) {
    val policyArnVal = "arn:aws:iam::aws:policy/AmazonEventBridgeFullAccess"
    val policyRequest = DetachRolePolicyRequest {
        policyArn = policyArnVal
        roleName = roleNameVal
    }
    IamClient { region = "us-east-1" }.use { iam ->
        iam.detachRolePolicy(policyRequest)
        println("Successfully detached policy $policyArnVal from role
 $roleNameVal")

        // Delete the role.
        val roleRequest = DeleteRoleRequest {
            roleName = roleNameVal
        }

        iam.deleteRole(roleRequest)
        println("*** Successfully deleted $roleNameVal")
    }
}

suspend fun deleteS3Bucket(bucketName: String?) {
    // Remove all the objects from the S3 bucket.
    val listObjects = ListObjectsRequest {
        bucket = bucketName
```

```
    }
    S3Client { region = "us-east-1" }.use { s3Client ->
        val res = s3Client.listObjects(listObjects)
        val myObjects = res.contents
        val toDelete = mutableListOf<ObjectIdentifier>()

        if (myObjects != null) {
            for (myValue in myObjects) {
                toDelete.add(
                    ObjectIdentifier {
                        key = myValue.key
                    }
                )
            }
        }

        val delOb = Delete {
            objects = toDelete
        }

        val dor = DeleteObjectsRequest {
            bucket = bucketName
            delete = delOb
        }
        s3Client.deleteObjects(dor)

        // Delete the S3 bucket.
        val deleteBucketRequest = DeleteBucketRequest {
            bucket = bucketName
        }
        s3Client.deleteBucket(deleteBucketRequest)
        println("You have deleted the bucket and the objects")
    }
}

// Delete the SNS topic.
suspend fun deleteSNSTopic(topicArnVal: String?) {
    val request = DeleteTopicRequest {
        topicArn = topicArnVal
    }

    SnsClient { region = "us-east-1" }.use { snsClient ->
        snsClient.deleteTopic(request)
        println(" $topicArnVal was deleted.")
```

```
    }
}

suspend fun deleteRuleByName(ruleName: String?) {
    val ruleRequest = DeleteRuleRequest {
        name = ruleName
    }
    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.deleteRule(ruleRequest)
        println("Successfully deleted the rule")
    }
}

suspend fun deleteTargetsFromRule(eventRuleName: String?) {
    // First, get all targets that will be deleted.
    val request = ListTargetsByRuleRequest {
        rule = eventRuleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(request)
        val allTargets = response.targets

        // Get all targets and delete them.
        if (allTargets != null) {
            for (myTarget in allTargets) {
                val removeTargetsRequest = RemoveTargetsRequest {
                    rule = eventRuleName
                    ids = listOf(myTarget.id.toString())
                }
                eventBrClient.removeTargets(removeTargetsRequest)
                println("Successfully removed the target")
            }
        }
    }
}

suspend fun triggerCustomRule(email: String) {
    val json = "{" +
        "\"UserEmail\": \"" + email + "\"," +
        "\"Message\": \"This event was generated by example code.\"" +
        "\"UtcTime\": \"Now.\"" +
        "}"
```

```
    val entry = PutEventsRequestEntry {
        source = "ExampleSource"
        detail = json
        detailType = "ExampleType"
    }

    val eventsRequest = PutEventsRequest {
        this.entries = listOf(entry)
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putEvents(eventsRequest)
    }
}

suspend fun updateCustomRuleTargetWithTransform(topicArn: String?, ruleName:
 String?) {
    val targetId = UUID.randomUUID().toString()

    val inputTransformerOb = InputTransformer {
        inputTemplate = "\"Notification: sample event was received.\""
    }

    val target = Target {
        id = targetId
        arn = topicArn
        inputTransformer = inputTransformerOb
    }

    val targetsRequest = PutTargetsRequest {
        rule = ruleName
        targets = listOf(target)
        eventBusName = null
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(targetsRequest)
    }
}

suspend fun updateToCustomRule(ruleName: String?) {
    val customEventsPattern = "{" +
        "\"source\": [\"ExampleSource\"]," +
        "\"detail-type\": [\"ExampleType\"]" +
```

```
        "}"
    val request = PutRuleRequest {
        name = ruleName
        description = "Custom test rule"
        eventPattern = customEventsPattern
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putRule(request)
    }
}

// Update an Amazon S3 object created rule with a transform on the target.
suspend fun updateSnsEventRule(topicArn: String?, ruleName: String?) {
    val targetId = UUID.randomUUID().toString()
    val myMap = mutableMapOf<String, String>()
    myMap["bucket"] = "$.detail.bucket.name"
    myMap["time"] = "$.time"

    val inputTransOb = InputTransformer {
        inputTemplate = "\"Notification: an object was uploaded to bucket
 <bucket> at <time>.\""
        inputPathsMap = myMap
    }
    val targetOb = Target {
        id = targetId
        arn = topicArn
        inputTransformer = inputTransOb
    }

    val targetsRequest = PutTargetsRequest {
        rule = ruleName
        targets = listOf(targetOb)
        eventBusName = null
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(targetsRequest)
    }
}

suspend fun checkRule(eventRuleName: String?) {
    val ruleRequest = DescribeRuleRequest {
        name = eventRuleName
```

```
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.describeRule(ruleRequest)
        println("The state of the rule is $response")
    }
}

suspend fun changeRuleState(eventRuleName: String, isEnabled: Boolean?) {
    if (!isEnabled!!) {
        println("Disabling the rule: $eventRuleName")
        val ruleRequest = DisableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.disableRule(ruleRequest)
        }
    } else {
        println("Enabling the rule: $eventRuleName")
        val ruleRequest = EnableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.enableRule(ruleRequest)
        }
    }
}

// Create and upload a file to an S3 bucket to trigger an event.
@Throws(IOException::class)
suspend fun uploadTextFiletoS3(bucketName: String?) {
    val fileSuffix = SimpleDateFormat("yyyyMMddHHmmss").format(Date())
    val fileName = "TextFile$fileSuffix.txt"
    val myFile = File(fileName)
    val fw = FileWriter(myFile.absoluteFile)
    val bw = BufferedWriter(fw)
    bw.write("This is a sample file for testing uploads.")
    bw.close()

    val putOb = PutObjectRequest {
        bucket = bucketName
        key = fileName
        body = myFile.asByteStream()
    }
```

```
    S3Client { region = "us-east-1" }.use { s3Client ->
        s3Client.putObject(putOb)
    }
}

suspend fun listTargetRules(topicArnVal: String?) {
    val ruleNamesByTargetRequest = ListRuleNamesByTargetRequest {
        targetArn = topicArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response =
 eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest)
        response.ruleNames?.forEach { rule ->
            println("The rule name is $rule")
        }
    }
}

suspend fun listTargets(ruleName: String?) {
    val ruleRequest = ListTargetsByRuleRequest {
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(ruleRequest)
        response.targets?.forEach { target ->
            println("Target ARN: ${target.arn}")
        }
    }
}

// Add a rule that triggers an SNS target when a file is uploaded to an S3
 bucket.
suspend fun addSnsEventRule(ruleName: String?, topicArn: String?, topicName:
 String, eventRuleName: String, bucketName: String) {
    val targetID = UUID.randomUUID().toString()
    val myTarget = Target {
        id = targetID
        arn = topicArn
    }

    val targetsOb = mutableListOf<Target>()
```

```
    targetsOb.add(myTarget)

    val request = PutTargetsRequest {
        eventBusName = null
        targets = targetsOb
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(request)
        println("Added event rule $eventRuleName with Amazon SNS target
 $topicName for bucket $bucketName.")
    }
}

suspend fun subEmail(topicArnVal: String?, email: String?) {
    val request = SubscribeRequest {
        protocol = "email"
        endpoint = email
        returnSubscriptionArn = true
        topicArn = topicArnVal
    }

    SnsClient { region = "us-east-1" }.use { snsClient ->
        val result = snsClient.subscribe(request)
        println(" Subscription ARN: ${result.subscriptionArn}")
    }
}

suspend fun createSnsTopic(topicName: String): String? {
    val topicPolicy = "{" +
        "\"Version\": \"2012-10-17\"," +
        "\"Statement\": [{" +
        "\"Sid\": \"EventBridgePublishTopic\"," +
        "\"Effect\": \"Allow\"," +
        "\"Principal\": {" +
        "\"Service\": \"events.amazonaws.com\"" +
        "}," +
        "\"Resource\": \"*\"," +
        "\"Action\": \"sns:Publish\"" +
        "}]" +
        "}"

    val topicAttributes = mutableMapOf<String, String>()
```

```
    topicAttributes["Policy"] = topicPolicy

    val topicRequest = CreateTopicRequest {
        name = topicName
        attributes = topicAttributes
    }

    SnsClient { region = "us-east-1" }.use { snsClient ->
        val response = snsClient.createTopic(topicRequest)
        println("Added topic $topicName for email subscriptions.")
        return response.topicArn
    }
}

suspend fun listRules() {
    val rulesRequest = ListRulesRequest {
        eventBusName = "default"
        limit = 10
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listRules(rulesRequest)
        response.rules?.forEach { rule ->
            println("The rule name is ${rule.name}")
            println("The rule ARN is ${rule.arn}")
        }
    }
}

// Create a new event rule that triggers when an Amazon S3 object is created in a
 bucket.
suspend fun addEventRule(roleArnVal: String?, bucketName: String, eventRuleName:
 String?) {
    val pattern = """{
        "source": ["aws.s3"],
        "detail-type": ["Object Created"],
        "detail": {
        "bucket": {
            "name": ["$bucketName"]
            }
        }
    }"""

    val ruleRequest = PutRuleRequest {
```

```
        description = "Created by using the AWS SDK for Kotlin"
        name = eventRuleName
        eventPattern = pattern
        roleArn = roleArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val ruleResponse = eventBrClient.putRule(ruleRequest)
        println("The ARN of the new rule is ${ruleResponse.ruleArn}")
    }
}

// Set the Amazon S3 bucket notification configuration.
suspend fun setBucketNotification(bucketName: String) {
    val eventBridgeConfig = EventBridgeConfiguration {
    }

    val configuration = NotificationConfiguration {
        eventBridgeConfiguration = eventBridgeConfig
    }

    val configurationRequest = PutBucketNotificationConfigurationRequest {
        bucket = bucketName
        notificationConfiguration = configuration
        skipDestinationValidation = true
    }

    S3Client { region = "us-east-1" }.use { s3Client ->
        s3Client.putBucketNotificationConfiguration(configurationRequest)
        println("Added bucket $bucketName with EventBridge events enabled.")
    }
}

// Create an S3 bucket using a waiter.
suspend fun createBucket(bucketName: String) {
    val request = CreateBucketRequest {
        bucket = bucketName
    }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.createBucket(request)
        s3.waitUntilBucketExists {
            bucket = bucketName
        }
```

```
        println("$bucketName is ready")
    }
}

suspend fun checkBucket(bucketName: String?): Boolean {
    try {
        // Determine if the S3 bucket exists.
        val headBucketRequest = HeadBucketRequest {
            bucket = bucketName
        }

        S3Client { region = "us-east-1" }.use { s3Client ->
            s3Client.headBucket(headBucketRequest)
            return true
        }
    } catch (e: S3Exception) {
        System.err.println(e.message)
    }
    return false
}

suspend fun createIAMRole(rolenameVal: String?, polJSON: String?): String? {
    val request = CreateRoleRequest {
        roleName = rolenameVal
        assumeRolePolicyDocument = polJSON
        description = "Created using the AWS SDK for Kotlin"
    }

    val rolePolicyRequest = AttachRolePolicyRequest {
        roleName = rolenameVal
        policyArn = "arn:aws:iam::aws:policy/AmazonEventBridgeFullAccess"
    }

    IamClient { region = "us-east-1" }.use { iam ->
        val response = iam.createRole(request)
        iam.attachRolePolicy(rolePolicyRequest)
        return response.role?.arn
    }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Kotlin.

- DeleteRule
- DescribeRule
- DisableRule
- EnableRule
- ListRuleNamesByTarget
- ListRules
- ListTargetsByRule
- PutEvents
- PutRule
- PutTargets

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de vários serviços para EventBridge usar SDKs AWS

Os aplicativos de exemplo a seguir usam AWS SDKs para combinar EventBridge com outros Serviços da AWS. Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções sobre como configurar e executar o aplicativo.

Exemplos

- Usar eventos programados para chamar uma função do Lambda

Usar eventos programados para chamar uma função do Lambda

Os exemplos de código a seguir mostram como criar uma AWS Lambda função invocada por um evento EventBridge agendado pela Amazon.

Java

SDK para Java 2.x

Mostra como criar um evento EventBridge programado pela Amazon que invoca uma AWS Lambda função. Configure EventBridge para usar uma expressão cron para agendar quando

a função Lambda é invocada. Neste exemplo, você cria uma função do Lambda usando a API de runtime de Java do Lambda. Este exemplo invoca AWS serviços diferentes para realizar um caso de uso específico. Este exemplo mostra como criar uma aplicação que envia uma mensagem de texto móvel para seus funcionários que os parabeniza na data de aniversário de um ano.

Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Serviços usados neste exemplo

- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

JavaScript

SDK para JavaScript (v3)

Mostra como criar um evento EventBridge programado pela Amazon que invoca uma AWS Lambda função. Configure EventBridge para usar uma expressão cron para agendar quando a função Lambda é invocada. Neste exemplo, você cria uma função Lambda usando a API de tempo de execução do JavaScript Lambda. Este exemplo invoca AWS serviços diferentes para realizar um caso de uso específico. Este exemplo mostra como criar uma aplicação que envia uma mensagem de texto móvel para seus funcionários que os parabeniza na data de aniversário de um ano.

Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Esse exemplo também está disponível no Guia do desenvolvedor do AWS SDK for JavaScript v3.

Serviços usados neste exemplo

- DynamoDB
- EventBridge
- Lambda

- Amazon SNS

Python

SDK para Python (Boto3)

Este exemplo mostra como registrar uma AWS Lambda função como alvo de um EventBridge evento programado da Amazon. O manipulador do Lambda grava uma mensagem amigável e os dados completos do evento no Amazon CloudWatch Logs para recuperação posterior.

- Implanta uma função do Lambda.
- Cria um evento EventBridge agendado e torna a função Lambda o alvo.
- Concede permissão para permitir a EventBridge invocação da função Lambda.
- Imprime os dados mais recentes do CloudWatch Logs para mostrar o resultado das invocações programadas.
- Limpa todos os recursos criados durante a demonstração.

Este exemplo é melhor visualizado em GitHub. Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Serviços utilizados neste exemplo

- CloudWatch Registros
- EventBridge
- Lambda

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando EventBridge com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Amazon EventBridge segurança

Amazon EventBridge usa AWS Identity and Access Management para controlar o acesso a outros AWS serviços e recursos. Para obter uma visão geral de como o IAM funciona, consulte Visão geral do gerenciamento de acesso no Guia do usuário do IAM. Para obter uma visão geral das credenciais de segurança, consulte Credenciais de segurança da AWS na Referência geral da Amazon Web Services.

Tópicos

Proteção de dados na Amazon EventBridge

O modelo de responsabilidade AWS compartilhada modelo se aplica à proteção de dados em Amazon EventBridge. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os 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 AWS postagem do blog Shared Responsibility Model and GDPR no AWS Blog de segurança da.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou 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 recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão 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 AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint 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 inclui quando você trabalha com EventBridge ou Serviços da AWS usa o console, a API ou AWS os SDKs. AWS CLI 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 de dados para EventBridge barramentos de eventos

EventBridge fornece criptografia em repouso e criptografia em trânsito para proteger seus dados de eventos:

- Criptografia inativa

  EventBridge integra-se com AWS Key Management Service (KMS) para criptografar dados de eventos armazenados em barramentos de eventos. Por padrão, EventBridge usa an Chave pertencente à AWS para criptografar dados de eventos. Você também pode especificar EventBridge para usar um chave gerenciada pelo cliente para eventos personalizados e de parceiros.
- Criptografia em trânsito

  EventBridge criptografa dados que passam entre EventBridge outros serviços usando o Transport Layer Security (TLS). Para barramentos de eventos, isso inclui durante o envio de um evento EventBridge, bem como quando EventBridge envia um evento para um destino de regra.

Criptografia em repouso para ônibus de eventos

EventBridge fornece criptografia transparente do lado do servidor por meio da integração com AWS Key Management Service (KMS). A criptografia de dados em repouso por padrão ajuda a reduzir a sobrecarga operacional e a complexidade envolvidas na proteção de dados confidenciais. Ao mesmo tempo, ela permite que você crie aplicações seguras que atendam aos rigorosos requisitos regulatórios e de conformidade de criptografia.

A EventBridge criptografia de dados do barramento de eventos em repouso inclui:

- Dados de eventos para AWSeventos personalizados e de parceiros.

  Para ônibus de eventos, os dados do evento incluem todos os campos contidos no ??? elemento do evento.

  EventBridge não criptografa os metadados do evento. Para obter mais informações sobre metadados de eventos, consulte???.
- Padrões de eventos

- Transformadores de entrada

Por padrão, EventBridge usa an Chave pertencente à AWS para criptografar dados de eventos. Você também pode especificar EventBridge para usar um chave gerenciada pelo cliente para eventos personalizados e de parceiros.

Considerações de segurança para criptografia de barramento de eventos

É altamente recomendável que você nunca coloque informações confidenciais ou sigilosas nos seguintes campos, pois elas não são criptografadas em repouso:

- Nomes de ônibus de eventos
- Nomes de regras
- Recursos compartilhados, como tags

KMS key opções para criptografia de barramento de eventos

EventBridge usa um Chave pertencente à AWS para criptografar eventos AWS de serviço armazenados em barramentos de eventos.

Para cada barramento de eventos, você pode escolher o tipo de KMS key EventBridge uso para criptografar eventos personalizados e de parceiros armazenados nesse barramento:

- Chave pertencente à AWS

  Por padrão, EventBridge criptografa dados usando o Advanced Encryption Standard (AES-256) de 256 bits sob um Chave pertencente à AWS, o que ajuda a proteger seus dados contra acesso não autorizado.

  Você não pode visualizar, gerenciar Chaves pertencentes à AWS, usar ou auditar seu uso. No entanto, não é necessário tomar nenhuma medida nem alterar qualquer programa para proteger as chaves que criptografam seus dados.

  Em geral, a menos que você precise auditar ou controlar a chave de criptografia que protege seus recursos, uma Chave pertencente à AWS é uma boa escolha. Chaves pertencentes à AWS são totalmente gratuitos (sem taxas mensais ou taxas de uso) e não contam nas AWS KMS cotas da sua conta. Você não precisa criar ou manter a chave ou sua política de chave.

Para obter mais informações, consulte AWS owned keys (chaves de propriedade da) no AWS Key Management Service Guia do Desenvolvedor.

- Chave gerenciada pelo cliente

  EventBridge suporta o uso de uma simétrica chave gerenciada pelo cliente que você cria, possui e gerencia. Como você tem controle total desse tipo de KMS key, você pode realizar tarefas como:

  - Estabelecer e manter as políticas de chave
  - Estabelecer e manter subsídios e políticas do IAM
  - Habilitar e desabilitar políticas de chaves
  - Alternar os materiais de criptografia de chaves
  - Adicionar etiquetas
  - Criar aliases de chaves
  - Chaves de agendamento para exclusão

  Para obter mais informações, consulte Chaves gerenciadas pelo cliente no AWS Key Management Service Guia do desenvolvedor.

  EventBridge suporta chaves multirregionais e acesso a chaves entre contas.

  Chaves gerenciadas pelo cliente incorrem em uma taxa mensal. Para obter detalhes, consulte AWS Key Management Service Preços e cotas no Guia do AWS Key Management Service desenvolvedor.

  > **Note**
  >
  > EventBridge não oferece suporte aos seguintes recursos em barramentos de eventos criptografados usando chaves gerenciadas pelo cliente:
  >
  > - Arquivos
  > - Descoberta do esquema

  Para mais informações, consulte ???.

Criptografando eventos com chaves gerenciadas pelo cliente

Você pode especificar que EventBridge use a AWS KMS chave gerenciada pelo cliente para criptografar seus dados (eventos personalizados e de parceiros) armazenados em um barramento de eventos, em vez de usar um Chave pertencente à AWS como padrão. Você pode especificar um chave gerenciada pelo cliente ao criar ou atualizar um barramento de eventos. Você também pode atualizar o barramento de eventos padrão para usar um chave gerenciada pelo cliente para eventos personalizados e de parceiros. Para ter mais informações, consulte ???.

Se você especificar um chave gerenciada pelo cliente para um barramento de eventos, terá a opção de especificar uma fila de mensagens mortas (DLQ) para o barramento de eventos. EventBridge em seguida, entrega quaisquer eventos personalizados ou de parceiros que gerem erros de criptografia ou decodificação para essa DLQ. Para ter mais informações, consulte ???.

Especificando um chave gerenciada pelo cliente para criptografia ao criar um barramento de eventos (usando o console)

- Siga estas instruções:

  ???.

Especificando um chave gerenciada pelo cliente para criptografia ao criar um barramento de eventos (usando a CLI)

- Ao ligar`create-event-bus`, use a `kms-key-identifier` opção para especificar o chave gerenciada pelo cliente for EventBridge a ser usado para criptografia no barramento de eventos.

  Opcionalmente, use `dead-letter-config` para especificar uma fila de cartas mortas (DLQ).

Atualizando um barramento de eventos para usar um chave gerenciada pelo cliente para criptografia (usando o console)

- Siga estas instruções:

  ???.

