aws

Guia do Desenvolvedor

Amazon Data Firehose

Amazon Data Firehose: Guia do Desenvolvedor

Table of Contents

O Amazon Data Firehose era conhecido anteriormente como Amazon Kinesis Data Firehose

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.

O que é o Amazon Data Firehose?

O Amazon Data Firehose é um serviço totalmente gerenciado para fornecer dados de streaming em tempo real para destinos como Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon Service, Amazon Serverless, Splunk e qualquer endpoint HTTP personalizado ou endpoints HTTP de propriedade de provedores de serviços terceirizados compatíveis, incluindo Datadog LogicMonitor, Dynatrace, MongoDB, New Relic, Coralogix e Elastic. OpenSearch OpenSearch Com o Amazon Data Firehose, você não precisa criar aplicativos nem gerenciar recursos. Você configura seus produtores de dados para enviar dados para o Amazon Data Firehose, e ele entrega automaticamente os dados para o destino que você especificou. Você também pode configurar o Amazon Data Firehose para transformar seus dados antes de entregá-los.

Para obter mais informações sobre soluções de AWS big data, consulte Big Data on AWS. Para obter mais informações sobre as soluções de dados em streaming da AWS , consulte O que são dados em streaming?

> **Note**
>
> Observe a mais recente solução AWS de dados de streaming para Amazon MSK, que fornece AWS CloudFormation modelos em que os dados fluem por produtores, armazenamento de streaming, consumidores e destinos.

Aprenda os principais conceitos

Ao começar a usar o Amazon Data Firehose, você pode se beneficiar da compreensão dos seguintes conceitos:

Stream Firehose

A entidade subjacente do Amazon Data Firehose. Você usa o Amazon Data Firehose criando um stream do Firehose e enviando dados para ele. Para obter mais informações, consulte Crie um stream do Firehose e Enviar dados para um stream do Firehose.

registro

Os dados de interesse que seu produtor de dados envia para um stream do Firehose. Um registro pode ter, no máximo, 1000 KB.

produtor de dados

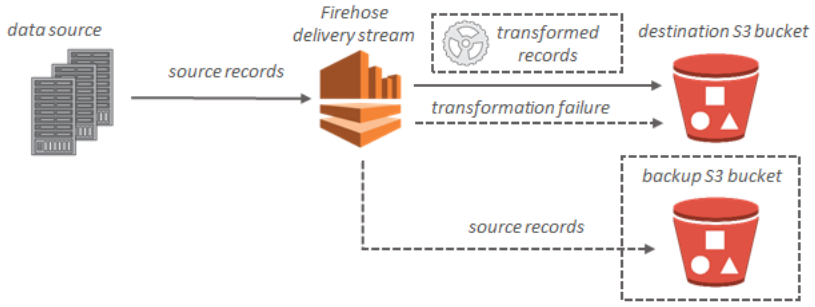
Os produtores enviam discos para as transmissões do Firehose. Por exemplo, um servidor web que envia dados de log para um stream do Firehose é um produtor de dados. Você também pode configurar seu stream do Firehose para ler automaticamente os dados de um stream de dados existente do Kinesis e carregá-los nos destinos. Para ter mais informações, consulte Enviar dados para um stream do Firehose.

tamanho e intervalo do buffer

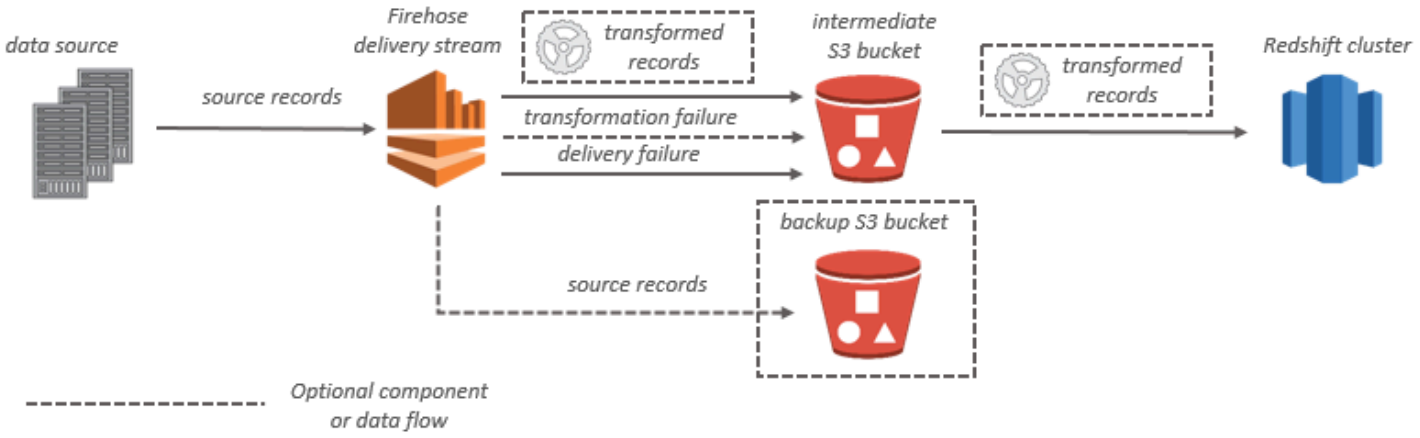
O Amazon Data Firehose armazena os dados de streaming recebidos em um determinado tamanho ou por um determinado período de tempo antes de entregá-los aos destinos. Buffer Sizeestá em MBs e Buffer Interval está em segundos.

Entenda o fluxo de dados no Amazon Data Firehose

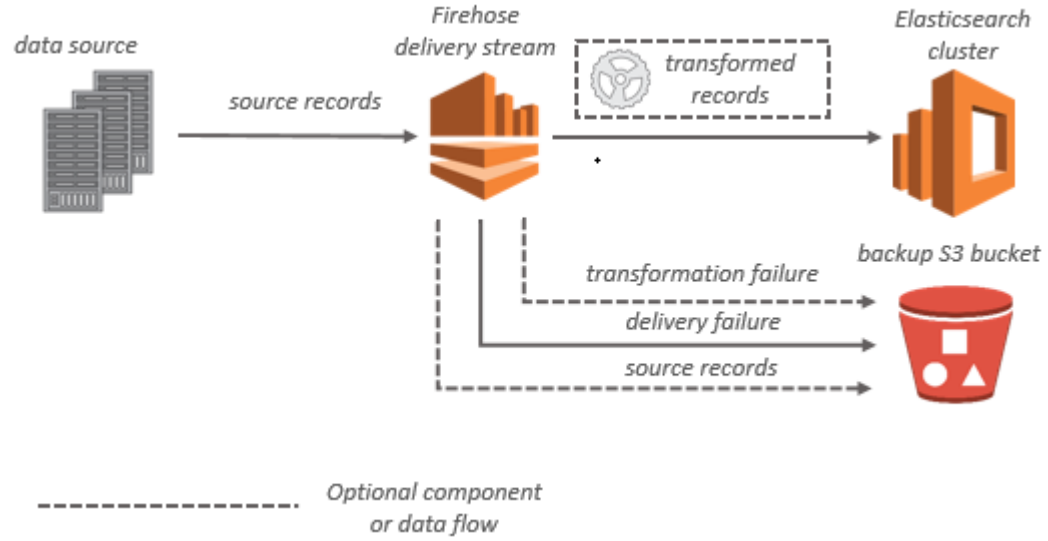
Para destinos do Amazon S3, os dados em streaming são entregues no bucket do S3. Se a transformação de dados estiver habilitada, você também poderá fazer backup dos dados da fonte em outro bucket do Amazon S3.



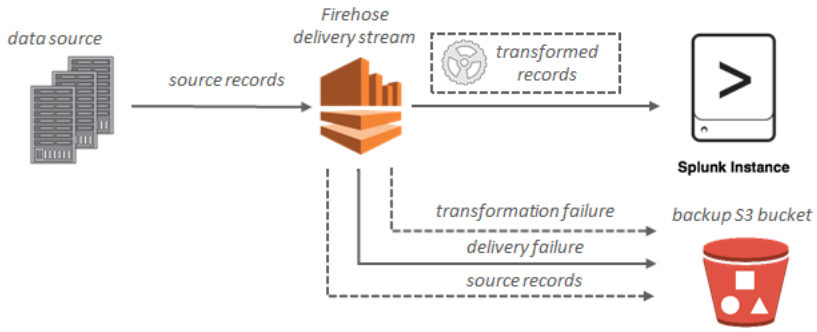
Para os destinos do Amazon Redshift, os dados em streaming são entregues primeiro no bucket do S3. Em seguida, o Amazon Data Firehose emite um comando do Amazon COPY Redshift para carregar dados do seu bucket do S3 para o seu cluster do Amazon Redshift. Se a transformação de dados estiver habilitada, você também poderá fazer backup dos dados da fonte em outro bucket do Amazon S3.



Para destinos OpenSearch de serviço, os dados de streaming são entregues ao seu cluster de OpenSearch serviços e, opcionalmente, podem ser copiados para seu bucket do S3 simultaneamente.



Para destinos do Splunk, os dados em streaming são entregues ao Splunk e eles podem ser submetidos a backup no bucket do S3 simultaneamente, se você desejar.


data source
Firehose
delivery stream
source records
transformed
records
Splunk Instance
transformation failure
backup S3 bucket
delivery failure
source records

Configuração para o Amazon Data Firehose

Antes de usar o Amazon Data Firehose pela primeira vez, conclua as tarefas a seguir.

Tarefas

- Inscreva-se para AWS
- (Opcional) Baixe bibliotecas e ferramentas

Inscreva-se para AWS

Quando você se inscreve no Amazon Web Services (AWS), sua AWS conta é automaticamente cadastrada em todos os serviços AWS, incluindo o Amazon Data Firehose. Você será cobrado apenas pelos serviços que usar.

Se você já tiver uma AWS conta, vá para a próxima tarefa. Se você ainda não possuir uma conta da AWS , use o procedimento a seguir para criar uma.

Para se inscrever em uma AWS conta

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

   Parte do procedimento de inscrição envolve receber uma chamada telefônica e inserir 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 atributos 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.

(Opcional) Baixe bibliotecas e ferramentas

As bibliotecas e ferramentas a seguir ajudarão você a trabalhar com o Amazon Data Firehose de forma programática e a partir da linha de comando:

- O Firehose API Operations é o conjunto básico de operações que o Amazon Data Firehose suporta.

- Os AWS SDKs para Go, Java, .NET, Node.js, Python e Ruby incluem suporte e amostras do Amazon Data Firehose.

  Se sua versão do AWS SDK for Java não incluir amostras para o Amazon Data Firehose, você também pode baixar o AWS SDK mais recente em. GitHub
- O AWS Command Line Interfacesuporta o Amazon Data Firehose. AWS CLI Isso permite que você controle vários AWS serviços a partir da linha de comando e os automatize por meio de scripts.

Crie um stream do Firehose

Você pode usar o AWS Management Console ou um AWS SDK para criar um stream do Firehose para o destino escolhido.

Você pode atualizar a configuração do seu stream do Firehose a qualquer momento após sua criação, usando o console Amazon Data Firehose ou. UpdateDestination Seu stream do Firehose permanece no `Active` estado enquanto sua configuração é atualizada, e você pode continuar enviando dados. A configuração atualizada normalmente entra em vigor em poucos minutos. O número da versão de um stream do Firehose é aumentado em um valor de 1 após a atualização da configuração. Ele é refletido no nome do objeto do Amazon S3 entregue. Para ter mais informações, consulte Configurar o formato de nome de objeto do Amazon S3.

Os tópicos a seguir descrevem como criar um stream do Firehose.

Tópicos

Configurar origem e destino

1. Faça login AWS Management Console e abra o console do Amazon Data Firehose em https://console.aws.amazon.com/firehose
2. Escolha Create Firehose stream.
3. Insira valores para os seguintes campos:

   Origem

   - Direct PUT: escolha essa opção para criar um stream do Firehose no qual os aplicativos produtores gravam diretamente. Atualmente, os seguintes são AWS serviços, agentes e serviços de código aberto integrados ao Direct PUT no Amazon Data Firehose:
     - AWS SDK

- AWS Lambda
- AWS CloudWatch Registros
- AWS CloudWatch Eventos
- AWS Fluxos métricos de nuvem
- AWS IOT
- AWS Eventbridge
- Amazon Simple Email Service
- Amazon SNS
- AWS Registros de ACL na web do WAF
- Amazon API Gateway: logs de acesso
- Amazon Pinpoint
- Logs do agente do Amazon MSK
- Logs de consultas do Amazon Route 53 Resolver
- AWS Registros de alertas do Firewall de Rede
- AWS Registros de fluxo do Firewall de Rede
- SLOWLOG do Amazon Elasticache Redis
- Kinesis Agent (linux)
- Kinesis Tap (windows)
- Fluentbit
- Fluentd
- Apache Nifi
- Snowflake

- Stream do Kinesis: escolha essa opção para configurar um stream do Firehose que usa um stream de dados do Kinesis como fonte de dados. Em seguida, você pode usar o Amazon Data Firehose para ler dados facilmente de um stream de dados existente do Kinesis e carregá-los nos destinos. Para obter mais informações sobre o uso do Kinesis Data Streams como fonte de dados, consulte Gravando no Amazon Data Firehose usando o Kinesis Data Streams.
- Amazon MSK: escolha essa opção para configurar um stream do Firehose que usa o Amazon MSK como fonte de dados. Em seguida, você pode usar o Firehose para ler dados facilmente de um cluster Amazon MSK existente e carregá-los em buckets S3

específicos. Para obter mais informações sobre o uso do Amazon MSK como fonte de dados, consulte Gravando no Amazon Data Firehose usando o Amazon MSK.