Atualizando um barramento de eventos para usar um chave gerenciada pelo cliente para criptografia (usando a CLI)

- Ao ligarupdate-event-bus, use a `kms-key-identifier` opção para especificar o chave gerenciada pelo cliente for EventBridge a ser usado para criptografia no barramento de eventos.

  Opcionalmente, use `dead-letter-config` para especificar uma fila de cartas mortas (DLQ).

Atualizando o barramento de eventos padrão para usar um chave gerenciada pelo cliente para criptografia usando CloudFormation

Como EventBridge provisiona o barramento de eventos padrão em sua conta automaticamente, você não pode criá-lo usando um CloudFormation modelo, como faria normalmente com qualquer recurso que desejasse incluir em uma CloudFormation pilha. Para incluir o barramento de eventos padrão em uma CloudFormation pilha, você deve primeiro importá-lo para uma pilha. Depois de importar o barramento de eventos padrão para uma pilha, você pode atualizar as propriedades do barramento de eventos conforme desejado.

- Siga estas instruções:

  ???.

Autorizando o uso EventBridge de um chave gerenciada pelo cliente

Se você usa um chave gerenciada pelo cliente em sua conta para proteger seu ônibus de EventBridge eventos, as políticas sobre isso KMS key devem dar EventBridge permissão para usá-lo em seu nome. Você fornece essas permissões em uma política importante.

EventBridge não precisa de autorização adicional para usar o padrão Chave pertencente à AWS para proteger os EventBridge recursos em sua AWS conta.

EventBridge requer as seguintes permissões em um chaves gerenciadas pelo cliente:

- `kms:DescribeKey`

  EventBridge requer essa permissão para recuperar o KMS key ARN do ID de chave fornecido e para verificar se a chave é simétrica.
- `kms:GenerateDataKey`

EventBridge requer essa permissão para gerar uma chave de dados como chave de criptografia para os dados do evento.

- `kms:Decrypt`

  EventBridge requer essa permissão para descriptografar a chave de dados que é criptografada e armazenada com os dados criptografados do evento.

  EventBridge usa isso para correspondência de regras; os usuários nunca têm acesso aos dados.

O exemplo de política de chaves a seguir fornece as permissões necessárias:

```
{
  "Sid": "Allow EventBridge to encrypt events",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": [
    "kms:DescribeKey",
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ]
  "Resource": "*",
  "Condition": {
    "StringEquals": {
        "kms:EncryptionContext:aws:events:event-bus:arn":
 "arn:aws:events:region:account-id:event-bus/event-bus-arn",
        "aws:SourceArn": "arn:aws:events:region:account-id:event-bus/event-bus-name"
    }
  }
}
```

Segurança ao usar chaves gerenciadas pelo cliente para criptografia de barramento de EventBridge eventos

Como prática recomendada de segurança, adicione uma `aws:SourceArn` chave de `kms:EncryptionContext:aws:events:event-bus:arn` condição ou à política de AWS KMS chaves. `aws:sourceAccount` A chave de condição IAM global ajuda a garantir que EventBridge use a chave KMS somente para o barramento ou conta especificada.

O exemplo a seguir demonstra como seguir essa prática recomendada em sua IAM política:

```
{
      "Sid": "Allow the use of key",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition" : {
        "StringEquals": {
          "aws:SourceAccount": "arn:aws:events:region:account-id",
          "aws:SourceArn": "arn:aws:events:region:account-id:event-bus/event-bus-name",
          "kms:EncryptionContext:aws:events:event-bus:arn":
 "arn:aws:events:region:account-id:event-bus/event-bus-arn"
        }
      }
```

Gerenciamento da chaves gerenciadas pelo cliente criptografia EventBridge do barramento de eventos

Para garantir que EventBridge sempre tenha acesso ao necessário chave gerenciada pelo cliente:

- Não exclua um chave gerenciada pelo cliente até ter certeza de que todos os eventos criptografados com ele foram processados.

  Ao realizar qualquer uma das operações a seguir, retenha o material da chave anterior para garantir que EventBridge possa continuar a usá-lo para eventos previamente criptografados:

  - Alternância automática de chaves
  - Rotação manual da chave
  - Atualização de um alias de chave

  Em geral, se você estiver pensando em excluir uma AWS KMS chave, desative-a primeiro e defina um CloudWatch alarme ou mecanismo semelhante para garantir que você nunca precise usar a chave para descriptografar dados criptografados.
- Não exclua a política de chaves que fornece EventBridge as permissões para usar a chave.

Outras considerações incluem:

- Especifique chaves gerenciadas pelo cliente os alvos das regras, conforme apropriado.

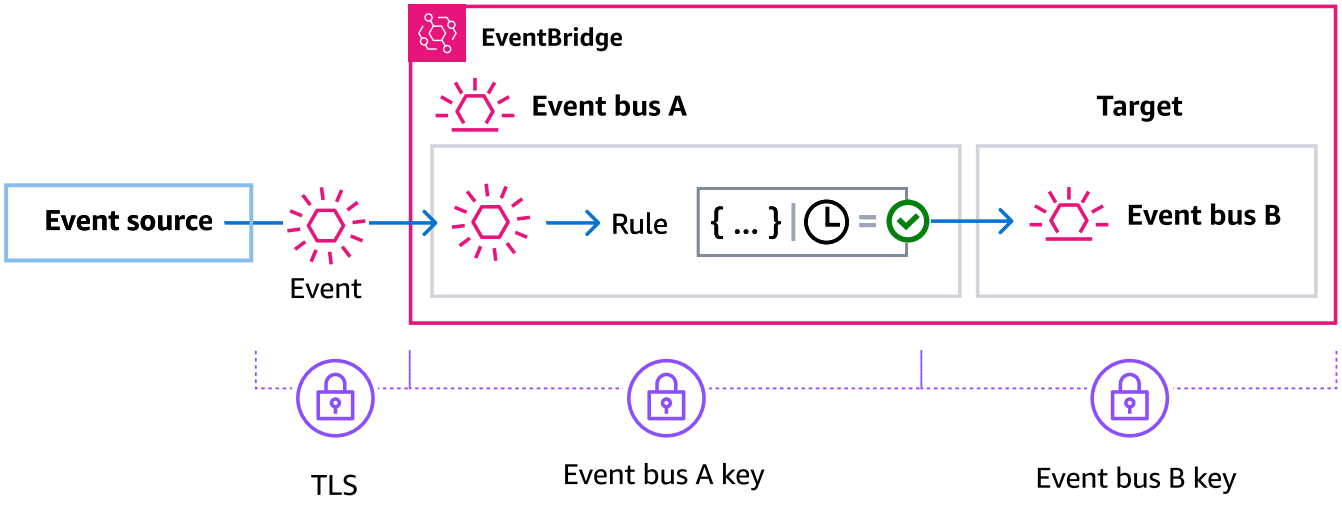
  Quando EventBridge envia um evento para um destino de regra, o evento é enviado usando Transport Layer Security (TLS). No entanto, a criptografia aplicada ao evento quando ele é armazenado no destino depende da criptografia que você configurou no próprio destino.

Criptografia de eventos quando um barramento de eventos é o alvo da regra

Quando um evento personalizado ou de parceiro é enviado para um barramento de eventos, EventBridge criptografa esse evento de acordo com a configuração da chave KMS de criptografia em repouso desse barramento de eventos - seja a padrão Chave pertencente à AWS ou a chave gerenciada pelo cliente, se tiver sido especificada. Se um evento corresponder a uma regra, EventBridge criptografa o evento com a configuração da chave KMS desse barramento de eventos até que o evento seja enviado ao destino da regra, a menos que o destino da regra seja outro barramento de eventos.

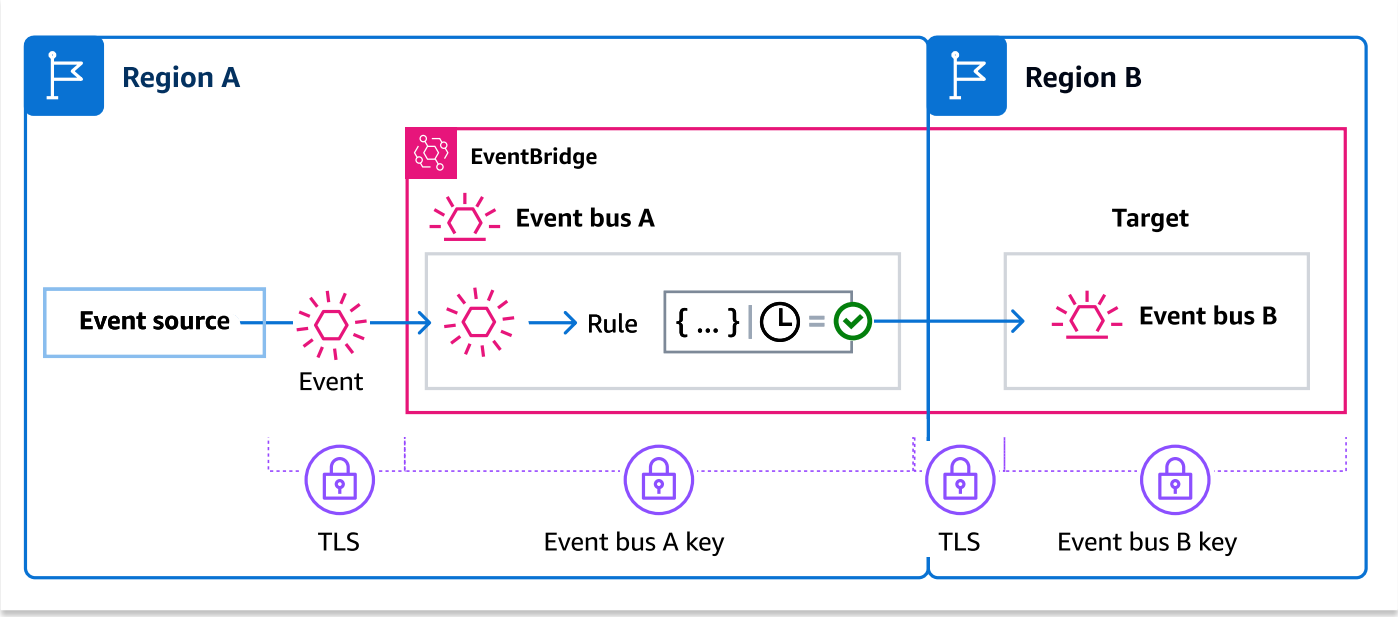
- Se o destino de uma regra for outro barramento de eventos na mesma AWS região:

  Se o barramento de eventos de destino tiver um especificado chave gerenciada pelo cliente, EventBridge criptografará o evento com o chave gerenciada pelo cliente do barramento de eventos de destino para entrega.



- Se o destino de uma regra for outro barramento de eventos em uma AWS região diferente:

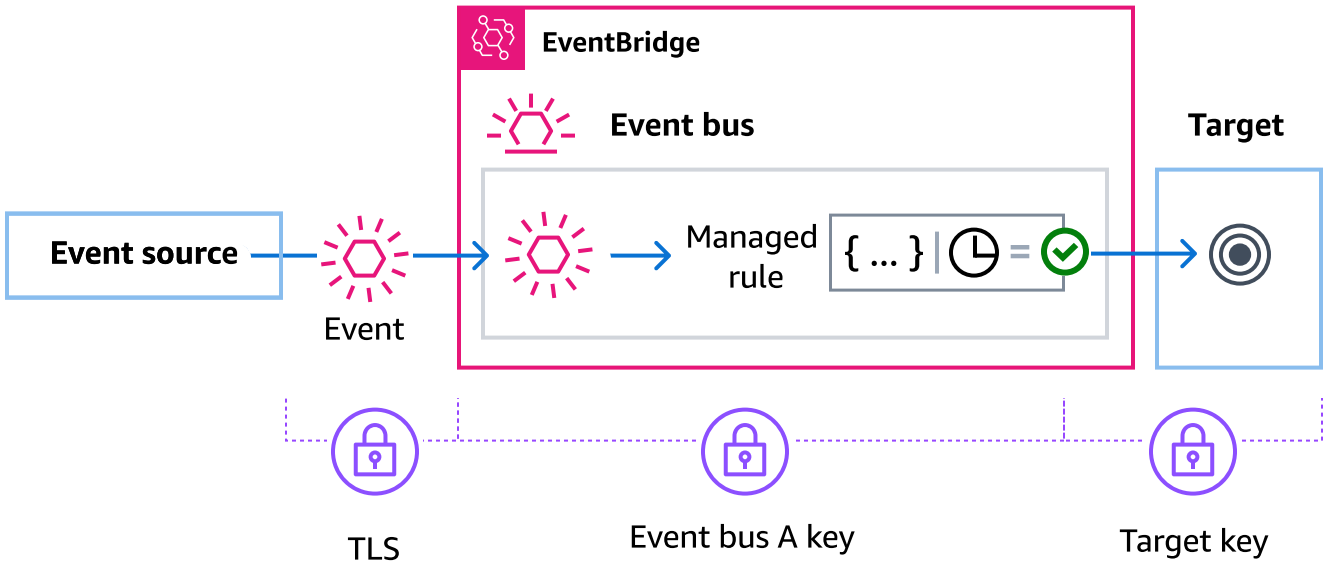
EventBridge criptografa o evento em repouso de acordo com a configuração da chave KMS no primeiro barramento de eventos. EventBridge usa TLS para enviar o evento para o segundo barramento de eventos na região diferente, onde ele é criptografado de acordo com a configuração da chave KMS especificada para o barramento de eventos de destino.



Criptografia de eventos para regras gerenciadas

AWS os serviços podem criar e gerenciar regras de barramento de eventos em sua AWS conta que são necessárias para determinadas funções nesses serviços. Como parte de uma regra gerenciada, o AWS serviço pode especificar o EventBridge uso chave gerenciada pelo cliente especificado para o destino da regra. Isso lhe dá a flexibilidade de especificar qual chave gerenciada pelo cliente usar com base na meta da regra.

Nesses casos, quando um evento personalizado ou de parceiro coincide com a regra gerenciada, EventBridge usa o destino chave gerenciada pelo cliente especificado pela regra gerenciada para criptografar o evento até que ele seja enviado ao destino da regra. Isso ocorre independentemente de o barramento de eventos ter sido configurado para usar seu próprio barramento chave gerenciada pelo cliente para criptografia. Esse é o caso mesmo se o destino da regra gerenciada for outro barramento de eventos e esse barramento de eventos tiver seu próprio barramento chave gerenciada pelo cliente especificado para criptografia. EventBridge continua usando o destino chave gerenciada pelo cliente especificado na regra gerenciada até que o evento seja enviado para um destino que não seja um barramento de eventos.



Nos casos em que o destino da regra é um barramento de eventos em outra região, você deve fornecer uma chave multirregional. O barramento de eventos na primeira região criptografa o evento usando o chave gerenciada pelo cliente especificado na regra gerenciada. Em seguida, ele envia o evento para o barramento de eventos de destino na segunda região. Esse barramento de eventos deve ser capaz de continuar usando o chave gerenciada pelo cliente até enviar o evento para seu destino.

EventBridge contexto de criptografia do barramento de eventos

Um contexto de criptografia é um conjunto de pares de chave-valor que contêm dados arbitrários não secretos. Quando você inclui um contexto de criptografia em uma solicitação para criptografar dados, o AWS KMS vincula de forma criptográfica o contexto de criptografia aos dados criptografados. Para descriptografar os dados, você deve passar o mesmo contexto de criptografia.

Você também pode usar o contexto de criptografia como condição para autorização em políticas e concessões.

Para barramentos de eventos, EventBridge usa o mesmo contexto de criptografia em todas as operações AWS KMS criptográficas. Se você usar uma chave gerenciada pelo cliente para proteger seus EventBridge recursos, poderá usar o contexto de criptografia para identificar o uso da chave KMS key nos registros e registros de auditoria. Ele também é exibido em texto simples em logs, como AWS CloudTrail e Amazon CloudWatch Logs.

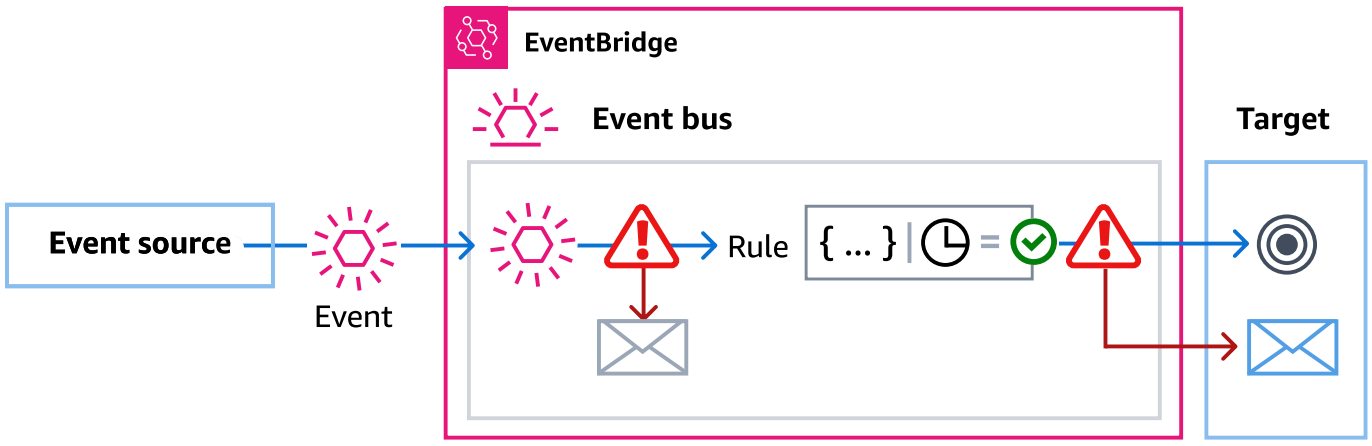
Em suas solicitações para AWS KMS, EventBridge usa um contexto de criptografia com um único par chave-valor, que contém o ARN do barramento de eventos:

```
"encryptionContext": {
    "kms:EncryptionContext:aws:events:event-bus:arn": "event-bus-arn"
}
```

Usando filas de mensagens mortas para capturar erros de eventos criptografados

Se você configurar a chave gerenciada pelo cliente criptografia em um barramento de eventos, recomendamos que você especifique uma fila de mensagens mortas (DLQ) para esse barramento de eventos. EventBridge envia eventos personalizados e de parceiros para essa DLQ se ela encontrar um erro não recuperável ao processar o evento no barramento de eventos. Um erro não recuperável é aquele em que a ação do usuário é necessária para resolver o problema subjacente, como o especificado chave gerenciada pelo cliente estar desativado ou ausente.

- Se ocorrer um erro de criptografia ou descriptografia não recuperável durante EventBridge o processamento do evento no barramento de eventos, o evento será enviado à DLQ para o barramento de eventos, se houver um especificado.
- Se ocorrer um erro de criptografia ou descriptografia não recuperável durante EventBridge a tentativa de enviar o evento para um destino, o evento será enviado ao DLQ do destino, se houver um especificado.



Para obter mais informações, incluindo considerações sobre o uso de DLQs e instruções sobre como definir permissões, consulte. ???

Descriptografando eventos em filas de mensagens mortas EventBridge

Depois de resolver o problema subjacente que está causando um erro não recuperável, você pode processar os eventos enviados para o barramento de eventos ou para as DLQs de destino. Para eventos criptografados, você deve primeiro descriptografar o evento para processá-lo.

O exemplo a seguir demonstra como descriptografar um evento que EventBridge foi entregue a um barramento de eventos ou DLQ de destino.

```
// You will receive an encrypted event in the following json format.
        // ```
        //   {
        //     "version": "0",
        //     "id": "053afa53-cdd7-285b-e754-b0dfd0ac0bfb",  // New event id not the
 same as the original one
        //     "account": "123456789012",
        //     "time": "2020-02-10T10:22:00Z",
        //     "resources": [ ],
        //     "region": "us-east-1",
        //     "source": "aws.events",
        //     "detail-type": "Encrypted Events",
        //     "detail": {
        //       "event-bus-arn": "arn:aws:events:region:account:event-bus/bus-name",
        //       "rule-arn": "arn:aws:events:region:account:event-bus/bus-name/rule-
name",
        //       "kms-key-arn": "arn:aws:kms:region:account:key/key-arn",
        //       "encrypted-payload": "AgR4qiru/XNwTUyCgRHqP7rbbHn/
xpmVeVeRIAd12TDYYVwAawABABRhd3M6ZXZlbnRzOmV2ZW50LWJ1cwB
        //
 RYXJuOmF3czpldmVudHM6dXMtZWFzdC0xOjE0NjY4NjkwNDY3MzpldmVudC1idXMvY21rbXMtZ2EtY3Jvc3
        //
 MtYWNjb3VudC1zb3VyY2UtYnVzAAEAB2F3cy1rbXMAS2Fybjphd3M6a21zOnVzLWVhc3QtMToxNDY2ODY5"
        //     }
        //   }
        // ```

        // Construct an AwsCrypto object with the encryption algorithm
 `ALG_AES_256_GCM_HKDF_SHA512_COMMIT_KEY` which
        // is used by EventBridge for encryption operation. This object is an entry
 point for decryption operation.
        // It can later use decryptData(MasterKeyProvider, byte[]) method to decrypt
 data.
        final AwsCrypto crypto = AwsCrypto.builder()
```

```
 .withEncryptionAlgorithm(CryptoAlgorithm.ALG_AES_256_GCM_HKDF_SHA512_COMMIT_KEY)
                .build();

        // Construct AWS KMS master key provider with AWS KMS Client Supplier and AWS
 KMS Key ARN. The KMS Client Supplier can
        // implement a RegionalClientSupplier interface. The AWS KMS Key ARN can be
 fetched from kms-key-arn property in
        // encrypted event json detail.
        final KmsMasterKeyProvider kmsMasterKeyProvider =
 KmsMasterKeyProvider.builder()
                .customRegionalClientSupplier(...)
                .buildStrict(KMS_KEY_ARN);

        // The string of encrypted-payload is base64 encoded. Decode it into byte
 array, so it can be furthur
        // decrypted. The encrypted payload can be fetched from encrypted-payload field
 in encrypted event json detail.
        byte[] encryptedByteArray = Base64.getDecoder().decode(ENCRYPTED_PAYLOAD);

        // The decryption operation. It retrieves the encryption context and encrypted
 data key from the cipher
        // text headers, which is parsed from byte array encrypted data. Then it
 decrypts the data key, and
        // uses it to finally decrypt event payload. This encryption/decryption
 strategy is called envelope
        // encryption, https://docs.aws.amazon.com/kms/latest/developerguide/
concepts.html#enveloping
        final CryptoResult<byte[], KmsMasterKey> decryptResult =
 crypto.decryptData(kmsMasterKeyProvider, encryptedByteArray);

        final byte[] decryptedByteArray = decryptResult.getResult();

        // Decode the event json plaintext from byte array into string with UTF_8
 standard.
        String eventJson = new String(decryptedByteArray, StandardCharsets.UTF_8);
```

Políticas baseadas em tags

No Amazon EventBridge, é possível usar políticas baseadas em tags para controlar o acesso a recursos.