Destino do stream Firehose

O destino do seu stream do Firehose. O Amazon Data Firehose pode enviar registros de dados para vários destinos, incluindo Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon Service e qualquer endpoint HTTP que seja de sua propriedade ou de qualquer um de seus provedores de serviços terceirizados. OpenSearch Estes são os destinos compatíveis:

- OpenSearch Serviço Amazon
- Amazon sem OpenSearch servidor
- Amazon Redshift
- Amazon S3
- Coralogix
- Datadog
- Dynatrace
- Elastic
- Endpoint HTTP
- Honeycomb
- Logic Monitor
- Logz.io
- MongoDB Cloud
- New Relic
- Splunk
- Splunk Observability Cloud
- Sumo Logic
- Snowflake

Nome do stream Firehose

O nome do seu stream do Firehose.

Configurar a transformação de registros e a conversão de formatos

Configure o Amazon Data Firehose para transformar e converter seus dados de registro.

- Se você escolher o Amazon MSK como fonte para seu stream do Firehose.

  1. Na seção Transformar registros de origem com AWS Lambda, forneça valores para o seguinte campo:

     Transformação de dados

     Para criar um stream do Firehose que não transforme os dados recebidos, não marque a caixa de seleção Ativar transformação de dados.

     Para especificar uma função Lambda para o Firehose invocar e usar para transformar os dados recebidos antes de entregá-los, marque a caixa de seleção Ativar transformação de dados. Você pode configurar uma nova função do Lambda usando um dos esquemas do Lambda ou selecionar uma função do Lambda já existente. Sua função Lambda deve conter o modelo de status exigido pelo Firehose. Para ter mais informações, consulte Transformação de dados do Amazon Data Firehose.
  2. Na seção Convert record format (Converter formato do registro), forneça valores para o seguinte campo:

     Record format conversion (Conversão do formato do registro)

     Para criar um stream do Firehose que não converta o formato dos registros de dados recebidos, escolha Disabled.

     Para converter o formato dos registros de entrada, selecione Enabled (Habilitado) e especifique o formato de saída que deseja. Você precisa especificar uma AWS Glue tabela que contenha o esquema que você deseja que o Firehose use para converter seu formato de registro. Para ter mais informações, consulte Conversão do formato do registro.

     Para obter um exemplo de como configurar a conversão do formato de registro com AWS CloudFormation, consulte AWS::KinesisFirehose:: DeliveryStream.

- Se você escolher Managed Service para Apache Flink ou Direct PUT como fonte para seu stream do Firehose, na seção Configurações de origem:

  1. Em Transformar registros, escolha uma das seguintes opções:

a. Se seu destino for Amazon S3 ou Splunk, na seção Descompactar registros de origem CloudWatch Amazon Logs, escolha Ativar descompressão.

b. Na seção Transformar registros de origem com AWS Lambda, forneça valores para o seguinte campo:

   Transformação de dados

   Para criar um stream do Firehose que não transforme os dados recebidos, não marque a caixa de seleção Ativar transformação de dados.

   Para especificar uma função Lambda para o Amazon Data Firehose invocar e usar para transformar dados recebidos antes de entregá-los, marque a caixa de seleção Habilitar transformação de dados. Você pode configurar uma nova função do Lambda usando um dos esquemas do Lambda ou selecionar uma função do Lambda já existente. Sua função Lambda deve conter o modelo de status exigido pelo Amazon Data Firehose. Para ter mais informações, consulte Transformação de dados do Amazon Data Firehose.

2. Na seção Convert record format (Converter formato do registro), forneça valores para o seguinte campo:

   Record format conversion (Conversão do formato do registro)

   Para criar um stream do Firehose que não converta o formato dos registros de dados recebidos, escolha Disabled.

   Para converter o formato dos registros de entrada, selecione Enabled (Habilitado) e especifique o formato de saída que deseja. Você precisa especificar uma AWS Glue tabela que contenha o esquema que você deseja que o Amazon Data Firehose use para converter seu formato de registro. Para ter mais informações, consulte Conversão do formato do registro.

   Para obter um exemplo de como configurar a conversão do formato de registro com AWS CloudFormation, consulte AWS::KinesisFirehose:: DeliveryStream.

Definir configurações de destino

Este tópico descreve as configurações de destino do seu stream do Firehose com base no destino selecionado. Para obter mais informações sobre dicas de buffer, consulte. Entenda as dicas de buffer

Tópicos

Defina as configurações de destino para o Amazon S3

Você deve especificar as seguintes configurações para usar o Amazon S3 como destino para seu stream do Firehose.

- Insira valores para os seguintes campos.

  S3 bucket

  Escolha um bucket do S3 do qual você seja proprietário; os dados em streaming serão entregues nesse bucket. É possível criar um novo bucket do S3 ou escolher um já existente.

Novo delimitador de linha

Você pode configurar seu stream do Firehose para adicionar um novo delimitador de linha entre registros em objetos que são entregues ao Amazon S3. Para fazer isso, escolha Habilitado. Para não adicionar um novo delimitador de linha entre registros nos objetos que são entregues ao Amazon S3, escolha Desabilitado. Se você planeja usar o Athena para consultar objetos do S3 com registros agregados, habilite essa opção.

Particionamento dinâmico

Escolha Habilitado para habilitar e configurar o particionamento dinâmico.

Desagregação de vários registros

Esse é o processo de analisar os registros no stream do Firehose e separá-los com base no JSON válido ou no novo delimitador de linha especificado.

Se você agregar vários eventos, registros ou registros em uma única PutRecord chamada de PutRecordBatch API, ainda poderá ativar e configurar o particionamento dinâmico. Com dados agregados, quando você ativa o particionamento dinâmico, o Amazon Data Firehose analisa os registros e procura vários objetos JSON válidos em cada chamada de API. Quando o stream do Firehose é configurado com o Kinesis Data Stream como fonte, você também pode usar a agregação integrada na Kinesis Producer Library (KPL). A funcionalidade de partição de dados é executada após a desagregação dos dados. Portanto, cada registro em cada chamada de API pode ser entregue a diferentes prefixos do Amazon S3. Você também pode aproveitar a integração da função Lambda para realizar qualquer outra desagregação ou qualquer outra transformação antes da funcionalidade de particionamento de dados.

> **Important**
>
> Se os dados estiverem agregados, o particionamento dinâmico só poderá ser aplicado após a desagregação de dados ser realizada. Portanto, se você habilitar o particionamento dinâmico para seus dados agregados, deverá escolher Habilitado para habilitar a desagregação de vários registros.

O Firehose stream executa as seguintes etapas de processamento na seguinte ordem: desagregação KPL (protobuf), desagregação JSON ou delimitadora, processamento

Lambda, particionamento de dados, conversão de formato de dados e entrega do Amazon S3.

Tipo de desagregação de vários registros

Se você ativou a desagregação de vários registros, deverá especificar o método para o Firehose desagregar seus dados. Use o menu suspenso para escolher JSON ou Delimitado.

Análise em linha

Esse é um dos mecanismos compatíveis com o particionamento dinâmico dos dados vinculados ao Amazon S3. Para usar a análise em linha para fazer o particionamento dinâmico de dados, você deve especificar os parâmetros de registro de dados a serem usados como chaves de particionamento e fornecer um valor para cada chave de particionamento especificada. Escolha Habilitado para habilitar e configurar o particionamento em linha.

> **Important**
>
> Se você especificou uma função AWS Lambda nas etapas acima para transformar seus registros de origem, poderá usar essa função para particionar dinamicamente seus dados vinculados ao S3 e ainda poderá criar suas chaves de particionamento com análise embutida. Com o particionamento dinâmico, você pode usar a análise embutida ou a função AWS Lambda para criar suas chaves de particionamento. Ou você pode usar a análise embutida e a função AWS Lambda ao mesmo tempo para criar suas chaves de particionamento.

Chaves de particionamento dinâmico

Você pode usar os campos Chave e Valor para especificar os parâmetros de registro de dados a serem usados como chaves de particionamento dinâmico e consultas jq para gerar os valores das chaves de particionamento dinâmico. O Firehose suporta somente o jq 1.6. É possível especificar até 50 chaves de particionamento dinâmico. Você deve inserir expressões jq válidas para seus valores de chave de particionamento dinâmico para configurar com êxito o particionamento dinâmico para seu stream do Firehose.

Prefixo de bucket do S3

Ao habilitar e configurar o particionamento dinâmico, você deve especificar os prefixos de bucket do S3 para os quais o Amazon Data Firehose deve entregar dados particionados.

Para que o particionamento dinâmico seja configurado corretamente, o número dos prefixos de bucket do S3 deve ser idêntico ao número de chaves de particionamento especificadas.

Você pode particionar seus dados de origem com análise embutida ou com a função Lambda especificada AWS . Se você especificou uma função AWS Lambda para criar chaves de particionamento para seus dados de origem, deverá digitar manualmente o (s) valor (es) do prefixo do bucket do S3 usando o seguinte formato: "lambda:keyID". partitionKeyFrom Se você estiver usando análise embutida para especificar as chaves de particionamento para seus dados de origem, você pode digitar manualmente os valores de visualização do bucket do S3 usando o seguinte formato: "partitionKeyFromquery:keyID" ou escolher o botão Aplicar chaves de particionamento dinâmico para usar seus pares de chave/valor de particionamento dinâmico para gerar automaticamente seus prefixos de bucket do S3. Ao particionar seus dados com análise embutida ou AWS Lambda, você também pode usar os seguintes formulários de expressão no prefixo do bucket do S3:! {namespace:value}, em que o namespace pode ser Query ou Lambda. partitionKeyFrom partitionKeyFrom

Fuso horário do bucket S3 e do prefixo de saída de erro do S3

Escolha um fuso horário que você deseja usar para data e hora em Prefixos personalizados para objetos do Amazon Simple Storage Service. Por padrão, o Firehose adiciona um prefixo de hora em UTC. Você pode alterar o fuso horário usado nos prefixos do S3 se quiser usar um fuso horário diferente.

Dicas de armazenamento em buffer

O Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Compressão S3

Escolha a compactação de dados GZIP, Snappy, Zip ou Snappy compatível com Hadoop ou nenhuma compactação de dados. A compactação Snappy compatível com Snappy, Zip e Hadoop não está disponível para streams do Firehose com o Amazon Redshift como destino.

Formato de extensão de arquivo S3 (opcional)

Especifique um formato de extensão de arquivo para objetos entregues ao bucket de destino do Amazon S3. Se você habilitar esse recurso, a extensão de arquivo especificada substituirá as extensões de arquivo padrão anexadas pelos recursos de conversão de formato de dados ou compactação do S3, como .parquet ou .gz. Verifique se você configurou

a extensão de arquivo correta ao usar esse recurso com a conversão de formato de dados ou a compactação S3. A extensão do arquivo deve começar com um ponto (.) e pode conter caracteres permitidos: 0-9a-z! -_.*' (). A extensão do arquivo não pode exceder 128 caracteres.

Criptografia S3

O Firehose oferece suporte à criptografia do lado do servidor Amazon S3 AWS Key Management Service com (SSE-KMS) para criptografar dados entregues no Amazon S3. Você pode optar por usar o tipo de criptografia padrão especificado no bucket S3 de destino ou criptografar com uma chave da lista de AWS KMS chaves que você possui. Se você criptografar os dados com AWS KMS chaves, poderá usar a chave AWS gerenciada padrão (aws/s3) ou uma chave gerenciada pelo cliente. Para obter mais informações, consulte Proteção de dados usando criptografia do lado do servidor com chaves AWS gerenciadas pelo KMS (SSE-KMS).

Defina as configurações de destino para o Amazon Redshift

Esta seção descreve as configurações para usar o Amazon Redshift como seu destino de stream do Firehose.

Escolha um dos procedimentos a seguir dependendo de você ter um cluster provisionado pelo Amazon Redshift ou um grupo de trabalho do Amazon Redshift Sem Servidor.

- Cluster provisionado do Amazon Redshift
- Defina as configurações de destino para o grupo de trabalho Amazon Redshift Serverless

Cluster provisionado do Amazon Redshift

Esta seção descreve as configurações para usar o cluster provisionado do Amazon Redshift como seu destino de stream do Firehose.

- Insira valores para os seguintes campos:

  Cluster

  O cluster do Amazon Redshift no qual os dados do bucket do S3 são copiados. Configure o cluster do Amazon Redshift para ser acessível publicamente e desbloqueie os endereços IP

do Amazon Data Firehose. Para ter mais informações, consulte Conceda ao Amazon Data Firehose acesso a um destino do Amazon Redshift .

Autenticação

Você pode optar por inserir o nome de usuário/senha diretamente ou recuperar o segredo AWS Secrets Manager para acessar o cluster do Amazon Redshift.

- Nome do usuário

  Especifique um usuário do Amazon Redshift com permissões para acessar o cluster do Amazon Redshift. Esse usuário deve ter a permissão `INSERT` do Amazon Redshift para copiar dados do bucket do S3 no cluster do Amazon Redshift.

- Senha

  Especifique a senha do usuário que tem permissões para acessar o cluster.

- Secret

  Selecione um segredo AWS Secrets Manager que contenha as credenciais para o cluster do Amazon Redshift. Se você não vê seu segredo na lista suspensa, crie um AWS Secrets Manager para suas credenciais do Amazon Redshift. Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

Database

O banco de dados do Amazon Redshift no qual os dados são copiados.

Tabela

A tabela do Amazon Redshift no qual os dados são copiados.

Columns

(Opcional) As colunas específicas da tabela na qual os dados serão copiados. Use essa opção se o número de colunas definidas nos objetos do Amazon S3 for menor que o número de colunas na tabela do Amazon Redshift.