Por exemplo, é possível restringir o acesso a todos os recursos que incluam uma tag com a chave `environment` e o valor `production`: O exemplo de política a seguir nega a qualquer recurso com esta tag a capacidade de criar, excluir ou modificar tags, regras ou barramentos de eventos para recursos que tenham sido marcados com `environment/production`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "events:PutRule",
                "events:DescribeRule",
                "events:DeleteRule",
                "events:CreateEventBus",
                "events:DescribeEventBus"
                "events:DeleteEventBus"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/environment": "production"}
            }
        }
    ]
}
```

Para obter mais informações sobre a atribuição de tags, consulte:

- EventBridge Etiquetas da Amazon
- Controlar o acesso com tags do IAM

Amazon EventBridge e AWS Identity and Access Management

Para acessar a Amazon EventBridge, você precisa de credenciais que AWS possam ser usadas para autenticar suas solicitações. Suas credenciais devem ter permissões para acessar os recursos da AWS, como recuperar dados de eventos de outros recursos da AWS. As seções a seguir fornecem detalhes sobre como você pode usar o AWS Identity and Access Management(IAM) e como ajudar EventBridge a proteger seus recursos controlando quem pode acessá-los.

Tópicos

Autenticação

Você pode acessar a AWS usando qualquer um dos seguintes tipos de identidade:

- Usuário raiz da conta da AWS: ao se cadastrar na AWS, é fornecido um endereço de e-mail e uma senha que são associados à sua conta. Estas são suas credenciais raiz e elas fornecem acesso total a todos os seus recursos da AWS.

> **Important**
>
> Por motivos de segurança, recomendamos que você use as credenciais raiz somente para criar um administrador, que é um usuário do IAM com permissões totais na sua conta. Depois, você pode usar esse administrador para criar outros usuários e funções do com permissões limitadas. Para mais informações, consulte Melhores práticas do IAM e Criar um grupo e um usuário administrador no Guia do usuário do IAM.

- Usuário do IAM — Um usuário do IAM é uma identidade em sua conta que tem permissões específicas, por exemplo, permissão para enviar dados de eventos para um alvo em EventBridge. É possível usar credenciais de login do IAM para acessar páginas da web seguras da AWS, como o AWS Management Console, os Fóruns de discussão da AWS ou o AWS Support Center.

  Além das credenciais de login, você também pode gerar chaves de acesso para cada usuário. É possível usar essas chaves ao acessar serviços da AWS de forma programática para assinar sua solicitação com criptografia, seja com um dos vários SDKs ou usando o AWS Command Line Interface (AWS CLI). Se não usar ferramentas da AWS, você mesmo deverá assinar a solicitação com o Signature Version 4, um protocolo para a autenticação de solicitações da API de entrada. Para obter mais informações sobre a autenticação de solicitações, consulte Processo de cadastramento do Signature versão 4 na Referência geral da Amazon Web Services.
- Perfil do IAM: um perfil do IAM é uma identidade do IAM que é possível criar em sua conta com permissões específicas. É semelhante a um usuário do IAM mas não está associada a uma pessoa específica. Um perfil do IAM permite obter chaves de acesso temporárias que podem acessar os serviços e recursos da AWS. Os perfis do IAM com credenciais temporárias são úteis nas seguintes situações:
  - Acesso de usuário federado: em vez de criar um usuário, é possível usar identidades já existente do AWS Directory Service, o diretório de usuário da sua empresa ou um provedor de identidades (IdP) da web. Eles são conhecidos como usuários federados. A AWS atribui um perfil a um usuário federado quando o usuário solicita o acesso por meio de um provedor de identidades. Para obter mais informações sobre usuários federados, consulte Usuários federados e funções no Manual do usuário do IAM.
  - Acesso entre contas: é possível usar um perfil do IAM em sua conta para conceder, a outra conta, permissões de acesso aos recursos de sua conta. Para ver um exemplo, consulte o Tutorial: delegar acesso em AWS usando funções do IAM no Manual do usuário do IAM.
  - Acesso de serviço da AWS: é possível usar um perfil do IAM em sua conta para conceder permissões de serviço da AWS para acessar os recursos de sua conta. Por exemplo, é possível criar um perfil que permita ao Amazon Redshift carregar os dados armazenados em um bucket do Amazon S3 em um cluster do Amazon Redshift. Para mais informações, consulte Criar uma função para delegar permissões a um serviço da AWS, no Manual do usuário do IAM.
  - Aplicativos em execução no Amazon EC2 — Para aplicativos do Amazon EC2 que precisam de acesso EventBridge, você pode armazenar chaves de acesso na instância do EC2 ou usar uma função do IAM para gerenciar credenciais temporárias. Para atribuir um perfil da AWS a uma instância do EC2, crie um perfil de instância anexado à instância. Um perfil de instância contém o perfil e fornece credenciais temporárias para aplicações em execução na instância do

EC2. Para obter mais informações, consulte Uso de funções para aplicações no Amazon EC2 no Manual do usuário do IAM.

Controle de acesso

Para criar ou acessar EventBridge recursos, você precisa de credenciais e permissões válidas. Por exemplo, para invocar destinos do AWS Lambda, o Amazon Simple Notification Service (Amazon SNS) e o Amazon Simple Queue Service (Amazon SQS) (Amazon SQS), é preciso ter permissões para esses serviços.

Como gerenciar permissões de acesso aos seus recursos do Amazon EventBridge

O acesso aos recursos do EventBridge, como regras ou eventos, são gerenciados usando políticas baseadas em identidade ou recursos.

Recursos do EventBridge

Recursos e sub-recursos do EventBridge têm nomes do recurso da Amazon (ARNs) exclusivos associados a eles. ARNs no EventBridge são usados para criar padrões de eventos. Para obter mais informações sobre ARNs, consulte Nomes de recurso da Amazon (ARN) e namespaces de serviço da AWS no Referência geral da Amazon Web Services.

Para obter uma lista das operações que o EventBridge fornece para trabalhar com recursos, consulte Referência de permissões do Amazon EventBridge.

> **Note**
>
> A maioria dos serviços na AWS trata os dois pontos (`:`) e a barra inclinada (`/`) como o mesmo caractere em ARNs. No entanto, o EventBridge usa uma correspondência exata nas regras e nos padrões de eventos. Use os caracteres de ARN corretos ao criar padrões de evento para que eles correspondam à sintaxe do ARN no evento a que você quer corresponder.

A tabela a seguir mostra os recursos do EventBridge.

Tipo de recurso	Formato de Nome de região da Amazon (ARN)
Arquivo	`arn:aws:events:` *`region`*`:`*`account`*`:archive/` *`archive-name`*
Reproduzir novamente	`arn:aws:events:` *`region`*`:`*`account`*`:replay/`*`replay-name`*
Regra	`arn:aws:events:` *`region`*`:`*`account`*`:rule/[event-bus-name]/`*`rule-name`*

Tipo de recurso	Formato de Nome de região da Amazon (ARN)
Barramento de eventos	`arn:aws:events:` *`region`*`:`*`account`*`:event-bus/` *`event-bus-name`*
Todos os recursos do EventBridge	`arn:aws:events:*`
Todos os recursos do EventBridge pertencentes à conta especificada na região especificada	`arn:aws:events:` *`region`*`:`*`account`*`:*`

O exemplo a seguir mostrar como indicar uma regra específica (*`myRule`*) em sua declaração usando o ARN.

```
"Resource": "arn:aws:events:us-east-1:123456789012:rule/myRule"
```

Para especificar todas as regras pertencentes a uma conta específica usando o curinga de asterisco (*), conforme o seguinte:

```
"Resource": "arn:aws:events:us-east-1:123456789012:rule/*"
```

Para especificar todos os recursos ou se uma ação de API específica não for compatível com ARNs, use o curinga de asterisco (*) no elemento `Resource`, conforme o seguinte:

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

Para especificar vários recursos ou `PutTargets` em uma única instrução, separe seus ARNs com vírgulas, conforme o seguinte:

```
"Resource": ["arn1", "arn2"]
```

Propriedade do recurso

Uma conta é proprietária dos recursos da conta, independentemente de quem os cria. Isto é, o proprietário do recurso é a conta da entidade principal, a conta do usuário-raiz, um usuário ou perfil

do IAM que autentica a solicitação que cria o recurso. Os exemplos a seguir mostram como isso funciona:

- Se usar as credenciais o usuário-raiz da sua conta para criar uma regra, sua conta será a proprietária do recurso do EventBridge.
- Se criar um usuário na sua conta e conceder permissões para criar recursos do EventBridge para esse usuário, ele poderá criar recursos do EventBridge. No entanto, sua conta, à qual o usuário pertence, é proprietária dos recursos do EventBridge.
- Se criar um perfil do IAM na sua conta com permissões para criar recursos do EventBridge, qualquer pessoa que puder assumir o perfil poderá criar recursos do EventBridge. Sua conta, à qual o perfil pertence, é proprietária dos recursos do EventBridge.

Gerenciamento de acesso aos recursos

A política de permissões descreve quem tem acesso a quê. A seção a seguir explica as opções disponíveis para a criação das políticas de permissões.

> **Note**
>
> Esta seção discute o uso do IAM no contexto do EventBridge. Não são fornecidas informações detalhadas sobre o serviço IAM. Para ver a documentação completa do IAM, consulte O que é o IAM? no Guia do usuário do IAM. Para obter informações sobre a sintaxe e as descrições da política do IAM, consulte a referência da política do IAM da no Guia do usuário do IAM.

As políticas anexadas a uma identidade do IAM são conhecidas como políticas baseadas em identidade (políticas do IAM;) e as políticas anexadas a um recurso são conhecidas como políticas baseadas em recurso. No EventBridge, é possível usar ambas as políticas baseadas em identidade (políticas do IAM) e recurso.

Tópicos

- Políticas baseadas em identidade (políticas do IAM)
- Políticas baseadas em recursos (Políticas do IAM)

Políticas baseadas em identidade (políticas do IAM)

Você pode anexar políticas a identidades do IAM. Por exemplo, você pode fazer o seguinte:

- Anexar uma política de permissões a um usuário ou um grupo em sua conta: para conceder a um usuário permissão para visualizar regras no console do Amazon CloudWatch, é possível associar uma política de permissões a um usuário ou a um grupo ao qual o usuário pertence.
- Anexar uma política de permissões a uma função (grant cross-account permissions): você pode anexar uma política de permissões baseada em identidade a um perfil do IAM para conceder permissões entre contas. Por exemplo, o administrador na conta A pode criar um perfil para conceder permissões entre contas para outra conta B ou para um serviço da AWS da seguinte forma:
  1. Um administrador da conta A cria um perfil do IAM e anexa uma política de permissões ao perfil que concede permissão em recursos da conta A.
  2. Um administrador da conta A anexa uma política de confiança à função identificando a conta B como a entidade principal, que pode assumir a função.
  3. O administrador da conta B pode delegar permissões para assumir o perfil para todos os usuários na conta B. Fazer isso permite que os usuários na conta B criem ou acessem recursos na conta A. A entidade principal na política de confiança também pode ser a entidade principal do serviço da AWS para conceder a um serviço da AWS as permissões para assumir o perfil.

  Para obter mais informações sobre o uso do IAM para delegar permissões, consulte Gerenciamento de acesso no Guia do usuário do IAM.

É possível criar políticas do IAM específicas para restringir as chamadas e os recursos a que os usuários em sua conta têm acesso e associar essas políticas aos usuários. Para obter mais informações sobre como criar perfis do IAM e ver exemplos de declarações de política do IAM no CodeCommit, consulte Como gerenciar permissões de acesso aos seus recursos do Amazon EventBridge.

Políticas baseadas em recursos (Políticas do IAM)

Quando uma regra é executada no EventBridge, todos os destinos associados à regra são invocados, o que significa invocar as funções do AWS Lambda, publicar nos tópicos do Amazon SNS ou retransmitir o evento para os fluxos do Amazon Kinesis. Para fazer chamadas de API para os seus próprios recursos, o EventBridge precisa das permissões adequadas. Para recursos do

Lambda, do Amazon SNS e do Amazon SQS, o EventBridge conta com políticas baseadas em recursos. Para fluxos do Kinesis, o EventBridge usa perfis do IAM.

Para obter mais informações sobre como criar perfis do IAM e explorar instruções de política baseadas em recursos do EventBridge, consulte Como usar políticas baseadas em recursos para esquemas do Amazon EventBridge.

Especificar elementos da política: ações, efeitos e entidades principais

Para cada recurso do EventBridge, o EventBridge define um conjunto de operações da API. Para conceder permissões a essas operações da API, o EventBridge define um conjunto de ações que podem ser especificadas em uma política. Algumas operações da API exigem permissões para mais de uma ação para realizar a operação da API. Para obter mais informações sobre os recursos e operações da API, consulte Recursos do EventBridge e Referência de permissões do Amazon EventBridge.

Estes são os elementos de política básicos:

- Recurso: use um nome do recurso da Amazon (ARN) para identificar o recurso ao qual a política se aplica. Para obter mais informações, consulte Recursos do EventBridge.
- Ação: use palavras-chave para identificar as operações de recurso que deseja permitir ou negar. Por exemplo, a permissão `events:Describe` permite que o usuário execute a operação `Describe`.
- Efeito: especifique permitir ou negar. Se não conceder (permitir) explicitamente acesso a um recurso, o acesso estará negado. Também é possível negar explicitamente o acesso a um recurso para ter certeza de que um usuário não conseguirá acessá-lo, mesmo que uma política diferente conceda acesso.
- Entidade principal: em políticas baseadas em identidade (políticas do IAM), o usuário ao qual a política é anexada é a entidade principal implícita. Para as políticas baseadas em recursos, você especifica quais usuários, contas, serviços ou outras entidades deseja que recebam permissões (isso se aplica somente a políticas baseadas em recursos).

Para obter informações sobre a sintaxe e as descrições da política do IAM, consulte a Referência da política JSON do IAM no Manual do usuário do IAM.

Para obter informações sobre ações de API do EventBridge e os recursos aos quais elas se aplicam, consulte Referência de permissões do Amazon EventBridge.

Especificar condições em uma política

Ao conceder permissões, você pode usar a linguagem da política de acesso para especificar as condições quando uma política deve entrar em vigor. Por exemplo, é recomendável aplicar uma política somente após uma data específica. Para obter mais informações sobre como especificar condições em uma linguagem de política, consulte Condição no Guia do usuário do IAM.

Para expressar condições, são usadas chaves de condição. Existem chaves de condição da AWS chaves específicas do EventBridge que podem ser usadas de forma adequada. Para obter uma lista completa de chaves da AWS, consulte Chaves disponíveis para condições no Guia do usuário do IAM. Para obter uma lista completa de chaves específicas do EventBridge, consulte Uso de condições de política do IAM para controle de acesso refinado.

Usando políticas baseadas em identidade (políticas do IAM) para a Amazon EventBridge

As políticas baseadas em identidade são políticas de permissões anexadas a identidades do IAM.

Tópicos

AWS políticas gerenciadas para EventBridge

AWS aborda muitos casos de uso comuns fornecendo políticas autônomas do IAM que são criadas e administradas pela AWS. Políticas gerenciadas, ou predefinidas, concedem as permissões necessárias 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 AWS gerenciadas que você pode anexar aos usuários em sua conta são específicas para EventBridge:

- AmazonEventBridgeFullAccess— Concede acesso total a EventBridge, incluindo EventBridge Pipes, EventBridge Schemas e EventBridge Scheduler.
- AmazonEventBridgeReadOnlyAccess— Concede acesso somente de leitura a EventBridge, incluindo EventBridge Pipes, EventBridge Schemas e Scheduler. EventBridge

AmazonEventBridgeFullAccess política

A AmazonEventBridgeFullAccess política concede permissões para usar todas EventBridge as ações, bem como as seguintes permissões:

- `iam:CreateServiceLinkedRole`— EventBridge requer essa permissão para criar a função de serviço em sua conta para destinos de API. Esta permissão concede somente ao serviço do IAM permissões para criar um perfil em sua conta especificamente para destinos de API.
- `iam:PassRole`— EventBridge requer essa permissão para passar uma função de invocação para EventBridge invocar o destino de uma regra.

- Permissões do Secrets Manager — EventBridge exigem essas permissões para gerenciar segredos em sua conta quando você fornece credenciais por meio do recurso de conexão para autorizar destinos de API.

O JSON a seguir mostra a AmazonEventBridgeFullAccess política.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EventBridgeActions",
            "Effect": "Allow",
            "Action": [
                "events:*",
                "schemas:*",
                "scheduler:*",
                "pipes:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IAMCreateServiceLinkedRoleForApiDestinations",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
AmazonEventBridgeApiDestinationsServiceRolePolicy",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "apidestinations.events.amazonaws.com"
                }
            }
        },

        {
            "Sid": "SecretsManagerAccessForApiDestinations",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:UpdateSecret",
                "secretsmanager:DeleteSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue"
```

```
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:events!*"
        },
        {
            "Sid": "IAMPassRoleAccessForEventBridge",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "events.amazonaws.com"
                }
            }
        },
        {
            "Sid": "IAMPassRoleAccessForScheduler",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        },
        {
            "Sid": "IAMPassRoleAccessForPipes",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "pipes.amazonaws.com"
                }
            }
        }
    ]
}
```

> **Note**
>
> As informações nesta seção também se aplicam à política `CloudWatchEventsFullAccess`. No entanto, é altamente recomendável que você use a Amazon EventBridge em vez da Amazon CloudWatch Events.

AmazonEventBridgeReadOnlyAccess política

A AmazonEventBridgeReadOnlyAccess política concede permissões para usar todas as EventBridge ações de leitura.

O JSON a seguir mostra a AmazonEventBridgeReadOnlyAccess política.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "events:DescribeRule",
                "events:DescribeEventBus",
                "events:DescribeEventSource",
                "events:ListEventBuses",
                "events:ListEventSources",
                "events:ListRuleNamesByTarget",
                "events:ListRules",
                "events:ListTargetsByRule",
                "events:TestEventPattern",
                "events:DescribeArchive",
                "events:ListArchives",
                "events:DescribeReplay",
                "events:ListReplays",
                "events:DescribeConnection",
                "events:ListConnections",
                "events:DescribeApiDestination",
                "events:ListApiDestinations",
                "events:DescribeEndpoint",
                "events:ListEndpoints",
                "schemas:DescribeCodeBinding",
                "schemas:DescribeDiscoverer",
                "schemas:DescribeRegistry",
```

```
                "schemas:DescribeSchema",
                "schemas:ExportSchema",
                "schemas:GetCodeBindingSource",
                "schemas:GetDiscoveredSchema",
                "schemas:GetResourcePolicy",
                "schemas:ListDiscoverers",
                "schemas:ListRegistries",
                "schemas:ListSchemas",
                "schemas:ListSchemaVersions",

                "schemas:ListTagsForResource",
                "schemas:SearchSchemas",
                "scheduler:GetSchedule",
                "scheduler:GetScheduleGroup",
                "scheduler:ListSchedules",
                "scheduler:ListScheduleGroups",
                "scheduler:ListTagsForResource",
                "pipes:DescribePipe",
                "pipes:ListPipes",
                "pipes:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> As informações nesta seção também se aplicam à política `CloudWatchEventsReadOnlyAccess`. No entanto, é altamente recomendável que você use a Amazon EventBridge em vez da Amazon CloudWatch Events.

EventBridge Políticas gerenciadas específicas do esquema

Um esquema define a estrutura dos eventos para os quais são enviados EventBridge. EventBridge fornece esquemas para todos os eventos gerados pelos AWS serviços. As seguintes políticas AWS gerenciadas específicas para EventBridge esquemas estão disponíveis:

- AmazonEventBridgeSchemasServiceRolePolicy
- AmazonEventBridgeSchemasFullAccess

- AmazonEventBridgeSchemasReadOnlyAccess

EventBridge Políticas gerenciadas específicas do agendador

O Amazon EventBridge Scheduler é um programador sem servidor que permite criar, executar e gerenciar tarefas a partir de um serviço gerenciado central. Para políticas AWS gerenciadas específicas do EventBridge Scheduler, consulte políticas AWS gerenciadas para o EventBridge Scheduler no Guia do usuário do EventBridge Scheduler.