Destino intermediário do S3

Primeiro, o Firehose entrega seus dados para o bucket do S3 e, em seguida, emite um COPY comando do Amazon Redshift para carregar os dados no seu cluster do Amazon Redshift. Especifique um bucket do S3 do qual você seja proprietário; os dados em streaming serão entregues nesse bucket. Crie um novo bucket do S3 ou escolha um bucket já existente do qual você seja proprietário.

O Firehose não exclui os dados do seu bucket do S3 depois de carregá-los no cluster do Amazon Redshift. Você pode gerenciar os dados no bucket do S3 usando uma configuração de ciclo de vida. Para obter mais informações, consulte Gerenciamento do ciclo de vida de objetos no Guia do usuário do Amazon Simple Storage Service.

Prefixo de bucket intermediário do S3

(Opcional) Para usar o prefixo padrão para objetos do Amazon S3, deixe esta opção em branco. O Firehose usa automaticamente um prefixo no formato de hora UTC "`YYYY/MM/dd/HH`" para objetos Amazon S3 entregues. Você pode adicionar ao início deste prefixo. Para ter mais informações, consulte Configurar o formato de nome de objeto do Amazon S3.

Opções do COPY

Parâmetros que você pode especificar no comando COPY do Amazon Redshift. Eles podem ser necessários para a configuração. Por exemplo, "`GZIP`" é necessário se a compactação de dados do Amazon S3 estiver ativada. “`REGION`" é necessário se seu bucket do S3 não estiver na mesma AWS região do seu cluster do Amazon Redshift. Para obter mais informações, consulte COPY no Guia do desenvolvedor de banco de dados do Amazon Redshift.

COPY command

O comando COPY do Amazon Redshift. Para obter mais informações, consulte COPY no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Período de novas tentativas

Tempo de duração (0—7200 segundos) para o Firehose tentar novamente se os dados do seu cluster do COPY Amazon Redshift falharem. O Firehose tenta novamente a cada 5 minutos até que a duração da nova tentativa termine. Se você definir a duração da nova tentativa para 0 (zero) segundos, o Firehose não tentará novamente em caso COPY de falha de comando.

Dicas de armazenamento em buffer

O Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Compressão S3

Escolha a compactação de dados GZIP, Snappy, Zip ou Snappy compatível com Hadoop ou nenhuma compactação de dados. A compactação Snappy compatível com Snappy, Zip e Hadoop não está disponível para streams do Firehose com o Amazon Redshift como destino.

Formato de extensão de arquivo S3 (opcional)

Formato de extensão de arquivo S3 (opcional) — Especifique um formato de extensão de arquivo para objetos entregues ao bucket de destino do Amazon S3. Se você habilitar esse recurso, a extensão de arquivo especificada substituirá as extensões de arquivo padrão anexadas pelos recursos de conversão de formato de dados ou compactação do S3, como .parquet ou .gz. Verifique se você configurou a extensão de arquivo correta ao usar esse recurso com a conversão de formato de dados ou a compactação S3. A extensão do arquivo deve começar com um ponto (.) e pode conter caracteres permitidos: 0-9a-z! -_.*' (). A extensão do arquivo não pode exceder 128 caracteres.

Criptografia S3

O Firehose oferece suporte à criptografia do lado do servidor Amazon S3 AWS Key Management Service com (SSE-KMS) para criptografar dados entregues no Amazon S3. Você pode optar por usar o tipo de criptografia padrão especificado no bucket S3 de destino ou criptografar com uma chave da lista de AWS KMS chaves que você possui. Se você criptografar os dados com AWS KMS chaves, poderá usar a chave AWS gerenciada padrão (aws/s3) ou uma chave gerenciada pelo cliente. Para obter mais informações, consulte Proteção de dados usando criptografia do lado do servidor com chaves AWS gerenciadas pelo KMS (SSE-KMS).

Defina as configurações de destino para o grupo de trabalho Amazon Redshift Serverless

Esta seção descreve as configurações para usar o grupo de trabalho Amazon Redshift Serverless como seu destino de stream do Firehose.

- Insira valores para os seguintes campos:

Workgroup name (Nome do grupo de trabalho)

O grupo de trabalho do Amazon Redshift Sem Servidor ou um grupo de trabalho do Amazon Redshift no qual os dados do bucket do S3 são copiados. Configure o grupo de trabalho Amazon Redshift Serverless para ser acessível ao público e desbloqueie os endereços IP do Firehose. Para obter mais informações, consulte a seção Conectar-se a uma instância do Amazon Redshift Sem Servidor acessível publicamente em Conectar-se ao Amazon Redshift Sem Servidor e também Conceda ao Amazon Data Firehose acesso a um destino do Amazon Redshift .

Autenticação

Você pode optar por inserir o nome de usuário/senha diretamente ou recuperar o segredo AWS Secrets Manager para acessar o grupo de trabalho Amazon Redshift Serverless.

- Nome do usuário

  Especifique um usuário do Amazon Redshift com permissões para acessar o grupo de trabalho do Amazon Redshift Serverless. Esse usuário deve ter a permissão `INSERT` do Amazon Redshift para copiar dados do bucket do S3 para o grupo de trabalho do Amazon Redshift Sem Servidor.

- Senha

  Especifique a senha do usuário que tem permissões para acessar o grupo de trabalho Amazon Redshift Serverless.

- Secret

  Selecione um segredo AWS Secrets Manager que contenha as credenciais do grupo de trabalho Amazon Redshift Serverless. Se você não vê seu segredo na lista suspensa, crie um AWS Secrets Manager para suas credenciais do Amazon Redshift. Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

Database

O banco de dados do Amazon Redshift no qual os dados são copiados.

Tabela

A tabela do Amazon Redshift no qual os dados são copiados.

Columns

(Opcional) As colunas específicas da tabela na qual os dados serão copiados. Use essa opção se o número de colunas definidas nos objetos do Amazon S3 for menor que o número de colunas na tabela do Amazon Redshift.

Destino intermediário do S3

O Amazon Data Firehose entrega primeiro seus dados para o bucket do S3 e, em seguida, emite um COPY comando do Amazon Redshift para carregar os dados no seu grupo de trabalho sem servidor do Amazon Redshift. Especifique um bucket do S3 do qual você seja proprietário; os dados em streaming serão entregues nesse bucket. Crie um novo bucket do S3 ou escolha um bucket já existente do qual você seja proprietário.

O Firehose não exclui os dados do seu bucket do S3 depois de carregá-los no seu grupo de trabalho Amazon Redshift Serverless. Você pode gerenciar os dados no bucket do S3 usando uma configuração de ciclo de vida. Para obter mais informações, consulte Gerenciamento do ciclo de vida de objetos no Guia do usuário do Amazon Simple Storage Service.