EventBridge Políticas gerenciadas específicas para tubos

O Amazon EventBridge Pipes conecta as fontes de eventos aos alvos. O Pipes reduz a necessidade de conhecimento especializado e de código de integração ao desenvolver arquiteturas orientada por eventos. Isto ajuda a garantir a consistência em todas as aplicações da sua empresa. As seguintes políticas AWS gerenciadas específicas do EventBridge Pipes estão disponíveis:

- AmazonEventBridgePipesFullAccess

  Fornece acesso total ao Amazon EventBridge Pipes.

  > **Note**
  >
  > Esta política fornece `iam:PassRole` — EventBridge Pipes requer essa permissão para passar uma função de invocação EventBridge para criar e iniciar pipes.

- AmazonEventBridgePipesReadOnlyAccess

  Fornece acesso somente para leitura ao Amazon EventBridge Pipes.

- AmazonEventBridgePipesOperatorAccess

  Fornece acesso somente de leitura e de operador (ou seja, a capacidade de parar e começar a executar o Pipes) ao Amazon EventBridge Pipes.

Perfis do IAM para o envio de eventos

Para retransmitir eventos para alvos, EventBridge precisa de uma função do IAM.

Para criar uma função do IAM para enviar eventos para EventBridge

1. Abra o console IAM em https://console.aws.amazon.com/iam/.

2. Para criar uma função do IAM, siga as etapas em Criação de uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM. À medida que você realizar as etapas, observe o seguinte:

   - Em Nome do perfil, use um nome que seja exclusivo em sua conta.
   - Em Selecionar tipo de função, escolha AWS Service Roles e, em seguida, escolha Amazon EventBridge. Isso concede EventBridge permissões para assumir a função.
   - Em Anexar política, escolha AmazonEventBridgeFullAccess.

Você também pode criar suas próprias políticas personalizadas do IAM para permitir permissões para EventBridge ações e recursos. Você pode anexar essas políticas personalizadas a usuários ou grupos do IAM que exijam essas permissões. Para obter mais informações sobre políticas do IAM, consulte Visão geral das políticas do IAM no Guia do usuário do IAM. Para obter mais informações sobre o gerenciamento e a criação de políticas personalizadas do IAM, consulte Gerenciamento de políticas do IAM no Guia do usuário do IAM.

Permissões necessárias EventBridge para acessar destinos usando funções do IAM

EventBridge Os destinos geralmente exigem funções do IAM que concedam permissão EventBridge para invocar o destino. A seguir estão alguns exemplos de vários AWS serviços e alvos. Para outros, use o EventBridge console para criar uma regra e criar uma nova função que será criada com uma política com permissões bem definidas pré-configuradas.

Amazon SQS, Amazon SNS, CloudWatch Lambda, Logs e destinos de barramento não usam funções EventBridge , e EventBridge as permissões devem ser concedidas por meio de uma política de recursos. Os destinos do API Gateway podem usar políticas de recursos ou perfis do IAM.

Se o destino for um destino da API, o perfil especificado deverá incluir a seguinte política:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Action": [ "events:InvokeApiDestination" ],
            "Resource": [ "arn:aws:events:::api-destination/*" ]
        }
     ]
}
```

Se o destino for um fluxo do Kinesis, o perfil usado para enviar dados de eventos para o destino deverá incluir a seguinte política:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
    ]
}
```

Se o destino for o comando executar do Systems Manager e forem especificados um ou mais valores `InstanceIds` para o comando, o perfil especificado deverá incluir a política a seguir.

```
{
     "Version": "2012-10-17",
     "Statement": [
           {
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
                "Resource": [
                     "arn:aws:ec2:region:accountId:instance/instanceIds",
                     "arn:aws:ssm:region:*:document/documentName"
                ]
           }
     ]
}
```

Se o destino for o comando executar do Systems Manager e você especificar uma ou mais tags para o comando, o perfil que você especificar deverá incluir a seguinte política:

```
{
     "Version": "2012-10-17",
     "Statement": [
           {
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
```

```
                "Resource": [
                     "arn:aws:ec2:region:accountId:instance/*"
                ],
                "Condition": {
                    "StringEquals": {
                        "ec2:ResourceTag/*": [
                            "[[tagValues]]"
                        ]
                    }
                }
            },
            {
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
                "Resource": [
                     "arn:aws:ssm:region:*:document/documentName"
                ]
            }
     ]
}
```

Se o destino for uma máquina de AWS Step Functions estado, a função especificada deverá incluir a política a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Action": [ "states:StartExecution" ],
            "Resource": [ "arn:aws:states:*:*:stateMachine:*" ]
        }
     ]
}
```

Se o destino for uma tarefa do Amazon ECS, o perfil especificado deverá incluir a seguinte política:

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ecs:RunTask"
```

```
        ],
        "Resource": [
            "arn:aws:ecs:*:account-id:task-definition/task-definition-name"
        ],
        "Condition": {
            "ArnLike": {
                "ecs:cluster": "arn:aws:ecs:*:account-id:cluster/cluster-name"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action":"iam:PassRole",
        "Resource": [
            "*"
        ],
        "Condition": {
            "StringLike": {
                "iam:PassedToService": "ecs-tasks.amazonaws.com"
            }
        }
    }]
}
```

A política a seguir permite que alvos EventBridge incorporados executem ações do Amazon EC2 em seu nome. Você precisa usar o AWS Management Console para criar regras com alvos integrados.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TargetInvocationAccess",
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:RebootInstances",
                "ec2:StopInstances",
                "ec2:TerminateInstances",
                "ec2:CreateSnapshot"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

A política a seguir permite EventBridge retransmitir eventos para os streams do Kinesis em sua conta.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "KinesisAccess",
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
    ]
}
```

Exemplo de política gerenciada pelo cliente: uso de marcações para controlar o acesso às regras

O exemplo a seguir mostra uma política de usuário que concede permissões para EventBridge ações. Essa política funciona quando você usa a EventBridge API, AWS os SDKs ou o. AWS CLI

Você pode conceder aos usuários acesso a EventBridge regras específicas e, ao mesmo tempo, impedir que eles acessem outras regras. Para fazer isso, marque ambos os conjuntos de regras e use as políticas do IAM que se referem a essas tags. Para obter mais informações sobre a marcação de EventBridge recursos, consulte EventBridge Etiquetas da Amazon.

É possível conceder uma política do IAM a um usuário para permitir acesso apenas às regras com uma determinada tag. As regras são escolhidas para conceder acesso ao marcá-las com esta tag específica. Por exemplo, a política a seguir concede acesso a regras com o valor `Prod` para a chave de tag `Stack`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "events:*",
            "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Stack": "Prod"
                }
            }
        }
    ]
}
```

Para obter mais informações sobre como usar instruções de política do IAM, consulte Controlar o acesso usando políticas no Manual do usuário do IAM.

EventBridge Atualizações da Amazon para políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas EventBridge desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página Histórico do EventBridge documento.

Alteração	Descrição	Data
AmazonEventBridgeFullAccess: política atualizada	AWS GovCloud (US) Regions somente A permissão a seguir não está incluída, pois não é usada: • `iam:CreateServiceLinkedRole` permissão para o EventBridge Schema Registry	9 de maio de 2024
AmazonEventBridgeSchemasFullAccess: política atualizada	AWS GovCloud (US) Regions somente A permissão a seguir não está incluída, pois não é usada: • `iam:CreateServiceLinkedRole` permissão	9 de maio de 2024

Alteração	Descrição	Data
	para o EventBridge Schema Registry	
AmazonEventBridgePipesFullAccess— Nova política adicionada	EventBridge adicionou uma política gerenciada para permissões completas de uso do EventBridge Pipes.	1º de dezembro de 2022
AmazonEventBridgePipesReadOnlyAccess— Nova política adicionada	EventBridge adicionou uma política gerenciada para permissões para visualizar os recursos EventBridge de informações do Pipes.	1º de dezembro de 2022
AmazonEventBridgePipesOperatorAccess— Nova política adicionada	EventBridge adicionou uma política gerenciada para obter permissões para visualizar as informações do EventBridge Pipes, bem como iniciar e parar de operar os tubos.	1º de dezembro de 2022
AmazonEventBridgeFullAccess: atualização para uma política existente	EventBridge atualizou a política para incluir as permissões necessárias para usar os recursos do EventBridge Pipes.	1º de dezembro de 2022

Alteração	Descrição	Data
AmazonEventBridgeReadOnlyAccess: atualização para uma política existente	EventBridge adicionou as permissões necessárias para visualizar os recursos EventBridge de informações do Pipes. As seguintes ações foram adicionadas: • `pipes:DescribePipe` • `pipes:ListPipes` • `pipes:ListTagsForResource`	1º de dezembro de 2022
CloudWatchEventsReadOnlyAccess: atualização para uma política existente	Atualizado para corresponder AmazonEventBridgeReadOnlyAccess.	1º de dezembro de 2022
CloudWatchEventsFullAccess: atualização para uma política existente	Atualizado para corresponder AmazonEventBridgeFullAccess.	1º de dezembro de 2022

Alteração	Descrição	Data
AmazonEventBridgeFullAccess: atualização para uma política existente	EventBridge atualizou a política para incluir as permissões necessárias para usar esquemas e recursos do agendador. As seguintes permissões foram adicionadas: • EventBridge Ações do Registro do Esquema • EventBridge Ações do agendador • `iam:CreateServiceLinkedRole` permissão para o EventBridge Schema Registry • `iam:PassRole` permissão para o EventBridge Scheduler	10 de novembro de 2022

Alteração	Descrição	Data
AmazonEventBridgeReadOnlyAccess: atualização para uma política existente	EventBridge adicionou as permissões necessárias para visualizar os recursos de informações do esquema e do agendador. As seguintes ações foram adicionadas: • `schemas:DescribeCodeBinding` • `schemas:DescribeDiscoverer` • `schemas:DescribeRegistry` • `schemas:DescribeSchema` • `schemas:ExportSchema` • `schemas:GetCodeBindingSource` • `schemas:GetDiscoveredSchema` • `schemas:GetResourcePolicy` • `schemas:ListDiscoverers` • `schemas:ListRegistries` • `schemas:ListSchemas` • `schemas:ListSchemaVersions`	10 de novembro de 2022

Alteração	Descrição	Data
	• schemas:ListTagsForResource • schemas:SearchSchemas • scheduler:GetSchedule • scheduler:GetScheduleGroup • scheduler:ListSchedules • scheduler:ListScheduleGroups • scheduler:ListTagsForResource	
AmazonEventBridgeReadOnlyAccess: atualização para uma política existente	EventBridge adicionou as permissões necessárias para visualizar as informações do endpoint. As seguintes ações foram adicionadas: • events:ListEndpoints • events:DescribeEndpoint	7 de abril de 2022

Alteração	Descrição	Data
AmazonEventBridgeReadOnlyAccess: atualização para uma política existente	EventBridge adicionou as permissões necessárias para visualizar as informações de conexão e destino da API. As seguintes ações foram adicionadas: • `events:DescribeConnection` • `events:ListConnections` • `events:DescribeApiDestination` • `events:ListApiDestinations`	4 de março de 2021

Alteração	Descrição	Data
AmazonEventBridgeFullAccess: atualização para uma política existente	EventBridge atualizou a política para incluir `iam:CreateServiceLinkedRole` as AWS Secrets Manager permissões necessárias para usar destinos de API. As seguintes ações foram adicionadas: • `secretsmanager:CreateSecret` • `secretsmanager:UpdateSecret` • `secretsmanager:DeleteSecret` • `secretsmanager:GetSecretValue` • `secretsmanager:PutSecretValue`	4 de março de 2021
EventBridge começou a rastrear alterações	EventBridge começou a rastrear as mudanças em suas políticas AWS gerenciadas.	4 de março de 2021

Como usar políticas baseadas em recursos para esquemas do Amazon EventBridge

Quando uma regra é executada no EventBridge, todos os destinos associados à regra são invocados. As regras podem invocar funções do AWS Lambda, publicar em tópicos do Amazon SNS ou retransmitir o evento para os fluxos do Kinesis. Para fazer chamadas de API com seus próprios recursos, o EventBridge precisa das permissões adequadas. Para recursos do Lambda, do Amazon SNS, do Amazon SQS e do Amazon CloudWatch Logs o EventBridge conta com políticas baseadas em recursos. Para fluxos do Kinesis, o EventBridge usa políticas baseadas em identidade.

Você usa a AWS CLI para adicionar permissões aos seus destinos. Para obter informações sobre como instalar e configurar a AWS CLI, consulte Configuração com a AWS Command Line Interface no Guia do usuário do AWS Command Line Interface.

Tópicos

Permissões do Amazon API Gateway

Para invocar seu endpoint do Amazon API Gateway usando uma regra do EventBridge, adicione a permissão a seguir à política do seu endpoint do API Gateway.

```
{
  "Version": "2012-10-17",
  "Statement": [
       {
           "Effect": "Allow",
           "Principal": {
               "Service": "events.amazonaws.com"
           },
           "Action": "execute-api:Invoke",
           "Condition": {
```

```
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:events:region:account-id:rule/rule-name"
                }
            },
            "Resource": [
                "execute-api:/stage/GET/api"
            ]
        }
    ]
}
```

Permissões do CloudWatch Logs

Quando o CloudWatch Logs é o destino de uma regra, o EventBridge cria fluxos de logs e o CloudWatch Logs armazena o texto dos eventos de disparo como entradas de log. Para permitir que o EventBridge crie o fluxo de logs e os eventos de log, o CloudWatch Logs deve incluir uma política baseada em recursos que permita ao EventBridge fazer a gravação no CloudWatch Logs.

Se usar o AWS Management Console para adicionar o CloudWatch Logs como destino de uma regra, essa política será criada automaticamente. Se usar a AWS CLI para adicionar o destino, deve criar essa política se ela não existir.

Este exemplo permite que o EventBridge grave em todos os grupos de logs que têm nomes que começam com `/aws/events/`. Se usar uma política diferente para dar nome a esses tipos de logs, ajuste a política de acordo com a necessidade.

```
{
    "Statement": [
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": ["events.amazonaws.com", "delivery.logs.amazonaws.com"]
            },
            "Resource": "arn:aws:logs:region:account:log-group:/aws/events/*:*",
            "Sid": "TrustEventsToStoreLogEvent"
        }
    ],
    "Version": "2012-10-17"
```

```
}
```

Para obter mais informações, consulte PutResourcePolicy na Guia de referência de APIs do Amazon CloudWatch Logs.

Permissões AWS Lambda

Para invocar sua função do AWS Lambda usando uma regra do EventBridge, adicione a seguinte permissão à política da função do Lambda.

```
{
  "Effect": "Allow",
  "Action": "lambda:InvokeFunction",
  "Resource": "arn:aws:lambda:region:account-id:function:function-name",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:events:region:account-id:rule/rule-name"
    }
  },
  "Sid": "InvokeLambdaFunction"
}
```

Para adicionar as permissões acima que permitem que o EventBridge invoque funções do Lambda usando a AWS CLI

- Em um prompt de comando, digite o seguinte comando.

```
aws lambda add-permission --statement-id "InvokeLambdaFunction" \
--action "lambda:InvokeFunction" \
--principal "events.amazonaws.com" \
--function-name "arn:aws:lambda:region:account-id:function:function-name" \
--source-arn "arn:aws:events:region:account-id:rule/rule-name"
```

Para obter mais informações sobre a configuração de permissões que permitem ao invocar funções do Lambda, consulte Adicionar permissão e Como usar o Lambda com eventos programados no Manual do desenvolvedor do AWS Lambda.

Permissões do Amazon SNS

Para permitir que o EventBridge publique um tópico do Amazon SNS, use os comandos `aws sns get-topic-attributes` e `aws sns set-topic-attributes`.

> **Note**
>
> Não é possível usar blocos de `Condition` nas políticas de tópicos do Amazon SNS para o EventBridge.

Como adicionar permissões que permitam ao EventBridge publicar tópicos do SNS

1. Para listar os atributos de um tópico do SNS, use o comando a seguir.

   ```
   aws sns get-topic-attributes --topic-arn "arn:aws:sns:region:account-id:topic-name"
   ```

   O exemplo a seguir mostra o resultado de um novo tópico do SNS.

   ```
   {
       "Attributes": {
           "SubscriptionsConfirmed": "0",
           "DisplayName": "",
           "SubscriptionsDeleted": "0",
           "EffectiveDeliveryPolicy": "{\"http\":{\"defaultHealthyRetryPolicy\":
   {\"minDelayTarget\":20,\"maxDelayTarget\":20,\"numRetries\":3,\"numMaxDelayRetries
   \":0,\"numNoDelayRetries\":0,\"numMinDelayRetries\":0,\"backoffFunction\":\"linear
   \"},\"disableSubscriptionOverrides\":false}}",
           "Owner": "account-id",
           "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"__default_policy_ID\",
   \"Statement\":[{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal
   \":{\"AWS\":\"*\"},\"Action\":[\"SNS:GetTopicAttributes\",\"SNS:SetTopicAttributes
   \",\"SNS:AddPermission\",\"SNS:RemovePermission\",\"SNS:DeleteTopic\",
   \"SNS:Subscribe\",\"SNS:ListSubscriptionsByTopic\",\"SNS:Publish\"],\"Resource
   \":\"arn:aws:sns:region:account-id:topic-name\",\"Condition\":{\"StringEquals\":
   {\"AWS:SourceOwner\":\"account-id\"}}}]}",
           "TopicArn": "arn:aws:sns:region:account-id:topic-name",
           "SubscriptionsPending": "0"
       }
   }
   ```

2. Use um Conversor de JSON para string para converter a seguinte declaração em uma string.

```
{
  "Sid": "PublishEventsToMyTopic",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": "sns:Publish",
  "Resource": "arn:aws:sns:region:account-id:topic-name"
}
```

Depois de converter a instrução em uma string, ela deve ter a seguinte aparência:

```
{\"Sid\":\"PublishEventsToMyTopic\",\"Effect\":\"Allow\",\"Principal\":
{\"Service\":\"events.amazonaws.com\"},\"Action\":\"sns:Publish\",\"Resource\":
\"arn:aws:sns:region:account-id:topic-name\"}
```

3. Adicione a string que criada na etapa anterior à coleção de `"Statement"` dentro do atributo `"Policy"`.
4. Use o comando `aws sns set-topic-attributes` para especificar a nova política.

```
aws sns set-topic-attributes --topic-arn "arn:aws:sns:region:account-id:topic-name"
 \
--attribute-name Policy \
--attribute-value "{\"Version\":\"2012-10-17\",\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"*\"},\"Action\":[\"SNS:GetTopicAttributes\",\"SNS:SetTopicAttributes
\",\"SNS:AddPermission\",\"SNS:RemovePermission\",\"SNS:DeleteTopic\",
\"SNS:Subscribe\",\"SNS:ListSubscriptionsByTopic\",\"SNS:Publish\"],\"Resource
\":\"arn:aws:sns:region:account-id:topic-name\",\"Condition\":{\"StringEquals
\":{\"AWS:SourceOwner\":\"account-id\"}}}, {\"Sid\":\"PublishEventsToMyTopic\",
\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"},\"Action
\":\"sns:Publish\",\"Resource\":\"arn:aws:sns:region:account-id:topic-name\"}]}"
```

Para obter mais informações, consulte a ação SetTopicAttributes na Referência de API do Amazon Simple Notification Service.

Permissões do Amazon SQS

Para permitir que uma regra do EventBridge invoque uma fila do Amazon SQS, use os comandos `aws sqs get-queue-attributes` e `aws sqs set-queue-attributes`.

Se a política da fila SQS estiver vazia, primeiro será preciso criar uma política e depois adicionar a declaração de permissões a ela. Uma nova fila SQS tem uma política vazia.

Se a fila do SQS já tiver uma política, será preciso copiar a política original e combiná-la com uma nova instrução para adicionar a declaração de permissões a ela.

Para adicionar permissões que permitam ao EventBridge invocar uma fila do SQS

1. Para listar os atributos da fila do SQS. Em um prompt de comando, digite o seguinte comando.

```
aws sqs get-queue-attributes \
--queue-url https://sqs.region.amazonaws.com/account-id/queue-name \
--attribute-names Policy
```

2. Adicione a instrução a seguir.

```
{
      "Sid": "AWSEvents_custom-eventbus-ack-sqs-rule_dlq_sqs-rule-target",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sqs:SendMessage",
      "Resource": "arn:aws:sqs:region:account-id:queue-name",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:events:region:account-id:rule/bus-name/rule-
name"
        }
      }
    }
```

3. Use um Conversor de JSON para string para converter a declaração anterior em uma string. Depois de converter a política em uma string, ela deve ter a seguinte aparência:

```
{\"Sid\": \"EventsToMyQueue\", \"Effect\": \"Allow\", \"Principal\": {\"Service
\": \"events.amazonaws.com\"}, \"Action\": \"sqs:SendMessage\", \"Resource\":
```

```
 \"arn:aws:sqs:region:account-id:queue-name\", \"Condition\": {\"ArnEquals\":
 {\"aws:SourceArn\": \"arn:aws:events:region:account-id:rule/rule-name\"}}
```

4. Crie um arquivo denominado `set-queue-attributes.json` com o seguinte conteúdo.

```
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"arn:aws:sqs:region:account-
id:queue-name/SQSDefaultPolicy\",\"Statement\":[{\"Sid\": \"EventsToMyQueue\",
 \"Effect\": \"Allow\", \"Principal\": {\"Service\": \"events.amazonaws.com\"},
 \"Action\": \"sqs:SendMessage\", \"Resource\": \"arn:aws:sqs:region:account-
id:queue-name\", \"Condition\": {\"ArnEquals\": {\"aws:SourceArn\":
 \"arn:aws:events:region:account-id:rule/rule-name\"}}}]}"
}
```

5. Defina o atributo da política usando o arquivo `set-queue-attributes.json` que acabou de criar como entrada, conforme mostrado no comando a seguir.

```
aws sqs set-queue-attributes \
--queue-url https://sqs.region.amazonaws.com/account-id/queue-name \
--attributes file://set-queue-attributes.json
```

Para obter mais informações sobre o Amazon SQS, consulte o Guia do desenvolvedor do Amazon Simple Queue Service.

Especificações do EventBridge Pipes

O EventBridge Pipes não é compatível com políticas baseadas em recursos nem tem APIs que sejam compatíveis com as condições de políticas baseadas em recursos.

Prevenção contra o ataque “Confused deputy” em todos os serviços

O problema ‘confused deputy’ é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a personificação entre serviços pode resultar no problema do ‘confused deputy’. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, o AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

É recomendado o uso das chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` em políticas de recursos para limitar as permissões que o Amazon EventBridge concede ao recurso para outro serviço. Use `aws:SourceArn` se quiser que apenas um recurso seja associado ao acesso entre serviços. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou estiver especificando vários recursos, use a chave de condição de contexto global `aws:SourceArn` com caracteres curingas (*) para as partes desconhecidas do ARN. Por exemplo, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Se o valor de `aws:SourceArn` não contiver o ID da conta, como um ARN de bucket do Amazon S3, você deverá usar ambas as chaves de contexto de condição global para limitar as permissões.

Barramentos de eventos

Para destinos da regra de barramento de eventos do EventBridge, o valor de `aws:SourceArn` deve ser o ARN da regra.

O exemplo a seguir mostra como é possível usar as chaves de contexto de condição globais `aws:SourceArn` e `aws:SourceAccount` no EventBridge para evitar o problema "confused deputy". Este exemplo é para uso em uma política de confiança de perfil, para um perfil usado por uma regra do EventBridge.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "events.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:events:*:123456789012:rule/myRule"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
```

```
      }
    }
  }
}
```

Pipes do EventBridge

No EventBridge Pipes, o valor de `aws:SourceArn` deve ser o ARN do pipe.

O exemplo a seguir mostra como é possível usar as chaves de contexto de condição globais `aws:SourceArn` e `aws:SourceAccount` no EventBridge para evitar o problema "confused deputy". Este exemplo é para uso em uma política de confiança de perfil, para um perfil usado pelo EventBridge Pipes.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "events.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:pipe:*:123456789012::pipe/example"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Políticas baseadas em recursos para esquemas do Amazon EventBridge

O registro de esquemas do EventBridge é compatível com políticas baseadas em recursos. Uma política baseada em recursos é uma política anexada a um recurso em vez de ser anexada a uma identidade do IAM. Por exemplo, no Amazon Simple Storage Service (Amazon S3), uma política de recursos é anexada a um bucket do Amazon S3.

Para obter mais informações sobre os esquemas do EventBridge e políticas baseadas em recursos, consulte o seguinte:

- Referência da API REST do Amazon EventBridge Schemas
- Políticas baseadas em identidade e em recurso no Guia do usuário do IAM.

APIs compatíveis para políticas baseadas em recursos

É possível usar as seguintes APIs com políticas baseadas em recursos para o registro do esquema do EventBridge.

- `DescribeRegistry`
- `UpdateRegistry`
- `DeleteRegistry`
- `ListSchemas`
- `SearchSchemas`
- `DescribeSchema`
- `CreateSchema`
- `DeleteSchema`
- `UpdateSchema`
- `ListSchemaVersions`
- `DeleteSchemaVersion`
- `DescribeCodeBinding`
- `GetCodeBindingSource`
- `PutCodeBinding`

Exemplo de política que concede todas as ações compatíveis a uma conta da AWS

Para o registro do esquema do EventBridge, é preciso sempre anexar uma política baseada em recursos a um registro. Para conceder acesso a um esquema, o ARN do esquema e o ARN do registro na política são especificados.

Para conceder a um usuário acesso a todas as APIs disponíveis para os esquemas do EventBridge, use uma política semelhante à seguinte, substituindo a `"Principal"` pela ID da conta à qual você deseja conceder acesso.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Test",
            "Effect": "Allow",
            "Action": [
                "schemas:*"
            ],
            "Principal": {
                "AWS": [
                    "109876543210"
                ]
            },
            "Resource": [
                "arn:aws:schemas:us-east-1:012345678901:registry/default",
                "arn:aws:schemas:us-east-1:012345678901:schema/default*"
            ]
        }
    ]
}
```

Exemplo de política que concede ações somente leitura a uma conta da AWS

O exemplo a seguir concede acesso a uma conta somente para as APIs somente para a leitura dos esquemas do EventBridge.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Test",
```

```
            "Effect": "Allow",
            "Action": [
                "schemas:DescribeRegistry",
                "schemas:ListSchemas",
                "schemas:SearchSchemas",
                "schemas:DescribeSchema",
                "schemas:ListSchemaVersions",
                "schemas:DescribeCodeBinding",
                "schemas:GetCodeBindingSource"
            ],
            "Principal": {
                "AWS": [
                    "109876543210"
                ]
            },
            "Resource": [
                "arn:aws:schemas:us-east-1:012345678901:registry/default",
                "arn:aws:schemas:us-east-1:012345678901:schema/default*"
            ]
        }
    ]
}
```

Exemplo de política que concede todas as ações a uma organização

É possível usar políticas baseadas em recursos com o registro do esquema do EventBridge para conceder acesso a uma organização. Para obter mais informações, consulte o Guia do usuário do AWS Organizations. O exemplo a seguir concede à organização um ID de acesso `o-a1b2c3d4e5` ao registro do esquema.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Test",
            "Effect": "Allow",
            "Action": [
                "schemas:*"
            ],
            "Principal": "*",
            "Resource": [
                "arn:aws:schemas:us-east-1:012345678901:registry/default",
                "arn:aws:schemas:us-east-1:012345678901:schema/default*"
```

```
            ],
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-a1b2c3d4e5"
                    ]
                }
            }
        }
    ]
}
```

Referência de permissões do Amazon EventBridge

Para especificar uma ação em uma política do EventBridge, use o prefixo `events:` seguido do nome da operação da API, conforme mostrado no exemplo a seguir.

```
"Action": "events:PutRule"
```

Para especificar várias ações do em uma única declaração, separe-as com vírgulas, conforme a seguir.

```
"Action": ["events:action1", "events:action2"]
```

Para especificar várias ações, também é possível usar curingas. Por exemplo, é possível especificar todas as ações cujo nome começa com a palavra `"Put"`, da seguinte forma:

```
"Action": "events:Put*"
```

Para especificar todas as ações de API do `*`, use o curinga da seguinte forma:

```
"Action": "events:*"
```

A tabela a seguir lista as operações de API do EventBridge e as ações correspondentes que podem ser especificadas em uma política do IAM.

Operação da API do EventBridge	Permissões obrigatórias	Descrição
DeleteRule	`events:DeleteRule`	Necessária para excluir uma regra.
DescribeEventBus	`events:DescribeEventBus`	Necessário para listar as contas com permissão para gravar eventos no barramento de eventos da conta atual.
DescribeRule	`events:DescribeRule`	Necessária para listar os detalhes sobre uma regra.

Operação da API do EventBridge	Permissões obrigatórias	Descrição
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.
ListTagsForResource	`events:ListTagsForResource`	Necessária para listar todas as tags associadas a um recurso EventBridge. Atualmente, somente as regras podem ser marcadas com tags.
ListTargetsByRule	`events:ListTargetsByRule`	Necessária para listar todos os destinos associados a uma regra.
PutEvents	`events:PutEvents`	Necessária para adicionar eventos personalizados que podem ser vinculados a regras.
PutPermission	`events:PutPermission`	Necessário para conceder a uma outra conta a permissão para gravar eventos no barramento de eventos padrão da conta.

Operação da API do EventBridge	Permissões obrigatórias	Descrição
PutRule	`events:PutRule`	Necessária para criar ou atualizar uma regra.
PutTargets	`events:PutTargets`	Necessária para adicionar destinos a uma regra.
RemovePermission	`events:RemovePermission`	Necessário para revogar a permissão de uma outra conta para gravar eventos no barramento de eventos padrão da conta.
RemoveTargets	`events:RemoveTargets`	Necessária para remover um destino de uma regra.
TestEventPattern	`events:TestEventPattern`	Necessária para testar um evento padrão em um determinado evento.

Uso de condições de política do IAM para controle de acesso refinado

Ao conceder permissões, é possível usar a linguagem da política do IAM para especificar as condições de quando uma política deverá entrar em vigor. Por exemplo, é possível ter uma política que será aplicada somente após uma data específica.

Uma condição em uma política consiste em pares de chave/valor. As chaves de condição não fazem distinção entre maiúsculas e minúsculas.

Caso especifique várias condições ou chaves em uma única condição, todas as condições e chaves devem ser atendidas para que o EventBridge conceda permissão. Caso especifique uma única condição com vários valores para uma chave, o EventBridge concede permissão caso um dos valores seja atendido.

É possível usar espaço reservado ou variáveis de política ao especificar as condições. Para obter mais informações, consulte Variáveis de política no Guia do usuário do IAM. Para obter mais informações sobre como especificar condições em uma linguagem de política do IAM, consulte Condição no Guia do usuário do IAM.

Por padrão, os usuários e perfis do IAM não podem acessar os eventos em sua conta. Para consumir eventos, um usuário deve ser autorizado para a ação de API `PutRule`. Se você conceder permissão a um usuário ou perfil do IAM para a ação `events:PutRule`, em sua respectiva política, eles poderão criar uma regra que corresponda a determinados eventos. No entanto, para que a regra seja útil, o usuário também deve ter permissões para a ação `events:PutTargets` porque, se quiser que a regra faça mais do que publicar uma métrica do CloudWatch, também deve adicionar um destino a uma regra.

Você pode fornecer uma condição na declaração de política do usuário ou perfil do IAM que permita criar uma regra que só corresponda a um conjunto específico de origens e tipos de detalhes. Para conceder acesso a origens e tipos específicos de eventos, use as chaves de condição `events:source` e `events:detail-type`.

De modo semelhante, você pode fornecer uma condição na declaração de política do usuário ou perfil do IAM que permita criar uma regra que só corresponda a um recurso específico em suas contas. Para conceder acesso a um recurso específico, use a chave de condição `events:TargetArn`.

O exemplo a seguir é uma política que permite que os usuários acessem todos os eventos, exceto os eventos do Amazon EC2 no EventBridge, usando uma declaração de negação na ação da API `PutRule`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyPutRuleForAllEC2Events",
            "Effect": "Deny",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2"
                }
            }
        }
    ]
}
```

Chaves de condição do EventBridge

A tabela a seguir mostra as chaves de condição e os pares de chaves e valores que podem ser usados em uma política no EventBridge.

Chave de condição	Par de chave/valor	Tipos de avaliação
aws:SourceAccount	A conta na qual a regra especificada por `aws:SourceArn` existe.	ID da conta, nulo
aws:SourceArn	O ARN da regra que está enviando o evento.	ARN, nulo
events:creatorAccount	`"events:creatorAccount":" creatorAccount "` Para *`creatorAccount`* , use o ID da conta que criou a regra. Use essa condição para autorizar chamadas de API em regras de uma conta específica.	creatorAccount, Null

Chave de condição	Par de chave/valor	Tipos de avaliação
events:detail-type	`"events:detail-type":" detail-type "` Em que *detail-type* é a string literal para o campo detail-type (tipo de detalhe) do evento, como `"AWS API Call via CloudTrail"` e `"EC2 Instance State-change Notification"` .	Tipo de detalhe, nulo
events: detail.eventTypeCode	`"events:detail.eventTypeCode":" eventTypeCode "` Em *eventTypeCode* , use a string literal do campo detail.eventTypeCode do evento, como `"AWS_ABUSE_DOS_REPORT"` .	eventTypeCode, Null
eventos: detail.service	`"events:detail.service":" service"` Em *serviço*, use a string literal do campo detail.service do evento, como `"ABUSE"`.	serviço, Null

Chave de condição	Par de chave/valor	Tipos de avaliação
eventos: detail.userIdentity.principalId	`"events:detail.userIdentity.principalId":" principal-id "` Em *principal-id* é a string literal do campo detail.userIdentity.principalId do evento com o tipo de detalhe `"AWS API Call via CloudTrail"` como `"AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName."` .	Id principal, nulo
events:eventBusInvocation	`"events:eventBusInvocation":" boolean"` Para *booleani*, use "verdadeiro" quando uma regra envia um evento para um destino que é um barramento de eventos em outra conta. Use "false" quando uma chamada de API `PutEvents` for usada.	eventBusInvocation, nula
events:ManagedBy	Usado internamente pelos serviços da AWS. Se uma regra for criada por um serviço da AWS em seu nome, o valor será o nome da entidade principal do serviço que criou a regra.	Não se destina ao uso nas políticas do cliente.

Chave de condição	Par de chave/valor	Tipos de avaliação
events:source	`"events:source":" source "` Em que *origem* é a string literal para o campo de origem do evento, como `"aws.ec2"` e `"aws.s3"`. Para ver mais valores possíveis para *origem*, consulte os eventos de exemplo em Eventos de AWS serviços.	Origem, nulo
events:TargetArn	`"events:TargetArn" :" target-arn "` Para *target-arn* , use o ARN do destino para a regra como, por exemplo, `"arn:aws:lambda:*:*:function:*"`	ArrayOfARN, nulo

Para obter um exemplo de declarações de políticas para o EventBridge, consulte Como gerenciar permissões de acesso aos seus recursos do Amazon EventBridge.

Tópicos

- Especificações do EventBridge Pipes
- Exemplo: como usar a condição creatorAccount
- Exemplo: como usar a condição eventBusInvocation
- Exemplo: como limitar o acesso a uma origem específica
- Exemplo: como definir várias origens que podem ser usadas em um padrão de evento individualmente
- Exemplo: como definir uma origem e um DetailType que podem ser usados em um padrão de evento
- Exemplo: como verificar se a origem está definida no padrão de evento

- Exemplo: como definir uma lista de origens permitidas em um padrão de evento com várias origens
- Exemplo: como limitar o acesso PutRule por detail.service
- Exemplo: como limitar o acesso PutRule por detail.eventTypeCode
- Exemplo: garantir que somente eventos do AWS CloudTrail para chamadas de API de um determinado PrincipalId sejam permitidos
- Exemplo: como limitar o acesso a destinos

Especificações do EventBridge Pipes

O EventBridge Pipes não é compatível com nenhuma chave adicional de condição de política do IAM.

Exemplo: como usar a condição **`creatorAccount`**

O exemplo de declaração de política a seguir mostra como usar a condição `creatorAccount` em uma política para permitir a criação de regras somente se a conta especificada como `creatorAccount` for a conta que criou a regra.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForOwnedRules",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEqualsIfExists": {
                    "events:creatorAccount": "${aws:PrincipalAccount}"
                }
            }
        }
    ]
}
```

Exemplo: como usar a condição **`eventBusInvocation`**

A `eventBusInvocation` indica se a invocação se origina de um destino entre contas ou de uma solicitação de API `PutEvents`. O valor é verdadeiro quando a invocação resulta de uma regra que inclui um destino entre contas, como quando o destino é um barramento de eventos em outra conta. O valor é falso quando a invocação resulta de uma solicitação de API `PutEvents`. O exemplo a seguir indica uma invocação de um destino entre contas.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossAccountInvocationEventsOnly",
      "Effect": "Allow",
      "Action": "events:PutEvents",
      "Resource": "*",
      "Condition": {
        "BoolIfExists": {
          "events:eventBusInvocation": "true"
        }
      }
    }
  ]
}
```

Exemplo: como limitar o acesso a uma origem específica

Os exemplos de política a seguir podem ser anexados a um usuário do IAM. A política A permite a ação de API `PutRule` para todos os eventos, enquanto a Política B permite `PutRule` somente se o padrão de evento da regra que está sendo criada corresponder a eventos do Amazon EC2.

Política A: permitir todos os eventos

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForAllEvents",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*"
        }
```

```
    ]
    }
```

Política B: permitir eventos apenas a partir do Amazon EC2

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForAllEC2Events",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2"
                }
            }
        }
    ]
}
```

`EventPattern` é um argumento obrigatório para `PutRule`. Portanto, se o usuário com a Política B chamar `PutRule` com um padrão de evento como o seguinte.

```
{
    "source": [ "aws.ec2" ]
}
```

A regra será criada, pois a política permite essa origem específica, que é, `"aws.ec2"`. No entanto, se o usuário com a Política B chamar `PutRule` com um padrão de evento como o seguinte, a criação da regra será negada porque a política não permite essa origem específica: ou seja, `"aws.s3"`.

```
{
    "source": [ "aws.s3" ]
}
```

Basicamente, o usuário com a política B só pode criar uma regra que corresponda aos eventos originados pelo Amazon EC2. Portanto, eles só podem acessar os eventos do Amazon EC2.

Consulte a tabela a seguir para obter uma comparação das Políticas A e B.

Padrão de evento	Permitido pela Política A	Permitido pela Política B
`{ "source": [ "aws.ec2" ] }`	Sim	Sim
`{ "source": [ "aws.ec2", "aws.s3" ] }`	Sim	Não (A origem aws.s3 não é permitida)
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sim	Sim
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sim	Não (a origem deve ser especificado)

Exemplo: como definir várias origens que podem ser usadas em um padrão de evento individualmente

A política a seguir permite que um usuário ou perfil do IAM crie uma regra na qual a origem no `EventPattern` é o Amazon EC2 ou o Amazon ECS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsEC2OrECS",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": [ "aws.ec2", "aws.ecs" ]
                }
            }
        }
    ]
}
```

Consulte a tabela a seguir para obter exemplos de padrões de eventos que seriam permitidos ou negados por essa política.

Padrão de evento	Permitido pela política
`{ "source": [ "aws.ec2" ] }`	Sim
`{ "source": [ "aws.ecs" ] }`	Sim
`{ "source": [ "aws.s3" ] }`	Não
`{ "source": [ "aws.ec2", "aws.ecs" ] }`	Não

Padrão de evento	Permitido pela política

<pre>{
"detail-type": ["AWS API Call via CloudTrail"]
}</pre>
	Não

Exemplo: como definir uma origem e um **`DetailType`** que podem ser usados em um padrão de evento

A política a seguir permite apenas eventos da origem `aws.ec2` com `DetailType` igual a `EC2 instance state change notification`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid":
 "AllowPutRuleIfSourceIsEC2AndDetailTypeIsInstanceStateChangeNotification",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2",
                    "events:detail-type": "EC2 Instance State-change Notification"
                }
            }
        }
    ]
}
```

Consulte a tabela a seguir para obter exemplos de padrões de eventos que seriam permitidos ou negados por essa política.

Padrão de evento	Permitido pela política
<pre>{</pre>	Não

Padrão de evento	Permitido pela política
`{ "source": [ "aws.ec2" ] }`	
`{ "source": [ "aws.ecs" ] }`	Não
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sim
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance Health Failed" ] }`	Não
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	Não

Exemplo: como verificar se a origem está definida no padrão de evento

A seguinte política permite que usuários somente criem regras com `EventPatterns`, que devem ter o campo de origem. Em outras palavras, um usuário ou perfil do IAM não pode criar uma regra com um `EventPattern` que não forneça uma origem específica.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "AllowPutRuleIfSourceIsSpecified",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "Null": {
                    "events:source": "false"
                }
            }
        }
    ]
}
```

Consulte a tabela a seguir para obter exemplos de padrões de eventos que seriam permitidos ou negados por essa política.

Padrão de evento	Permitido pela política
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Sim
`{ "source": [ "aws.ecs", "aws.ec2" ] }`	Sim
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	Não

Exemplo: como definir uma lista de origens permitidas em um padrão de evento com várias origens

A política a seguir permite criar que usuários criem regras com `EventPatterns` que podem ter várias origens. Cada origem no padrão de evento deve ser um membro da lista fornecida na condição. Ao usar a condição `ForAllValues`, certifique-se de que pelo menos um dos itens na lista de condições esteja definido.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsSpecifiedAndIsEitherS3OrEC2OrBoth",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "events:source": [ "aws.ec2", "aws.s3" ]
                },
                "Null": {
                    "events:source": "false"
                }
            }
        }
    ]
}
```

Consulte a tabela a seguir para obter exemplos de padrões de eventos que seriam permitidos ou negados por essa política.

Padrão de evento	Permitido pela política
`{ "source": [ "aws.ec2" ] }`	Sim
`{ "source": [ "aws.ec2", "aws.s3" ]`	Sim

Padrão de evento	Permitido pela política
`}`	
`{ "source": [ "aws.ec2", "aws.autoscaling" ] }`	Não
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	Não

Exemplo: como limitar o acesso **`PutRule`** por **`detail.service`**

Você pode restringir um usuário ou perfil do IAM a criar regras apenas para eventos que têm um determinado valor no campo `events:details.service`. O valor de `events:details.service` não é necessariamente o nome de um serviço da AWS.

Esta condição de política é útil ao trabalhar com eventos do AWS Health que são relacionados a segurança ou abuso. Ao usar essa condição de política, você pode limitar o acesso a esses alertas confidenciais apenas aos usuários que precisam vê-los.

Por exemplo, a política a seguir permite a criação de regras apenas para eventos nos quais o valor de `events:details.service` é `ABUSE`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleEventsWithDetailServiceEC2",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail.service": "ABUSE"
```

```
                }
            }
        }
    ]
}
```

Exemplo: como limitar o acesso **PutRule** por **detail.eventTypeCode**

Você pode restringir um usuário ou perfil do IAM a criar regras apenas para eventos que têm um determinado valor no campo `events:details.eventTypeCode`. Esta condição de política é útil ao trabalhar com eventos do AWS Health que são relacionados a segurança ou abuso. Ao usar essa condição de política, você pode limitar o acesso a esses alertas confidenciais apenas aos usuários que precisam vê-los.