Prefixo de bucket intermediário do S3

(Opcional) Para usar o prefixo padrão para objetos do Amazon S3, deixe esta opção em branco. O Firehose usa automaticamente um prefixo no formato de hora UTC "`YYYY/MM/dd/HH`" para objetos Amazon S3 entregues. Você pode adicionar ao início deste prefixo. Para ter mais informações, consulte Configurar o formato de nome de objeto do Amazon S3.

Opções do COPY

Parâmetros que você pode especificar no comando COPY do Amazon Redshift. Eles podem ser necessários para a configuração. Por exemplo, "`GZIP`" é necessário se a compactação de dados do Amazon S3 estiver ativada. “`REGION`" é obrigatório se seu bucket do S3 não estiver na mesma AWS região do seu grupo de trabalho Amazon Redshift Serverless. Para obter mais informações, consulte COPY no Guia do desenvolvedor de banco de dados do Amazon Redshift.

COPY command

O comando COPY do Amazon Redshift. Para obter mais informações, consulte COPY no Guia do desenvolvedor de banco de dados do Amazon Redshift.

Período de novas tentativas

Tempo de duração (0—7200 segundos) para o Firehose tentar novamente se os dados do seu grupo de trabalho COPY Amazon Redshift Serverless falharem. O Firehose tenta novamente a cada 5 minutos até que a duração da nova tentativa termine. Se você definir a duração da nova tentativa para 0 (zero) segundos, o Firehose não tentará novamente em caso COPY de falha de comando.

Dicas de armazenamento em buffer

O Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Compressão S3

Escolha a compactação de dados GZIP, Snappy, Zip ou Snappy compatível com Hadoop ou nenhuma compactação de dados. A compactação Snappy compatível com Snappy, Zip e Hadoop não está disponível para streams do Firehose com o Amazon Redshift como destino.

Formato de extensão de arquivo S3 (opcional)

Formato de extensão de arquivo S3 (opcional) — Especifique um formato de extensão de arquivo para objetos entregues ao bucket de destino do Amazon S3. Se você habilitar esse recurso, a extensão de arquivo especificada substituirá as extensões de arquivo padrão anexadas pelos recursos de conversão de formato de dados ou compactação do S3, como .parquet ou .gz. Verifique se você configurou a extensão de arquivo correta ao usar esse recurso com a conversão de formato de dados ou a compactação S3. A extensão do arquivo deve começar com um ponto (.) e pode conter caracteres permitidos: 0-9a-z! -_.*' (). A extensão do arquivo não pode exceder 128 caracteres.

Criptografia S3

O Firehose oferece suporte à criptografia do lado do servidor Amazon S3 AWS Key Management Service com (SSE-KMS) para criptografar dados entregues no Amazon S3. Você pode optar por usar o tipo de criptografia padrão especificado no bucket S3 de destino ou criptografar com uma chave da lista de AWS KMS chaves que você possui. Se você criptografar os dados com AWS KMS chaves, poderá usar a chave AWS gerenciada padrão (aws/s3) ou uma chave gerenciada pelo cliente. Para obter mais informações, consulte Proteção de dados usando criptografia do lado do servidor com chaves AWS gerenciadas pelo KMS (SSE-KMS).

Definir configurações de destino para o OpenSearch Serviço

Esta seção descreve as opções para usar o OpenSearch Serviço em seu destino.

- Insira valores para os seguintes campos:

  OpenSearch Domínio do serviço

  O domínio do OpenSearch serviço para o qual seus dados são entregues.

  Índice

  O nome do índice de OpenSearch serviço a ser usado ao indexar dados em seu cluster OpenSearch de serviços.

  Index rotation

  Escolha se e com que frequência o índice OpenSearch de serviços deve ser rotacionado. Se a rotação do índice estiver ativada, o Amazon Data Firehose anexará o timestamp correspondente ao nome do índice especificado e rotacionará. Para ter mais informações, consulte Configurar a rotação do índice para o OpenSearch Serviço.

  Tipo

  O nome do tipo de OpenSearch serviço a ser usado ao indexar dados em seu cluster OpenSearch de serviços. Para o Elasticsearch 7.x e OpenSearch 1.x, só pode haver um tipo por índice. Se você tentar especificar um novo tipo para um índice existente que já tem outro tipo, o Firehose retornará um erro durante o tempo de execução.

  Para o Elasticsearch 7.x, deixe esse campo vazio.

  Período de novas tentativas

  Duração do tempo para que o Firehose tente novamente se uma solicitação de indexação falhar. OpenSearch Nesse caso, o Firehose tenta novamente a cada 5 minutos até que a duração da nova tentativa expire. Para a duração da nova tentativa, você pode definir qualquer valor entre 0 e 7200 segundos.

  Depois que a duração da nova tentativa expirar, o Firehose entrega os dados para o Dead Letter Queue (DLQ), um bucket de erros S3 configurado. Para dados entregues ao DLQ, você precisa redirecionar os dados do bucket de erro S3 configurado para o destino. OpenSearch

Se você quiser impedir que o stream do Firehose entregue dados ao DLQ devido ao tempo de inatividade ou à manutenção dos OpenSearch clusters, você pode configurar a duração da nova tentativa para um valor maior em segundos. Você pode aumentar o valor da duração da nova tentativa acima para 7200 segundos entrando em contato com o AWS suporte.

Tipo DocumentID

Indica o método para configurar o ID do documento. Os métodos compatíveis são ID do documento gerado pelo Firehose e ID do documento gerado pelo OpenSearch serviço. A ID do documento gerada pelo Firehose é a opção padrão quando o valor da ID do documento não está definido. OpenSearch O ID do documento gerado pelo serviço é a opção recomendada porque suporta operações de gravação pesada, incluindo análise de registros e observabilidade, consumindo menos recursos de CPU no domínio do OpenSearch Serviço e, portanto, resultando em melhor desempenho.

Destination VPC connectivity (Conectividade da VPC de destino)

Se seu domínio OpenSearch de serviço estiver em uma VPC privada, use esta seção para especificar essa VPC. Especifique também as sub-redes e os subgrupos que você deseja que o Amazon Data Firehose use ao enviar dados para o seu domínio de serviço. OpenSearch Você pode usar os mesmos grupos de segurança que o domínio do OpenSearch Serviço está usando. Se você especificar grupos de segurança diferentes, certifique-se de que eles permitam tráfego HTTPS de saída para o grupo de segurança do domínio do OpenSearch Serviço. Além disso, certifique-se de que o grupo de segurança do domínio OpenSearch Service permita tráfego HTTPS dos grupos de segurança que você especificou ao configurar seu stream do Firehose. Se você usar o mesmo grupo de segurança para o stream do Firehose e para o domínio OpenSearch Service, verifique se a regra de entrada do grupo de segurança permite tráfego HTTPS. Para obter mais informações sobre regras de grupo de segurança, consulte Regras de grupo de segurança na documentação da Amazon VPC.

> **Important**
>
> Ao especificar sub-redes para entregar dados ao destino em uma VPC privada, verifique se você tem um número suficiente de endereços IP livres nas sub-redes escolhidas. Se não houver um endereço IP gratuito disponível em uma sub-rede especificada, o Firehose não poderá criar ou adicionar ENIs para a entrega de dados na VPC privada, e a entrega será degradada ou falhará.