Por exemplo, a política a seguir permite a criação de regras apenas para eventos nos quais o valor de `events:details.eventTypeCode` é `AWS_ABUSE_DOS_REPORT`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleEventsWithDetailServiceEC2",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail.eventTypeCode": "AWS_ABUSE_DOS_REPORT"
                }
            }
        }
    ]
}
```

Exemplo: garantir que somente eventos do AWS CloudTrail para chamadas de API de um determinado **PrincipalId** sejam permitidos

Todos os eventos do AWS CloudTrail têm o PrincipalId do usuário que fez a chamada de API no caminho `detail.userIdentity.principalId` de um evento. Com a ajuda da chave de condição `events:detail.userIdentity.principalId`, é possível limitar o acesso de

usuários ou perfis do IAM aos eventos do CloudTrail apenas para os próximos eventos de uma conta específica.

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleOnlyForCloudTrailEventsWhereUserIsASpecificIAMUser",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail-type": [ "AWS API Call via CloudTrail" ],
                    "events:detail.userIdentity.principalId":
 [ "AIDAJ45Q7YFFAREXAMPLE" ]
                }
            }
        }
    ]
}
```

Consulte a tabela a seguir para obter exemplos de padrões de eventos que seriam permitidos ou negados por essa política.

Padrão de evento	Permitido pela política
`{ "detail-type": [ "AWS API Call via CloudTrail" ] }`	Não
`{ "detail-type": [ "AWS API Call via CloudTrail" ], "detail.userIdentity.principalId": [ "AIDAJ45Q7YFFAREXAMPLE" ] }`	Sim

Padrão de evento	Permitido pela política

<pre>{
"detail-type": ["AWS API Call via CloudTrail"],
"detail.userIdentity.principalId": ["AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName"]
}</pre>
	Não

Exemplo: como limitar o acesso a destinos

Se um usuário ou perfil do IAM tiver a permissão `events:PutTargets`, poderá adicionar qualquer destino na mesma conta às regras que tem permissão para acessar. A seguinte política limita usuários de adicionarem destinos a apenas uma regra específica: `MyRule` na conta `123456789012`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutTargetsOnASpecificRule",
            "Effect": "Allow",
            "Action": "events:PutTargets",
            "Resource": "arn:aws:events:us-east-1:123456789012:rule/MyRule"
        }
    ]
}
```

Para limitar o destino que pode ser adicionado à regra, use a chave de condição `events:TargetArn`. Por exemplo, é possível limitar destinos a funções do Lambda, como no exemplo a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutTargetsOnASpecificRuleAndOnlyLambdaFunctions",
            "Effect": "Allow",
```

```
            "Action": "events:PutTargets",
            "Resource": "arn:aws:events:us-east-1:123456789012:rule/MyRule",
            "Condition": {
                "ArnLike": {
                    "events:TargetArn": "arn:aws:lambda:*:*:function:*"
                }
            }
        }
    ]
}
```

Usar perfis vinculados a serviço do EventBridge

O Amazon EventBridge usa funções vinculadas a serviços do AWS Identity and Access Management (IAM). O perfil vinculado a serviço é um tipo exclusivo de perfil do IAM vinculado diretamente ao EventBridge. Os perfis vinculados a serviços são predefinidos pelo EventBridge e incluem todas as permissões que o serviço requer para chamar outros serviços da AWS em seu nome.

Tópicos

- Usando funções para criar segredos para destinos de API
- Usando funções para descoberta de esquemas

Usando funções para criar segredos para destinos de API

O Amazon EventBridge utiliza perfis vinculados a serviço do AWS Identity and Access Management (IAM). O perfil vinculado a serviço é um tipo exclusivo de perfil do IAM vinculado diretamente ao EventBridge. Os perfis vinculados a serviços são predefinidos pelo EventBridge e incluem todas as permissões que o serviço requer para chamar outros serviços da AWS em seu nome.

Uma função vinculada ao serviço facilita a configuração do EventBridge porque você não precisa adicionar as permissões necessárias manualmente. EventBridge define as permissões de suas funções vinculadas ao serviço e, a menos que definido de outra forma, somente EventBridge pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões. Essa política não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus recursos do EventBridge, pois você não pode remover por engano as permissões de acesso aos recursos.

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 de função vinculada ao serviço EventBridge

EventBridgeusa a função vinculada ao serviço chamada AWSServiceRoleForAmazonEventBridgeApiDestinations— Permite o acesso aos segredos do Secrets Manager criados por. EventBridge

A função vinculada ao serviço AWSServiceRoleForAmazonEventBridgeApiDestinations confia nos seguintes serviços para aceitar a função:

- `apidestinations.events.amazonaws.com`

A política de permissões de função chamada AmazonEventBridgeApiDestinationsServiceRolePolítica EventBridge permite concluir as seguintes ações nos recursos especificados:

- Ação: `create, describe, update and delete secrets; get and put secret values` em `secrets created for all connections by EventBridge`

Você deve configurar permissões para permitir que seus usuários, grupos ou perfis criem, editem ou excluam um perfil vinculado ao serviço. Para ter mais informações, consulte Permissões de função vinculada a serviços no Guia do usuário do IAM.

Crie uma função vinculada ao serviço para o EventBridge

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você cria uma conexão naAWS Management Console, na ou na AWS APIAWS CLI, EventBridge cria a função vinculada ao serviço para você.

> **Important**
>
> Esse perfil vinculado ao serviço pode aparecer em sua conta se você concluiu uma ação em outro serviço que usa os atributos compatíveis com esse perfil. Se você estava usando o EventBridge serviço antes de 11 de fevereiro de 2021, quando ele começou a oferecer suporte a funções vinculadas ao serviço, EventBridge criou a

AWSServiceRoleForAmazonEventBridgeApiDestinationsfunção em sua conta. Para saber mais, consulte Um novo perfil apareceu na minha Conta da AWS.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, você poderá usar esse mesmo processo para recriar o perfil em sua conta. Quando você cria uma conexão, EventBridge cria a função vinculada ao serviço para você novamente.

Editar uma função vinculada ao serviço para o EventBridge

O EventBridge não permite que você edite a função vinculada ao serviço AWSServiceRoleForAmazonEventBridgeApiDestinations. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição da função usando o IAM. Para obter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço para o EventBridge

Se você não precisar mais usar um atributo 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 suo perfil vinculado ao serviço para excluí-la manualmente.

Limpar um perfil vinculado ao serviço

Antes de usar o IAM para excluir um perfil vinculado ao serviço, você deverá excluir qualquer recurso usado pelo perfil.

> **Note**
>
> Se o serviço EventBridge estiver usando a função quando você tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Como excluir recursos do EventBridge usados pelo AWSServiceRoleForAmazonEventBridgeApiDestinations (console)

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. Em Integrações, escolha destinos de API e, em seguida, escolha a guia Conexões.

3. Escolha a conexão e, em seguida, escolha Excluir.

Como excluir recursos do EventBridge usados pelo AWSServiceRoleForAmazonEventBridgeApiDestinations (CLI da AWS)

- Use o seguinte comando:`delete-connection`.

Como excluir os recursos do EventBridge usados pelo AWSServiceRoleForAmazonEventBridgeApiDestinations (API)

- Use o seguinte comando:`DeleteConnection`.

Excluir manualmente o perfil vinculado ao serviço

Use o console do IAM, a AWS CLI ou a API da AWS para excluir o perfil vinculado ao serviço AWSServiceRoleForAmazonEventBridgeApiDestinations. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões suportadas por funções vinculadas ao serviço do EventBridge

O EventBridge oferece suporte a funções vinculadas a serviços em todas as regiões nas quais o serviço estiver disponível. Para obter mais informações, consulte Regiões e endpoints da AWS.

Usando funções para descoberta de esquemas

O Amazon EventBridge utiliza perfis vinculados a serviço do AWS Identity and Access Management (IAM). O perfil vinculado a serviço é um tipo exclusivo de perfil do IAM vinculado diretamente ao EventBridge. Os perfis vinculados a serviços são predefinidos pelo EventBridge e incluem todas as permissões que o serviço requer para chamar outros serviços da AWS em seu nome.

Uma função vinculada ao serviço facilita a configuração do EventBridge porque você não precisa adicionar as permissões necessárias manualmente. EventBridge define as permissões de suas funções vinculadas ao serviço e, a menos que definido de outra forma, somente EventBridge pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões. Essa política não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus recursos do EventBridge, pois você não pode remover por engano as permissões de acesso aos recursos.

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 de função vinculada ao serviço EventBridge

EventBridgeusa a função vinculada ao serviço chamada AWSServiceRoleForSchemas— Concede permissões às regras gerenciadas criadas por Amazon EventBridge esquemas.

A função vinculada ao serviço AWSServiceRoleForSchemas confia nos seguintes serviços para aceitar a função:

- `schemas.amazonaws.com`

A política de permissões de função nomeada AmazonEventBridgeSchemasServiceRolePolicyEventBridgepermite concluir as seguintes ações nos recursos especificados:

- Ação: `put, enable, disable, and delete rules; put and remove targets; list targets per rule` em `all managed rules created by EventBridge`

Você deve configurar permissões para permitir que seus usuários, grupos ou perfis criem, editem ou excluam um perfil vinculado ao serviço. Para ter mais informações, consulte Permissões de função vinculada a serviços no Guia do usuário do IAM.

Crie uma função vinculada ao serviço para o EventBridge

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você conduz uma descoberta de esquema naAWS Management Console, na ou na AWS APIAWS CLI, EventBridge cria a função vinculada ao serviço para você.

> **Important**
>
> Esse perfil vinculado ao serviço pode aparecer em sua conta se você concluiu uma ação em outro serviço que usa os atributos compatíveis com esse perfil. Se você estava usando o EventBridge serviço antes de 27 de novembro de 2019, quando ele começou a oferecer suporte a funções vinculadas ao serviço, EventBridge criou a

AWSServiceRoleForSchemasfunção em sua conta. Para saber mais, consulte Um novo perfil apareceu na minha Conta da AWS.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, você poderá usar esse mesmo processo para recriar o perfil em sua conta. Quando você conduz uma descoberta de esquema, EventBridge cria a função vinculada ao serviço para você novamente.

Editar uma função vinculada ao serviço para o EventBridge

O EventBridge não permite que você edite a função vinculada ao serviço AWSServiceRoleForSchemas. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição da função usando o IAM. Para obter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Excluir uma função vinculada ao serviço para o EventBridge

Se você não precisar mais usar um atributo 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 suo perfil vinculado ao serviço para excluí-la manualmente.

Limpar um perfil vinculado ao serviço

Antes de usar o IAM para excluir um perfil vinculado ao serviço, você deverá excluir qualquer recurso usado pelo perfil.

> **Note**
>
> Se o serviço EventBridge estiver usando a função quando você tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Como excluir recursos do EventBridge usados pelo AWSServiceRoleForSchemas (console)

1. Abra o EventBridge console da Amazon em https://console.aws.amazon.com/events/.
2. Em Ônibus, escolha Ônibus para eventos e, em seguida, escolha um ônibus para eventos.

3. Escolha Parar a descoberta.

Como excluir recursos do EventBridge usados pelo AWSServiceRoleForSchemas (CLI da AWS)

- Use o seguinte comando:`delete-discoverer`.

Como excluir os recursos do EventBridge usados pelo AWSServiceRoleForSchemas (API)

- Use o seguinte comando:`DeleteDiscoverer`.

Excluir manualmente o perfil vinculado ao serviço

Use o console do IAM, a AWS CLI ou a API da AWS para excluir o perfil vinculado ao serviço AWSServiceRoleForSchemas. Para obter mais informações, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões suportadas por funções vinculadas ao serviço do EventBridge

O EventBridge oferece suporte a funções vinculadas a serviços em todas as regiões nas quais o serviço estiver disponível. Para obter mais informações, consulte Regiões e endpoints da AWS.

Registrando chamadas de Amazon EventBridge API usando AWS CloudTrail

Amazon EventBridge é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou um AWS service (Serviço da AWS). CloudTrail captura todas as chamadas de API EventBridge como eventos. As chamadas capturadas incluem chamadas do EventBridge console e chamadas de código para as operações EventBridge da API. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita EventBridge, o endereço IP do qual a solicitação foi feita, quando foi feita e detalhes adicionais.

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 credenciais de usuário.
- Se a solicitação foi feita em nome de um usuário do Centro de Identidade do IAM.
- 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 AWS service (Serviço da AWS).

CloudTrail está ativo Conta da AWS quando você cria a conta e você tem acesso automático ao histórico de CloudTrail eventos. O histórico de CloudTrail eventos fornece um registro visível, pesquisável, baixável e imutável dos últimos 90 dias de eventos de gerenciamento registrados em um. Região da AWS Para obter mais informações, consulte Trabalhando com o histórico de CloudTrail eventos no Guia AWS CloudTrail do usuário. Não há CloudTrail cobrança pela visualização do histórico de eventos.

Para um registro contínuo dos eventos dos Conta da AWS últimos 90 dias, crie uma trilha ou um armazenamento de dados de eventos do CloudTrailLake.

CloudTrail trilhas

Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Todas as trilhas criadas usando o AWS Management Console são multirregionais. Só é possível criar uma trilha de região única ou de várias regiões usando a AWS CLI. É recomendável criar uma trilha multirregional porque você captura todas as atividades Regiões da AWS em sua conta. Se você criar uma trilha de região única, poderá visualizar somente os eventos registrados na Região da

AWS da trilha. Para obter mais informações sobre trilhas, consulte Criar uma trilha para a Conta da AWS e Criar uma trilha para uma organização no Guia do usuário do AWS CloudTrail .

Você pode entregar uma cópia dos seus eventos de gerenciamento contínuos para o bucket do Amazon S3 sem nenhum custo CloudTrail criando uma trilha. No entanto, há cobranças de armazenamento do Amazon S3. Para obter mais informações sobre CloudTrail preços, consulte AWS CloudTrail Preços. Para receber informações sobre a definição de preço do Amazon S3, consulte Definição de preço do Amazon S3.

CloudTrail Armazenamentos de dados de eventos em Lake

CloudTrail O Lake permite que você execute consultas baseadas em SQL em seus eventos. CloudTrail O Lake converte eventos existentes no formato JSON baseado em linhas para o formato Apache ORC. O ORC é um formato colunar de armazenamento otimizado para recuperação rápida de dados. Os eventos são agregados em armazenamentos de dados de eventos, que são coleções imutáveis de eventos baseados nos critérios selecionados com a aplicação de seletores de eventos avançados. Os seletores que você aplica a um armazenamento de dados de eventos controlam quais eventos persistem e estão disponíveis para você consultar. Para obter mais informações sobre o CloudTrail Lake, consulte Trabalhando com o AWS CloudTrail Lake no Guia AWS CloudTrail do Usuário.

CloudTrail Os armazenamentos e consultas de dados de eventos em Lake incorrem em custos. Ao criar um armazenamento de dados de eventos, você escolhe a opção de preço que deseja usar para ele. A opção de preço determina o custo para a ingestão e para o armazenamento de eventos, e o período de retenção padrão e máximo para o armazenamento de dados de eventos. Para obter mais informações sobre CloudTrail preços, consulte AWS CloudTrail Preços.

EventBridge eventos de dados em CloudTrail

Os eventos de dados fornecem informações sobre as operações de recursos realizadas em um recurso (por exemplo, leitura ou gravação em um objeto do Amazon S3). Elas também são conhecidas como operações de plano de dados. Eventos de dados geralmente são atividades de alto volume. Por padrão, CloudTrail não registra eventos de dados. O histórico de CloudTrail eventos não registra eventos de dados.

Há cobranças adicionais para eventos de dados. Para obter mais informações sobre CloudTrail preços, consulte AWS CloudTrail Preços.

Você pode registrar eventos de dados para os tipos de EventBridge recursos usando o CloudTrail console ou AWS CLI as operações CloudTrail da API. Para obter mais informações sobre como registrar eventos de dados em log, consulte Registrar eventos de dados com o AWS Management Console e Registrar eventos de dados com a AWS Command Line Interface no Guia do usuário do AWS CloudTrail .

A tabela a seguir lista os tipos de EventBridge recursos para os quais você pode registrar eventos de dados. A coluna Tipo de evento de dados (console) mostra o valor a ser escolhido na lista Tipo de evento de dados no CloudTrail console. A coluna de valor resources.type mostra o **`resources.type`** valor, que você especificaria ao configurar seletores de eventos avançados usando as APIs ou. AWS CLI CloudTrail A CloudTrail coluna Data APIs logged to mostra as chamadas de API registradas CloudTrail para o tipo de recurso.

Tipo de evento de dados (console)	valor resources.type	APIs de dados registradas em CloudTrail
Ônibus para eventos	`AWS::Events::Event Bus`	• DescribeEventBus
Regra de ônibus de eventos	`AWS::Events::Rule`	• DeleteRule • DescribeRule • DisableRule • EnableRule • ListRuleNamesByTarget • ListRules • ListTargetsByRule • PutRule • PutTargets • RemoveTargets • TestEventPattern
Tubo	`AWS::Pipes::Pipe`	• CreatePipe • DeletePipe • DescribePipe • ListPipes

Tipo de evento de dados (console)	valor resources.type	APIs de dados registradas em CloudTrail
		• StartPipe • StopPipe • UpdatePipe

É possível configurar seletores de eventos avançados para filtrar os campos `eventName`, `readOnly` e `resources.ARN` para registrar somente os eventos que são importantes para você. Para obter mais informações sobre esses campos, consulte AdvancedFieldSelector na Referência de API do AWS CloudTrail .

EventBridge eventos de gerenciamento em CloudTrail

Os eventos de gerenciamento fornecem informações sobre as operações de gerenciamento que são realizadas nos recursos do seu Conta da AWS. Elas também são conhecidas como operações de plano de controle. Por padrão, CloudTrail registra eventos de gerenciamento.

Amazon EventBridge registra todas as operações do plano de EventBridge controle como eventos de gerenciamento. Para ver uma lista das operações do plano de Amazon EventBridge controle EventBridge registradas CloudTrail, consulte a Referência da Amazon EventBridge API.

EventBridge exemplos de eventos

Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a operação de API solicitada, a data e a hora da operação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, os eventos não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra um CloudTrail evento que demonstra a `PutRule` operação.


```
{
  "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"
}
```


Para obter informações sobre o conteúdo do CloudTrail registro, consulte o conteúdo do CloudTrail registro no Guia AWS CloudTrail do usuário.

CloudTrail entradas de registro para ações tomadas pelo EventBridge Pipes

EventBridge O Pipes assume a função do IAM fornecida ao ler eventos de fontes, invocar enriquecimentos ou invocar destinos. Para CloudTrail entradas relacionadas às ações realizadas em sua conta em todos os enriquecimentos, destinos e fontes do Amazon SQS, Kinesis e DynamoDB, os campos e incluirão. `sourceIPAddress` `invokedBy` `pipes.amazonaws.com`

Exemplo de entrada de CloudTrail registro para todos os enriquecimentos, destinos e fontes do Amazon SQS, Kinesis e DynamoDB

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "...",
    "arn": "arn:aws:sts::111222333444:assumed-role/...",
    "accountId": "111222333444",
    "accessKeyId": "...",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "...",
        "arn": "...",
        "accountId": "111222333444",
        "userName": "userName"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-09-22T21:41:15Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "pipes.amazonaws.com"
  },
  "eventTime": ",,,",
  "eventName": "...",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "pipes.amazonaws.com",
  "userAgent": "pipes.amazonaws.com",
  "requestParameters": {
    ...
  },
  "responseElements": null,
  "requestID": "...",
  "eventID": "...",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "...",
  "eventCategory": "Management"
}
```

Para todas as outras fontes, o `sourceIPAddress` campo das entradas de CloudTrail registro terá um endereço IP dinâmico e não deve ser usado para nenhuma integração ou categorização de eventos. Além disso, essas entradas não terão o campo `invokedBy`.

Entrada de CloudTrail registro de amostra para todas as outras fontes

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    ...
  },
  "eventTime": ",,,",
  "eventName": "...",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Python-httplib2/0.8 (gzip)",
}
```

Validação de conformidade no Amazon EventBridge

Auditores externos, como SOC, PCI, FedRAMP e HIPAA, avaliam a segurança e a conformidade dos serviços da AWS como parte de vários programas de conformidade da AWS.

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 Download de relatórios no AWS Artifact.

Sua responsabilidade em relação à compatibilidade ao usar o EventBridge é determinada pela confidencialidade dos seus dados, pelos objetivos de compatibilidade da 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: as considerações de arquitetura e etapas para a implantação de ambientes de linha de base focados em compatibilidade e segurança na AWS.
- Architecting for HIPAA Security and Compliance Whitepaper : como as empresas podem usar a AWS para criar aplicações em conformidade com a HIPAA.
- Recursos de conformidade da AWS: uma coleção de manuais e guias.
- Avaliação de recursos com regras no Guia do desenvolvedor do AWS Config: informações sobre como o AWS Config avalia a conformidade das configurações de seus recursos com práticas internas, diretrizes do setor e regulamentos.
- AWS Security Hub: uma visão abrangente do estado da segurança na AWS que ajuda você a conferir a compatibilidade com os padrões e as práticas recomendadas do setor de segurança.

Resiliência do Amazon EventBridge

A infraestrutura global da AWS é criada com base em regiões da AWS e zonas de disponibilidade. As regiões da AWS As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, throughput elevada 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 EventBridge

Como um serviço gerenciado, o Amazon EventBridge 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.

Chamadas de API publicadas da AWS são usadas para acessar o EventBridge Scheduler 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.

É possível chamar essas operações de API de qualquer local da rede e usar oferece políticas de acesso baseadas em recursos, que podem incluir restrições com base no endereço IP de origem. Também é possível usar políticas do EventBridge para controlar o acesso de Amazon Virtual Private Cloud (Amazon VPC) endpoints ou de VPCs específicas. Efetivamente, isto isola o acesso à rede para um determinado recurso do EventBridge somente da VPC específica dentro da rede da AWS.

Análise de configuração e vulnerabilidade no Amazon EventBridge

A configuração e os controles de TI são uma responsabilidade compartilhada entre a AWS e você, nosso cliente. Para obter mais informações, consulte o modelo de responsabilidade compartilhada da AWS.

Monitorando a Amazon EventBridge

EventBridge envia métricas para a Amazon a CloudWatch cada minuto para tudo, desde o número de eventos correspondentes até o número de vezes que um alvo é invocado por uma regra.

O vídeo a seguir analisa o EventBridge comportamento de monitoramento e auditoria por meio de CloudWatch: Monitoramento e auditoria de eventos

Tópicos

EventBridge métricas

O namespace `AWS/Events` inclui as métricas a seguir.

Para as métricas que usam a contagem como uma unidade, soma e SampleCount tendem a ser as estatísticas mais úteis.

As métricas que especificam somente a `RuleName` dimensão se referem ao barramento de eventos padrão. As métricas que especificam as `RuleName` dimensões `EventBusName` e se referem a um barramento de eventos personalizado.

Métrica	Descrição	Dimensões	Unidades
`DeadLetterInvocations`	O número de vezes em que um destino de uma regra não é invocado em resposta a um evento. Inclui as invocações que causariam o acionamento da mesma regra novamente, resultando em um loop infinito.	RuleName	Contagem
`Events`	O número de eventos de parceiros ingeridos pelo EventBridge.	EventSourceName	Contagem
`FailedInvocations`	O número de invocações que apresentaram falha permanentemente. Não inclui as	RuleName	Contagem

Métrica	Descrição	Dimensões	Unidades
	invocações que foram repetidas ou que tiveram êxito após uma tentativa repetida. Também não considera invocações com falha que são consideradas em `DeadLetterInvocations`. **Note** EventBridge só envia essa métrica para CloudWatch se ela não for zero.		
`Invocations`	O número de vezes em que um destino é invocado por uma regra em resposta a um evento. Isso inclui invocações com êxito e com falha, mas não inclui tentativas limitadas ou repetidas até que elas apresentem falha permanentemente. Não inclui `DeadLetterInvocations`. **Note** EventBridge só envia essa métrica para CloudWatch se ela não for zero.	Nenhum, RuleName	Contagem
`InvocationAttempts`	Número de vezes que EventBridge tentou invocar um alvo.	Nenhum	Contagem
`InvocationsCreated`	O número total de invocações criadas em resposta a cada evento. Essa métrica é frequentemente usada para monitorar a utilização do limite de aceleração de invocações em transações por segundo da cota de serviço. EventBridge	Nenhum	Contagem

Métrica	Descrição	Dimensões	Unidades
`InvocationsFailedToBeSentToDlq`	O número de invocações que não puderam ser movidas para uma fila de mensagens não entregues. Os erros de fila de mensagens não entregues podem ocorrer devido a erros de permissões, recursos indisponíveis ou limites de tamanho. **Note** EventBridge só envia essa métrica para CloudWatch se ela não for zero.	RuleName	Contagem
`IngestionToInvocationCompleteLatency`	O tempo gasto desde a ingestão do evento até a conclusão da primeira tentativa de invocação com êxito.	EventBusName, Nenhuma, RuleName	Milissegundos
`IngestionToInvocationStartLatency`	O tempo para processar eventos, medido desde o momento em que um evento é ingerido até EventBridge a primeira invocação de um alvo.	EventBusName, Nenhuma, RuleName	Milissegundos
`InvocationsSentToDlq`	O número de invocações que são movidas para uma fila de mensagens não entregues. **Note** EventBridge só envia essa métrica para CloudWatch se ela não for zero.	RuleName	Contagem

Métrica	Descrição	Dimensões	Unidades
`MatchedEvents`	Se EventBusName ou EventSourceName for especificado, o número de eventos que corresponderam a qualquer regra. Se RuleName for especificado, o número de eventos que corresponderam a uma regra específica.	EventBusName, EventSourceName, RuleName	Contagem
`RetryInvocationAttempts`	Número de vezes que a invocação do destino foi repetida. **Note** EventBridge só envia essa métrica para CloudWatch se ela não for zero.	Nenhum	Contagem
`SuccessfulInvocationAttempts`	O número de vezes que o destino foi invocado com êxito.	Nenhum	Contagem
`ThrottledRules`	O número de vezes que a execução da regra foi limitada. As invocações dessas regras podem ser adiadas. Para obter mais informações, consulte Limite de controle de utilização de invocações em transações por segundo em ???.	EventBusName, Nenhuma, RuleName	Contagem
`TriggeredRules`	O número de regras que foram executadas e corresponderam a qualquer evento. Você não verá essa métrica CloudWatch até que uma regra seja acionada.	EventBusName, Nenhuma, RuleName	Contagem

EventBridge PutEvents métricas

O namespace `AWS/Events` inclui as métricas a seguir que pertencem às solicitações de API `PutEvents`.

Para as métricas que usam a contagem como uma unidade, soma e SampleCount tendem a ser as estatísticas mais úteis.

Métrica	Descrição	Dimensões	Unidades
`PutEventsApproximateCallCount`	O número aproximado de solicitações `PutEvents` recebidas.	Nenhum	Contagem
`PutEventsApproximateFailedCount`	O número aproximado de solicitações `PutEvents` com falha.	Nenhum	Contagem
`PutEventsApproximateSuccessCount`	O número de solicitações `PutEvents` recebidas com êxito.	Nenhum	Contagem
`PutEventsApproximateThrottledCount`	Número de solicitações `PutEvents` rejeitadas devido ao controle de utilização.	Nenhum	Contagem
`PutEventsEntriesCount`	O número de entradas de eventos contidas em uma solicitação `PutEvents`.	Nenhum	Contagem
`PutEventsFailedEntriesCount`	O número de entradas de eventos contidas em uma solicitação `PutEvents` que não foi ingerida.	Nenhum	Contagem

Métrica	Descrição	Dimensões	Unidades
`PutEvents Latency`	O tempo gasto por solicitação `PutEvents`.	Nenhum	Milissegundos
`PutEvents RequestSize`	O tamanho da solicitação `PutEvents`.	Nenhum	Bytes

EventBridge PutPartnerEvents métricas

O namespace `AWS/Events` inclui as métricas a seguir que pertencem às solicitações de API `PutPartnerEvents`.

> **Note**
>
> EventBridge inclui apenas métricas relacionadas a PutPartnerEventssolicitações em contas de parceiros de SaaS que enviam eventos. Para mais informações, consulte ???.

Para as métricas que usam a contagem como uma unidade, soma e SampleCount tendem a ser as estatísticas mais úteis.

Métrica	Descrição	Dimensões	Unidades
`PutPartnerEventsApproximateCallCount`	O número aproximado de solicitações `PutPartnerEvents` recebidas.	Nenhum	Contagem
`PutPartnerEventsApproximateFailedCount`	O número aproximado de solicitações `PutPartnerEvents` com falha.	Nenhum	Contagem

Métrica	Descrição	Dimensões	Unidades
PutPartnerEventsApproximateThrottledCount	Número de solicitações PutPartnerEvents rejeitadas devido ao controle de utilização.	Nenhum	Contagem
PutPartnerEventsApproximateSuccessCount	O número de solicitações PutPartnerEvents recebidas com êxito.	Nenhum	Contagem
PutPartnerEventsEntriesCount	O número de entradas de eventos contidas em uma solicitação PutPartnerEvents .	Nenhum	Contagem
PutPartnerEventsFailedEntriesCount	O número de entradas de eventos contidas em uma solicitação PutPartnerEvents que não foi ingerida.	Nenhum	Contagem
PutPartnerEventsLatency	O tempo gasto por solicitação PutPartnerEvents .	Nenhum	Milissegundos

Dimensões para EventBridge métricas

EventBridge as métricas têm dimensões ou atributos classificáveis, listados abaixo.

Dimensão	Descrição
EventBusName	Filtra as métricas disponíveis pelo nome do barramento de eventos.

Dimensão	Descrição
`EventSourceName`	Filtra as métricas disponíveis pelo nome da origem de eventos do parceiro.
`RuleName`	Filtra as métricas disponíveis pelo nome da regra.

Solução de problemas da Amazon EventBridge

Você pode usar as etapas nesta seção para solucionar problemas na Amazon EventBridge.

Tópicos

Minha regra foi executada, mas minha função do Lambda não foi invocada

Um dos motivos pelos quais sua função do Lambda pode não ser executada é se não tiver as permissões corretas.

Para verificar as permissões da função do Lambda

1. Usando o AWS CLI, execute o seguinte comando com sua função e sua AWS região:

```
aws lambda get-policy --function-name MyFunction --region us-east-1
```

   Você verá a saída a seguir.

```
{
    "Policy": "{\"Version\":\"2012-10-17\",
    \"Statement\":[
        {\"Condition\":{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:events:us-
east-1:123456789012:rule/MyRule\"}},
        \"Action\":\"lambda:InvokeFunction\",
        \"Resource\":\"arn:aws:lambda:us-east-1:123456789012:function:MyFunction\",
        \"Effect\":\"Allow\",
        \"Principal\":{\"Service\":\"events.amazonaws.com\"},
        \"Sid\":\"MyId\"}
    ],
    \"Id\":\"default\"}"
}
```

2. Se vir a seguinte mensagem:

```
A client error (ResourceNotFoundException) occurred when calling the GetPolicy
 operation: The resource you requested does not exist.
```

   Ou se vir a saída, mas não conseguir localizar events.amazonaws.com como uma entidade confiável na política, execute o seguinte comando:

```
aws lambda add-permission \
--function-name MyFunction \
--statement-id MyId \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/MyRule
```

3. Se a saída tiver um `SourceAccount` campo, será preciso removê-lo. Uma `SourceAccount` configuração EventBridge impede a possibilidade de invocar a função.

> **Note**
>
> Se a política estiver incorreta, você poderá editar a regra no EventBridge console removendo-a e adicionando-a novamente à regra. O EventBridge console então define as permissões corretas no destino.
> Se estiver usando uma versão ou um alias específico do Lambda, deverá incluir o parâmetro `--qualifier` nos comandos `aws lambda get-policy` e `aws lambda add-permission`, conforme mostrado no segunite comando
>
> ```
> aws lambda add-permission \
> --function-name MyFunction \
> --statement-id MyId \
> --action 'lambda:InvokeFunction' \
> --principal events.amazonaws.com \
> --source-arn arn:aws:events:us-east-1:123456789012:rule/MyRule
> --qualifier alias or version
> ```

Acabei de criar/modificar uma regra, mas ela não corresponde a um evento de teste

Quando fizer uma alteração em uma regra ou em seus destinos, os futuros eventos poderão não começar imediatamente ou parar a correspondência com regras novas ou atualizadas. Permita um curto período para que as alterações entrem em vigor.

Se os eventos ainda não coincidirem após um curto período de tempo `TriggeredRulesInvocations`, verifique as CloudWatch métricas e `FailedInvocations` sua regra. Para obter mais informações sobre essas métricas, consulte Monitoramento da Amazon EventBridge.

Se a regra se destina a corresponder a um evento de um AWS serviço, faça o seguinte:

- Use a ação `TestEventPattern` para testar o padrão de evento se sua regra corresponder a um evento de teste. Para obter mais informações, consulte TestEventPatterna Amazon EventBridge API Reference.
- Use o Sandbox no EventBridge console.

Minha regra não foi executada no momento em que eu especifiquei no `ScheduleExpression`

Verifique se definiu o agendamento para a regra ser acionada no fuso horário UTC. Se a `ScheduleExpression` estiver correta, siga as etapas em Acabei de criar/modificar uma regra, mas ela não corresponde a um evento de teste.

Minha regra não foi acionada no momento que eu esperava

EventBridge executa as regras dentro de um minuto da hora de início que você definiu. A contagem regressiva para execução começa assim que você cria a regra.

> **Note**
>
> As regras programadas têm um tipo de entrega `guaranteed`, o que significa que os eventos serão acionados para cada horário esperado pelo menos uma vez.

É possível usar uma expressão cron para invocar destinos em um horário especificado. Para criar uma regra que seja executada a cada quatro horas no 0º minuto, faça o seguinte:

- No EventBridge console, você usa a expressão `0 0/4 * * ? *` cron.
- Usando o AWS CLI, você usa a expressão`cron(0 0/4 * * ? *)`.

Por exemplo, para criar uma regra chamada `TestRule` que é executada a cada 4 horas usando o AWS CLI, use o comando a seguir.

```
aws events put-rule --name TestRule --schedule-expression 'cron(0 0/4 * * ? *)'
```

Para executar uma regra a cada cinco minutos, use a seguinte expressão cron.

```
aws events put-rule --name TestRule --schedule-expression 'cron(0/5 * * * ? *)'
```

A melhor resolução para uma EventBridge regra que usa uma expressão cron é de um minuto. Sua regra programada é acionada dentro desse minuto, mas não no 0º segundo preciso.

Como EventBridge os serviços de destino são distribuídos, pode haver um atraso de vários segundos entre o momento em que a regra programada é executada e o momento em que o serviço de destino executa a ação no recurso de destino.

Minha regra corresponde às chamadas de API de serviço AWS global, mas não foi executada

AWS serviços globais, como IAM e Amazon Route 53, só estão disponíveis na região Leste dos EUA (Norte da Virgínia), portanto, eventos de chamadas de AWS API de serviços globais só estão disponíveis nessa região. Para ter mais informações, consulte Eventos de AWS serviços.

O perfil do IAM associado à minha regra está sendo ignorado quando a regra é executada

EventBridge usa apenas funções do IAM para regras que enviam eventos para streams do Kinesis. Para regras que chamem funções do Lambda e tópicos do Amazon SNS, é preciso conceder permissões baseadas em recursos.

Certifique-se de que seus AWS STS endpoints regionais estejam habilitados, para que EventBridge possa usá-los ao assumir a função do IAM que você forneceu. Para obter mais informações, consulte Ativação e desativação AWS STS em uma AWS região no Guia do usuário do IAM.

Minha regra tem um padrão de evento que deveria corresponder a um recurso, mas nenhum evento corresponde

A maioria dos serviços AWS trata dois pontos (:) ou barra (/) como o mesmo caractere nos Amazon Resource Names (ARNs)., mas EventBridge usa uma correspondência exata nos padrões e regras de eventos. Certifique-se de usar os caracteres de ARN corretos ao criar padrões de evento para que eles correspondam à sintaxe do ARN no evento para corresponder.

Alguns eventos, como eventos de chamada de AWS API de CloudTrail, não têm nada no campo de recursos.

A entrega do meu evento no destino sofreu um atraso

EventBridge tenta entregar um evento a um alvo por até 24 horas, exceto em cenários em que seu recurso alvo é restrito. A primeira tentativa é feita assim que o evento chega no stream de eventos.

Se o serviço de destino estiver com problemas, reagenda EventBridge automaticamente outra entrega. Se já passaram 24 horas desde a chegada do evento, para EventBridge de tentar entregar o evento e publica a `FailedInvocations` métrica em CloudWatch. É recomendada a configuração de uma DLQ para armazenar eventos que não puderam ser entregues com êxito a um destino. Para mais informações, consulte Usando filas de cartas mortas para processar eventos não entregues.

Alguns eventos nunca foram entregues em meu destino

Se o alvo de uma EventBridge regra for restrito por um período prolongado, EventBridge talvez não tente novamente a entrega. Por exemplo, se o destino não estiver provisionado para lidar com o tráfego de eventos de entrada e o serviço de destino estiver limitando as solicitações feitas em seu nome, talvez não tente EventBridge fazer a entrega novamente EventBridge .

Minha regra foi executada mais de uma vez em resposta a um evento

Em casos raros, a mesma regra pode ser acionada mais de uma vez para um único evento ou horário programado, ou o mesmo destino pode ser invocado mais de uma vez para uma determinada regra acionada.

Como evitar loops infinitos

Em EventBridge, é possível criar uma regra que leva a loops infinitos, onde a regra é executada repetidamente. Se tiver uma regra que cause um loop infinito, reescreva-a para que as ações realizadas pela regra não correspondam à mesma regra.

Por exemplo, uma regra que detecta que as ACLs foram alteradas em um bucket do Amazon S3 e, em seguida, executa um software para alterá-las para um novo estado causa um loop infinito. Uma maneira de resolver isso é reescrever a regra para que ela corresponda apenas às ACLs que estão em um estado ruim.

Um loop infinito pode rapidamente causar cobranças acima do esperado. Recomendamos que você use o orçamento, que alerta quando as cobranças excederem o limite especificado. Para obter mais informações, consulte Gerenciamento de seus custos com orçamentos.

Os eventos não são entregues na fila de destino do Amazon SQS

Se sua fila do Amazon SQS estiver criptografada, deverá criar uma chave KMS gerenciada pelo cliente e incluir a seguinte seção de permissão na sua política de chave do KMS. Para obter mais informações, consulte Configurando AWS KMS permissões.

```
{
  "Sid": "Allow EventBridge to use the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

Minha regra é executada, mas eu não vejo nenhuma mensagem publicada no meu tópico do Amazon SNS

Cenário 1

É preciso de permissão para que as mensagens sejam publicadas em seu tópico do Amazon SNS. Use o comando a seguir usando o AWS CLI, substituindo us-east-1 pela sua região e usando o ARN do tópico.

```
aws sns get-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic"
```

Para ter a permissão correta, seus atributos de política são semelhantes aos seguintes.

```
"{\"Version\":\"2012-10-17\",
\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",
\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":\"*\"},
\"Action\":[\"SNS:Subscribe\",
```

```
\"SNS:ListSubscriptionsByTopic\",
\"SNS:DeleteTopic\",
\"SNS:GetTopicAttributes\",
\"SNS:Publish\",
\"SNS:RemovePermission\",
\"SNS:AddPermission\",
\"SNS:SetTopicAttributes\"],
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\",
\"Condition\":{\"StringEquals\":{\"AWS:SourceOwner\":\"123456789012\"}}},{\"Sid\":
\"Allow_Publish_Events\",
\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}]}"
```

Se não vê `events.amazonaws.com` com permissão de `Publish` em sua política, primeiro copie a política atual e adicione a seguinte declaração à lista de declarações.

```
{\"Sid\":\"Allow_Publish_Events\",
\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}
```

Em seguida, defina os atributos do tópico usando o comando AWS CLI, use o seguinte comando.

```
aws sns set-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic" --attribute-name Policy --attribute-
value NEW_POLICY_STRING
```

> **Note**
>
> Se a política estiver incorreta, você também poderá editar a regra no EventBridge console removendo-a e adicionando-a novamente à regra. EventBridge define as permissões corretas no alvo.

Cenário 2

Se seu tópico do SNS estiver criptografado, deverá incluir a seção a seguir em sua política de chave do KMS.