Sugestões de buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Definir configurações de destino para OpenSearch Serverless

Esta seção descreve as opções para usar o OpenSearch Serverless como seu destino.

- Insira valores para os seguintes campos:

  OpenSearch Coleção sem servidor

  O endpoint de um grupo de índices OpenSearch sem servidor para os quais seus dados são entregues.

  Índice

  O nome do índice OpenSearch Serverless a ser usado ao indexar dados para sua OpenSearch coleção Serverless.

  Destination VPC connectivity (Conectividade da VPC de destino)

  Se sua coleção OpenSearch Serverless estiver em uma VPC privada, use esta seção para especificar essa VPC. Especifique também as sub-redes e os subgrupos que você deseja que o Amazon Data Firehose use ao enviar dados para sua coleção Serverless. OpenSearch

  > **Important**
  >
  > Ao especificar sub-redes para entregar dados ao destino em uma VPC privada, verifique se você tem um número suficiente de endereços IP livres nas sub-redes escolhidas. Se não houver um endereço IP gratuito disponível em uma sub-rede especificada, o Firehose não poderá criar ou adicionar ENIs para a entrega de dados na VPC privada, e a entrega será degradada ou falhará.

Período de novas tentativas

Duração do tempo para o Firehose tentar novamente se uma solicitação de indexação para Serverless falhar OpenSearch . Nesse caso, o Firehose tenta novamente a cada 5 minutos até que a duração da nova tentativa expire. Para a duração da nova tentativa, você pode definir qualquer valor entre 0 e 7200 segundos.

Depois que a duração da nova tentativa expirar, o Firehose entrega os dados para o Dead Letter Queue (DLQ), um bucket de erros S3 configurado. Para dados entregues ao DLQ, você precisa redirecionar os dados do bucket de erro S3 configurado para OpenSearch o destino sem servidor.

Se você quiser impedir que o stream do Firehose entregue dados ao DLQ devido ao tempo de inatividade ou à manutenção de clusters OpenSearch sem servidor, você pode configurar a duração da nova tentativa para um valor maior em segundos. Você pode aumentar o valor da duração da nova tentativa acima para 7200 segundos entrando em contato com o AWS suporte.

Sugestões de buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Definir as configurações de destino para o Endpoint HTTP

Esta seção descreve as opções para usar o endpoint HTTP como destino.

> **Important**
>
> Se você escolher um endpoint HTTP como destino, revise e siga as instruções em Apêndice - Especificações de solicitação e resposta de entrega de endpoint HTTP.

- Forneça os valores para os seguintes campos:

Nome do endpoint HTTP - opcional

Especifique um nome de usuário amigável para o endpoint HTTP. Por exemplo, `My HTTP Endpoint Destination`.

URL do endpoint HTTP

Especifique a URL para o endpoint HTTP no seguinte formato: `https://xyz.httpendpoint.com`. A origem deve ser uma URL HTTPS.

Autenticação

Você pode optar por inserir a chave de acesso diretamente ou recuperar o segredo AWS Secrets Manager para acessar o endpoint HTTP.

- (Opcional) Chave de acesso

  Entre em contato com o proprietário do endpoint se precisar obter a chave de acesso para permitir a entrega de dados do Firehose para o endpoint.

- Secret

  Selecione um segredo AWS Secrets Manager que contenha a chave de acesso para o endpoint HTTP. Se você não encontrar seu segredo na lista suspensa, crie um AWS Secrets Manager para a chave de acesso. Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

Codificação de conteúdo

O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo de uma solicitação antes de enviá-la ao destino. Escolha GZIP ou Desabilitado para habilitar/desabilitar a codificação de conteúdo da solicitação.

Período de novas tentativas

Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o endpoint HTTP selecionado.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar.

Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

> **Important**
>
> Para os destinos do endpoint HTTP, se você estiver vendo 413 códigos de resposta do endpoint de destino no CloudWatch Logs, diminua o tamanho da dica de buffer no stream do Firehose e tente novamente.

Definir configurações de destino para o Datadog

Esta seção descreve as opções para usar o Datadog como destino. Para obter mais informações sobre o Datadog, consulte https://docs.datadoghq.com/integrations/amazon_web_services/.

- Forneça valores para os campos a seguir.

  URL do endpoint HTTP

  Escolha para onde você deseja enviar dados de uma das seguintes opções no menu suspenso.

  - Logs do Datadog - EUA1
  - Registros do Datadog - US3
  - Logs do Datadog - EUA5
  - Registros do Datadog - AP1
  - Logs do Datadog: EU
  - Logs do Datadog: GOV
  - Métricas do Datadog - EUA
  - Métricas do Datadog - US5
  - Métricas do Datadog - AP1
  - Métricas do Datadog: EU
  - Configurações do Datadog - US1
  - Configurações do Datadog - US3
  - Configurações do Datadog - US5
  - Configurações do Datadog - AP1
  - Configurações do Datadog - UE
  - Configurações do Datadog - US GOV

  Autenticação

  Você pode optar por inserir a chave da API diretamente ou recuperar o segredo AWS Secrets Manager para acessar o Datadog.

  - Chave de API

    Entre em contato com o Datadog para obter a chave de API necessária para permitir a entrega de dados do Firehose para esse endpoint.

  - Secret

    Selecione um segredo AWS Secrets Manager que contenha a chave de API do Datadog. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager

Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

Codificação de conteúdo

O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo de uma solicitação antes de enviá-la ao destino. Escolha GZIP ou Desabilitado para habilitar/desabilitar a codificação de conteúdo da solicitação.

Período de novas tentativas

Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o endpoint HTTP selecionado.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Defina as configurações de destino para o Honeycomb

Esta seção descreve as opções para usar o Honeycomb como destino. Para obter mais informações sobre o Honeycomb, consulte https://docs.honeycomb.io/ getting-data-in /metrics//aws-cloudwatch-metrics.

- Forneça os valores para os seguintes campos:

  Endpoint do Honeycomb Kinesis

  Especifique o URL do endpoint HTTP no seguinte formato: hhttps://api.honeycomb.io/1/kinesis_events/{{dataset}}

  Autenticação

  Você pode optar por inserir a chave da API diretamente ou recuperar o segredo AWS Secrets Manager para acessar o Honeycomb.

  - Chave de API

    Entre em contato com a Honeycomb para obter a chave de API necessária para permitir a entrega de dados a esse endpoint a partir do Firehose.

  - Secret

    Selecione um segredo AWS Secrets Manager que contenha a chave de API do Honeycomb. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

  Codificação de conteúdo

  O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo de uma solicitação antes de enviá-la ao destino. Escolha GZIP para habilitar a codificação de conteúdo da solicitação. Essa é a opção recomendada quando o destino é o Honeycomb.