```
{
  "Sid": "Allow EventBridge to use the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*"
}
```

Meu tópico do Amazon SNS ainda tem permissões para, EventBridge mesmo depois que eu excluí a regra associada ao tópico do Amazon SNS

Quando você cria uma regra com o Amazon SNS como alvo, EventBridge adiciona permissão ao seu tópico do Amazon SNS em seu nome. Se você excluir a regra logo após criá-la, EventBridge talvez não remova a permissão do seu tópico do Amazon SNS. Se isso acontecer, será possível remover a permissão do tópico ao usar o comando `aws sns set-topic-attributes`. Para obter informações sobre permissões baseadas em recursos para enviar eventos, consulte Como usar políticas baseadas em recursos para esquemas do Amazon EventBridge.

Com quais chaves de condição do IAM posso usar EventBridge?

EventBridge suporta as chaves de condição AWS-wide (consulte IAM e chaves de contexto de AWS STS condição no Guia do usuário do IAM), além das chaves listadas emUso de condições de política do IAM para controle de acesso refinado.

Como posso saber quando EventBridge as regras foram violadas?

Você pode usar o alarme a seguir para notificá-lo quando suas EventBridge regras forem violadas.

Para criar um alarme para alertar quando as regras são violadas

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

2. Escolha Create Alarm. No painel CloudWatch Métricas por categoria, escolha Métricas de eventos.
3. Na lista de métricas, selecione FailedInvocations.
4. Acima do gráfico, escolha Estatística, Soma.
5. Para Período, selecione um valor, por exemplo, 5 minutos. Selecione Next (Próximo).
6. Em Limite de alarme, em Nome, digite um nome exclusivo para o alarme, por exemplo myFailedRules. Em Descrição, digite uma descrição do alarme, por exemplo, Regras não estão entregando eventos para destinos.
7. Para is, escolha >= and 1. Para for, digite 10.
8. Em Actions (Ações), em Whenever this alarm (Sempre que este alarme), escolha State is ALARM (Estado é ALARME).
9. Em Send notification to (Enviar notificação para), selecione um tópico existente do Amazon SNS ou crie outro. Para criar um novo tópico do , selecione Nova lista. Digite um nome para o novo tópico do Amazon SNS, por exemplo:. myFailedRules
10. Para E-mail, digite uma lista de endereços de e-mail separados por vírgulas a serem notificados quando o alarme mudar para o estado ALARME.
11. Escolha Create Alarm.

Cotas do Amazon EventBridge

Há cotas para a maioria dos aspectos do EventBridge.

Tópicos

> **Note**
>
> Para obter uma lista das cotas do Agendador do EventBridge, consulte Cotas para o Agendador do EventBridge no Guia do Usuário do Agendador do EventBridge.

Cotas do EventBridge

O EventBridge tem as cotas a seguir.

O console Service Quotas fornece informações sobre as cotas do EventBridge. Além de visualizar as cotas padrão, você pode usar o console de Cotas de serviço para solicitar aumentos de cota para aquelas que podem ser ajustadas.

Nome	Padrão	Ajustá	Descrição
Destinos da API	Cada região compatível: 3.000	Sim	O número máximo de destinos de API por conta, por região.
Conexões	Cada região compatível: 3.000	Sim	O número máximo de conexões por conta por região.

Nome	Padrão	Ajustá	Descrição
Limite de controle de utilização CreateEndpoint em transações por segundo	Cada região compatível: 5 por segundo	Não	O número máximo de solicitações por segundo para API de CreateSecret. Solicitações adicionais são limitadas.
Limite de controle de utilização DeleteEndpoint em transações por segundo	Cada região compatível: 5 por segundo	Não	O número máximo de solicitações por segundo para API de DeleteEndpoint. Solicitações adicionais são limitadas.
Endpoints	Cada região com suporte: 100	Sim	O número máximo de endpoints por conta, por região.
Tamanho da política de barramento de eventos	Cada região com suporte: 10.240	Sim	Tamanho máximo da política, em caracteres. Esse tamanho de política aumenta cada vez que você concede acesso a outra conta. Você pode ver sua política atual e seu tamanho usando a API de DescribeEventBus.
Barramentos de eventos	Cada região com suporte: 100	Sim	Máximo de barramentos de eventos por conta.
Tamanho do padrão de eventos	Cada região compatível: 2.048	Sim	Tamanho máximo de um padrão de evento, em caracteres.

Nome	Padrão	Ajustá	Descrição
Limite de controle de utilização das invocações em transações por segundo	us-east-1: 18.750 por segundo us-east-2: 4.500 por segundo us-west-1: 2.250 por segundo us-west-2: 18.750 por segundo af-south-1: 750 por segundo ap-northeast-1: 2.250 por segundo ap-northeast-3: 750 por segundo ap-southeast-1: 2.250 por segundo ap-southeast-2: 2.250 por segundo ap-southeast-3: 750 por segundo eu-central-1: 4.500 por segundo eu-south-1: 750 por segundo eu-west-1: 18.750 por segundo	Sim	Uma invocação é um evento que correspon de a uma regra e que é enviado para os destinos das regras. Após o limite ser atingido, as invocaçõe s são limitadas; ou seja, elas ainda ocorrem, mas com atraso.

Nome	Padrão	Ajustá	Descrição
	eu-west-2: 2.250 por segundo Cada uma das outras regiões compatíveis: 1.100 por segundo		
Número de regras	af-south-1: 100 eu-south-1: 100 Cada uma das outras regiões compatíveis: 300	Sim	Número máximo de regras que uma conta pode ter por barramento de eventos

Nome	Padrão	Ajustá	Descrição
Limite de controle de utilização do PutEvents em transações por segundo	us-east-1: 10.000 por segundo us-east-2: 2.400 por segundo us-west-1: 1.200 por segundo us-west-2: 10.000 por segundo af-south-1: 400 por segundo ap-northeast-1: 1.200 por segundo ap-northeast-3: 400 por segundo ap-southeast-1: 1.200 por segundo ap-southeast-2: 1.200 por segundo ap-southeast-3: 400 por segundo eu-central-1: 2.400 por segundo eu-south-1: 400 por segundo eu-west-1: 10.000 por segundo	Sim	Número máximo de solicitações por segundo para API PutEvents. Solicitações adicionais são limitadas.

Nome	Padrão	Ajustá	Descrição
	eu-west-2: 1.200 por segundo Cada uma das outras regiões compatíveis: 600 por segundo		
Taxa de invocações por destino de API	Cada região compatível: 300 por segundo	Sim	O número máximo de invocações por segundo a serem enviadas para cada endpoint de destino da API por conta por região. Depois que a cota é atingida, as futuras invocações para este endpoint da API são limitadas. As invocaçõe s ainda ocorrerão, mas estão atrasadas.
Destinos por regra	Cada região compatível: 5	Não	Número máximo de destinos que podem ser associados a uma regra
Limite de controle de uso em transaçõe s por segundo	Cada região compatível: 50 por segundo	Sim	Número máximo de solicitações por segundo para todas as operações da API de EventBrid ge exceto PutEvents. Solicitações adicionais são limitadas

Nome	Padrão	Ajustá	Descrição
Limite de controle de uso do UpdateEndPoint em transações por segundo	Cada região compatível: 5 por segundo	Não	O número máximo de solicitações por segundo para a API de UpdateEndpoint. Solicitações adicionais são limitadas.

Além disso, o EventBridge tem as seguintes cotas que não são gerenciadas pelo console Service Quotas.

Nome	Padrão	Descrição
Barramentos de eventos	Cada região com suporte: 100	Máximo de barramentos de eventos por conta.
Tamanho da política de barramento de eventos	Cada região compatível: 10.240	Tamanho máximo da política, em caracteres. Esse tamanho de política aumenta cada vez que você concede acesso a outra conta. Você pode ver sua política atual e seu tamanho usando a API `DescribeEventBus` .
Tamanho do padrão de eventos	Cada região compatível: 2.048	Tamanho máximo de um padrão de evento, em caracteres. Isto é ajustável em até 4.096 caracteres. Se tiver requisitos para limites máximos mais altos, entre em contato com o suporte.
Regras que contêm curingas	Cada região compatível: 30 regras por barramento de eventos	O número máximo de regras, por barramento de eventos e por conta, que podem conter filtros de eventos que incluem curingas. Esta cota não pode ser ajustada. Para obter mais informações sobre o uso de curingas em padrões de eventos, consulte ???.

Nome	Padrão	Descrição
Níveis com descoberta de esquemas	Cada região compatível: 255 níveis	O número máximo de níveis que a descobert a do esquema inferirá eventos que estão aninhados. Todos os eventos acima de 255 níveis são ignorados.

Cotas do PutPartnerEvents por região

Se você tiver requisitos para limites máximos mais altos, entre em contato com o suporte.

Regiões	Transações por segundo
• AWS GovCloud (Oeste dos EUA) • AWS GovCloud (Leste dos EUA) • 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 (Mumbai) • 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)	O PutPartnerEvents tem um limite flexível de 1.400 solicitações de throughput por segundo e 3.600 solicitaç ões de intermitência por segundo por padrão em todas as regiões.

Regiões	Transações por segundo
• Europa (Paris) • Europa (Estocolmo) • Europa (Milão) • América do Sul (São Paulo) • China (Ningxia) • China (Pequim)	

Cotas do EventBridge Schema Registry

O EventBridge Schema Registry tem as cotas a seguir.

O console Service Quotas fornece informações sobre as cotas do EventBridge. Além de visualizar as cotas padrão, você pode usar o console de Cotas de serviço para solicitar aumentos de cota para aquelas que podem ser ajustadas.

Nome	Padrão	Ajustá	Descrição
DiscoveredSchemas	Cada região compatível: 200	Sim	O número máximo de esquemas para um registro de esquema descoberto que você pode criar na região atual
Descobridores	Cada região com suporte: 10	Sim	O número máximo de descobridores que podem ser criados na região atual.
Registros	Cada região com suporte: 10	Sim	O número máximo de registros que é possível criar na região atual.
SchemaVersions	Cada região com suporte: 100	Sim	O número máximo de versões que podem ser

Nome	Padrão	Ajustá	Descrição
			criadas por esquema na região atual.
Esquemas	Cada região com suporte: 100	Sim	O número máximo de esquemas por registro que você pode criar na região atual. (Exceto o registro do esquema descoberto)

Cotas do EventBridge Pipes

O EventBridge Pipes tem as cotas a seguir. Se você tiver requisitos para limites máximos mais altos, entre em contato com o suporte.

Recurso	Regiões	Limite padrão
Execuções simultâneas de pipes por conta	• AWS GovCloud (Oeste dos EUA) • AWS GovCloud (Leste dos EUA) • China (Ningxia) • China (Pequim) • Ásia-Pacífico (Osaka) • África (Cidade do Cabo) • Europa (Milão) • Leste dos EUA (Ohio) • Europa (Frankfurt) • Oeste dos EUA (N. da Califórnia) • Europa (Londres) • Ásia-Pacífico (Sydney)	1000

Recurso	Regiões	Limite padrão
	• Ásia-Pacífico (Tóquio) • Ásia-Pacífico (Singapura) • Canadá (Central) • Europa (Paris) • Europa (Estocolmo) • América do Sul (São Paulo) • Ásia-Pacífico (Seul) • Ásia-Pacífico (Mumbai) • Ásia-Pacífico (Hong Kong) • Oriente Médio (Barém) • China (Ningxia) • China (Pequim) • Ásia-Pacífico (Osaka) • África (Cidade do Cabo) • Europa (Milão)	
Execuções simultâneas de pipes por conta	• Leste dos EUA (Norte da Virgínia) • Oeste dos EUA (Oregon) • Europa (Irlanda)	3000
Pipes por conta	Todos	1000

EventBridge Etiquetas da Amazon

Uma tag é um rótulo de atributo personalizado que você atribui ou AWS atribui a um AWS recurso. Em EventBridge, você pode atribuir tags a barramentos de regras e eventos. Cada recurso pode ter um máximo de 50 tags.

Você usa tags para identificar e organizar seus AWS recursos. Muitos AWS serviços oferecem suporte à marcação, então você pode atribuir a mesma tag a recursos de serviços diferentes para indicar que os recursos estão relacionados. Por exemplo, você pode atribuir a mesma tag a uma EventBridge regra que você atribui a uma instância do EC2.

Uma tag tem duas partes:

- Uma chave de tag (por exemplo `CostCenter`, `Environment` ou `Project`).
  - Chaves de tag fazem distinção entre maiúsculas e minúsculas.
  - O comprimento máximo da chave da tag é de 128 caracteres Unicode em UTF-8.
  - Para cada recurso, cada chave de tag deve ser única.
  - Os caracteres permitidos são letras, números, espaços representáveis em UTF-8, além dos seguintes caracteres: . : + = @ _ / - (hífen).
  - O `aws:` prefixo é proibido para tags porque está reservado para AWS uso. Você não pode editar nem excluir chaves nem valores de tags com esse prefixo. As tags com esse prefixo não contam para as tags por limite de recurso.
- Um campo opcional de valor de tag (por exemplo, `111122223333` ou `Production`).
  - Cada chave de tag pode ter apenas um valor.
  - Os valores de tags não diferenciam maiúsculas de minúsculas.
  - Omitir o valor da tag é o mesmo que usar uma string vazia.
  - 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).

> **Tip**
>
> 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.

Você pode usar o EventBridge console, a EventBridge API ou o AWS CLI para adicionar, editar ou excluir tags. Para obter mais informações, consulte:

- TagResource, UntagResource, e ListTagsForResourcena Amazon EventBridge API Reference
- tag-resource, untag-resource e na Referência list-tags-for-resourceAWS CLI
- Trabalhar com o editor de tags no Manual do usuário do Resource Groups

Histórico do documento

A tabela a seguir descreve mudanças importantes em cada versão do Guia do EventBridge usuário da Amazon, a partir de julho de 2019. Para receber notificações sobre atualizações dessa documentação, você poderá se inscrever em um feed RSS.

Alteração	Descrição	Data de lançamento
Políticas AWS gerenciadas atualizadas.	AWS GovCloud (US) Regions somente `AmazonEventBridgeFullAccess` e `AmazonEventBridgeSchemasFullAccess` as políticas não incluem`iam:CreateServiceLinkedRole` , pois não são usadas. • the section called “Atualizações da política”	9 de maio de 2024
Gere AWS CloudFormation modelos a partir de regras e ônibus de eventos.	Agora você pode gerar AWS CloudFormation modelos a partir de seus ônibus e regras de EventBridge eventos existentes da Amazon. • Gere um modelo do AWS CloudFormation a partir de um barramento de eventos do Amazon EventBridge	18 de novembro de 2022
Documentação do EventBridge Pipes lançada.	Agora é possível criar pipes para conectar as origens aos destinos, com filtragem e enriquecimento opcionais. • Pipes	1º de dezembro de 2022
Gere AWS CloudFormation modelos a partir de regras e ônibus de eventos.	Agora você pode gerar AWS CloudFormation modelos a partir de seus ônibus e regras de EventBridge eventos existentes da Amazon. • Gere um modelo do AWS CloudFormation a partir de um barramento de eventos do Amazon EventBridge	18 de novembro de 2022

Alteração	Descrição	Data de lançamento
A AmazonEventBridgePipesFullAccess política foi adicionada.	Fornece acesso total ao Amazon EventBridge Pipes. • EventBridge Políticas gerenciadas específicas para tubos	1º de dezembro de 2022
A AmazonEventBridgePipesReadOnlyAccess política foi adicionada.	Fornece acesso somente para leitura ao Amazon EventBridge Pipes. • EventBridge Políticas gerenciadas específicas para tubos	1º de dezembro de 2022
A AmazonEventBridgePipesOperatorAccess política foi adicionada.	Fornece acesso somente de leitura e de operador (ou seja, a capacidade de parar e começar a executar o Pipes) ao Amazon EventBridge Pipes. • EventBridge Políticas gerenciadas específicas para tubos	1º de dezembro de 2022
Atualizou a CloudWatchEventsFullAccess política.	Atualizado para corresponder `AmazonEventBridgeFullAccess` . • AmazonEventBridgeFullAccess política	1º de dezembro de 2022
Atualizou a CloudWatchEventsReadOnlyAccess política.	Atualizado para corresponder `AmazonEventBridgeReadOnlyAccess` . • AmazonEventBridgeReadOnlyAccess política	1º de dezembro de 2022

Alteração	Descrição	Data de lançamento
Filtragem de conteúdo atualizada em padrões de eventos.	Agora é possível usar `suffix`, `equals-ignore-case` e opções de filtragem `$or` para criar padrões de eventos. • Filtragem de conteúdo nos padrões de EventBridge eventos da Amazon	14 de novembro de 2022
Atualizou a AmazonEventBridgeFullAccess política.	Foram adicionadas as permissões necessárias para usar o EventBridge Schema Registry and EventBridge Scheduler. • AmazonEventBridgeFullAccess política	10 de novembro de 2022
Atualizou a AmazonEventBridgeReadOnlyAccess política.	Agora você pode visualizar as informações do EventBridge Schema Registry e do EventBridge Scheduler. • AmazonEventBridgeReadOnlyAccess política	10 de novembro de 2022
Filtragem de conteúdo atualizada em padrões de eventos.	Agora é possível usar `suffix`, `equals-ignore-case` e opções de filtragem `$or` para criar padrões de eventos. • Filtragem de conteúdo nos padrões de EventBridge eventos da Amazon	14 de novembro de 2022
Atualizou a AmazonEventBridgeFullAccess política.	Foram adicionadas as permissões necessárias para usar o EventBridge Schema Registry and EventBridge Scheduler. • AmazonEventBridgeFullAccess política	10 de novembro de 2022

Alteração	Descrição	Data de lançamento
Atualizou a AmazonEventBridgeReadOnlyAccess política.	Agora você pode visualizar as informações do EventBridge Schema Registry e do EventBridge Scheduler. • AmazonEventBridgeReadOnlyAccess política	10 de novembro de 2022
Atualizou a AmazonEventBridgeReadOnlyAccess política.	Agora é possível visualizar as informações do endpoint. • AmazonEventBridgeReadOnlyAccess política	7 de abril de 2022
Compatibilidade adicionada para endpoints globais.	A Amazon EventBridge agora oferece suporte ao uso de endpoints globais para ajudar a tornar seu aplicativo tolerante a falhas regionais sem custo adicional. Para saber mais, consulte: • Como tornar as aplicações tolerantes a falhas regionais com endpoints globais e replicação de eventos • `CreateEndpoint`	7 de abril de 2022
Foi adicionada a compatibilidade para arquivos e repetições de eventos.	A Amazon EventBridge agora suporta o uso de arquivos para armazenar eventos e replays de eventos para reproduzir os eventos de um arquivo. Para saber mais, consulte: • Arquivamento de eventos da Amazon EventBridge . • `CreateArchive` • `StartReplay`	5 de novembro de 2020

Alteração	Descrição	Data de lançamento
Foi adicionado suporte para filas de mensagens não entregues e política de repetição para destinos.	A Amazon EventBridge agora suporta o uso de filas de mensagens mortas e a definição de uma política de repetição para os alvos. Para saber mais, consulte: • Usando filas de cartas mortas para processar eventos não entregues. • `PutTargets`	12 de outubro de 2020
Foi adicionada a compatibilidade para esquemas de formato JSONSchema Draft4.	A Amazon EventBridge agora oferece suporte a esquemas no formato JSONSchema Draft 4. Agora você também pode exportar esquemas usando a EventBridge API. Para saber mais, consulte: • EventBridge Esquemas da Amazon • `Export`na Referência da API EventBridge Schema Registry.	28 de setembro de 2020
Políticas baseadas em recursos para o Schema Registry EventBridge	O Amazon EventBridge Schema Registry agora oferece suporte a políticas baseadas em recursos. Para obter mais informações, consulte. • Políticas baseadas em recursos para esquemas do Amazon EventBridge • `Policy`na referência da API EventBridge Schema Registry • RegistryPolicy Tipo de recurso no Guia AWS CloudFormation do usuário	30 de abril de 2020

Alteração	Descrição	Data de lançamento
Tags para barramentos de eventos	Esta versão permite que você crie e gerencie tags para barramentos de eventos. É possível adicionar tags ao criar um barramento de eventos e adicionar ou gerenciar tags existentes chamando a API relacionada. Para obter mais informações, consulte. • EventBridge Etiquetas da Amazon • Políticas baseadas em tags • `TagResource` • `UntagResource` • `ListTagsForResource`	24 de fevereiro de 2020
Cotas de serviço aumentadas	A Amazon EventBridge aumentou as cotas para invocações e para. `PutEvents` As cotas variam de acordo com a região e podem ser aumentadas, se necessário.	11 de fevereiro de 2020

Alteração	Descrição	Data de lançamento
Foram adicionados um novo tópico sobre como transformar a entrada de destino e um link para eventos de ajuste de escala automático para aplicações.	Documentação melhorada sobre o transformador de entrada. • Transformação EventBridge de insumos da Amazon • Usar o transformador de entrada para extrair dados de um evento e inserir esses dados no destino • Tutorial: usar o transformador de entrada para personalizar o que o EventBridge aprova para o evento de destino Foi adicionado um link para os eventos de ajuste de escala automático para aplicações. • Eventos de Application Auto Scaling e EventBridge • Eventos de AWS serviços	20 de dezembro de 2019
Filtragem baseada em conteúdo		19 de dezembro de 2019
Links adicionados para exemplos de eventos do Amazon Augmented AI	Foi adicionado um link para o tópico Amazon Augmented AI no Guia do Desenvolvedor da SageMaker Amazon que fornece exemplos de eventos para o Amazon Augmented AI. Para obter mais informações, consulte. • Usar eventos no Amazon Augmented AI • Eventos de AWS serviços	13 de dezembro de 2019

Alteração	Descrição	Data de lançamento
Links adicionados para exemplos de eventos do Amazon Chime.	Foi adicionado um link para o tópico do Amazon Chime que fornece exemplos de eventos para esse serviço. Para obter mais informações, consulte. • Automatizando o Amazon Chime com EventBridge • Eventos de AWS serviços	12 de dezembro de 2019
EventBridge Esquemas da Amazon	Agora você pode gerenciar esquemas e gerar vinculações de código para eventos na Amazon. EventBridge Para obter mais informações, consulte. • EventBridge Esquemas da Amazon • EventBridge Referência da API de esquemas • EventSchemas Referência do tipo de recurso em AWS CloudFormation	1º de dezembro de 2019
AWS CloudFormation suporte para ônibus de eventos	AWS CloudFormation agora oferece suporte ao EventBus recurso. Ele também suporta o EventBusName parâmetro nos recursos EventBusPolicy e Rule. Para obter mais informações, consulte Amazon EventBridge Resource Type Reference.	7 de outubro de 2019
Novo serviço	Lançamento inicial da Amazon EventBridge.	11 de julho de 2019

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.