Período de novas tentativas

Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o endpoint HTTP selecionado.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Defina as configurações de destino para o Coralogix

Esta seção descreve as opções para usar o Coralogix como destino. Para obter mais informações sobre o Coralogix, consulte https://coralogix.com/integrations/aws-firehose.

- Forneça os valores para os seguintes campos:

  URL do endpoint HTTP

  Escolha o URL do endpoint HTTP entre as seguintes opções no menu suspenso:

  - Coralogix - EUA
  - Coralogix - SINGAPURA
  - Coralogix - IRLANDA
  - Coralogix - ÍNDIA
  - Coralogix - ESTOCOLMO

  Autenticação

  Você pode optar por inserir a chave privada diretamente ou recuperar o segredo AWS Secrets Manager para acessar o Coralogix.

  - Chave privada

    Entre em contato com a Coralogix para obter a chave privada necessária para permitir a entrega de dados a esse endpoint a partir do Firehose.

  - Secret

    Selecione um segredo AWS Secrets Manager que contenha a chave privada do Coralogix. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

  Codificação de conteúdo

  O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo de uma solicitação antes de enviá-la ao destino. Escolha GZIP para habilitar a codificação de conteúdo da solicitação. Essa é a opção recomendada quando o destino é o Coralogix.

Período de novas tentativas

Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o endpoint HTTP selecionado.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

- applicationName: o ambiente em que você está executando o Data Firehose
- subsystemName: o nome da integração do Data Firehose
- ComputerName: o nome do stream Firehose em uso

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho do buffer recomendado para o destino varia de acordo com o provedor de serviços.

Definir as configurações de destino para o Dynatrace

Esta seção descreve as opções para usar o Dynatrace como destino. Para obter mais informações, consulte https://www.dynatrace.com/support/help/technology-support/cloud-platforms/ amazon-web-services cloudwatch-metric-streams /integrations//.

- Escolha as opções para usar o Dynatrace como destino para seu stream do Firehose.

  Tipo de ingestão

  Escolha se você deseja fornecer métricas ou registros (padrão) no Dynatrace para análise e processamento adicionais.

  URL do endpoint HTTP

  Escolha a URL do endpoint HTTP (Dynatrace US, Dynatrace EU ou Dynatrace Global) no menu suspenso.

  Autenticação

  Você pode optar por inserir o token da API diretamente ou recuperar o segredo AWS Secrets Manager para acessar o Dynatrace.

  - Token de API

    Gere o token da API Dynatrace que você precisa para habilitar a entrega de dados para esse endpoint a partir do Firehose. Para obter mais informações, consulte API Dynatrace - Tokens e autenticação.

  - Secret

    Selecione um segredo AWS Secrets Manager que contenha o token da API para o Dynatrace. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

  URL da API

  Forneça o URL da API do ambiente do Dynatrace.

  Codificação de conteúdo

  Escolha se você deseja ativar a codificação de conteúdo para compactar o corpo da solicitação. O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo

de uma solicitação antes de enviá-la ao destino. Quando ativado, o conteúdo é compactado no formato GZIP.

Período de novas tentativas

Especifique por quanto tempo o Firehose tenta enviar dados novamente para o endpoint HTTP selecionado.

Depois de enviar os dados, o Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Firehose iniciará o contador de duração da nova tentativa. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Firehose envia dados para o endpoint HTTP, durante a tentativa inicial ou depois de tentar novamente, ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. As dicas de buffer incluem o tamanho e o intervalo do buffer para seus streams. O tamanho do buffer recomendado para o destino varia de acordo com o provedor de serviços.

Defina as configurações de destino para LogicMonitor

Esta seção descreve as opções LogicMonitorpara uso em seu destino. Para obter mais informações, consulte https://www.logicmonitor.com.

- Forneça os valores para os seguintes campos:

  URL do endpoint HTTP

  Especifique o URL do endpoint HTTP no formato a seguir.

  ```
  https://ACCOUNT.logicmonitor.com
  ```

  Autenticação

  Você pode optar por inserir a chave da API diretamente ou recuperar o segredo AWS Secrets Manager para acessar LogicMonitor.

  - Chave de API

    Entre em contato LogicMonitor para obter a chave de API necessária para habilitar a entrega de dados do Firehose para esse endpoint.

  - Secret

    Selecione um segredo AWS Secrets Manager que contenha a chave de API para LogicMonitor. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

  Codificação de conteúdo

  O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo de uma solicitação antes de enviá-la ao destino. Escolha GZIP ou Desabilitado para habilitar/desabilitar a codificação de conteúdo da solicitação.

  Período de novas tentativas

  Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o endpoint HTTP selecionado.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Defina as configurações de destino para o Logz.io

Esta seção descreve as opções para usar o Logz.io como destino. Para obter mais informações, consulte https://logz.io/.

> **Note**
>
> Na região da Europa (Milão), o Logz.io não é suportado como destino do Amazon Data Firehose.

- Forneça os valores para os seguintes campos:

  URL do endpoint HTTP

  Especifique o URL do endpoint HTTP no formato a seguir. O URL deve ser um HTTPS URL.

  ```
  https://listener-aws-metrics-stream-<region>.logz.io/
  ```

  Por exemplo

  ```
  https://listener-aws-metrics-stream-us.logz.io/
  ```

  Autenticação

  Você pode optar por inserir o token de envio diretamente ou recuperar o segredo AWS Secrets Manager para acessar o Logz.io.

  - Token de envio

    Entre em contato com a Logz.io para obter o token de envio necessário para habilitar a entrega de dados do Firehose para esse endpoint.

  - Secret

    Selecione um segredo AWS Secrets Manager que contenha o token de envio do Logz.io. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

  Período de novas tentativas

  Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o Logz.io.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Defina as configurações de destino para o MongoDB Cloud

Esta seção descreve as opções para usar a MongoDB Cloud como destino. Para obter mais informações, consulte https://www.mongodb.com.

- Forneça os valores para os seguintes campos:

URL do webhook Realm do MongoDB

Especifique o URL do endpoint HTTP no formato a seguir.

```
https://webhooks.mongodb-realm.com
```

O URL deve ser um HTTPS URL.

Autenticação

Você pode optar por inserir a chave da API diretamente ou recuperar o segredo AWS Secrets Manager para acessar o MongoDB Cloud.

- Chave de API

  Entre em contato com o MongoDB Cloud para obter a chave de API necessária para habilitar a entrega de dados a esse endpoint a partir do Firehose.

- Secret

  Selecione um segredo AWS Secrets Manager que contenha a chave de API para o MongoDB Cloud. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

Codificação de conteúdo

O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo de uma solicitação antes de enviá-la ao destino. Escolha GZIP ou Desabilitado para habilitar/desabilitar a codificação de conteúdo da solicitação.

Período de novas tentativas

Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o provedor terceirizado selecionado.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Defina as configurações de destino para o New Relic

Esta seção descreve as opções para usar o New Relic como destino. Para obter mais informações, consulte https://newrelic.com.

- Forneça os valores para os seguintes campos:

  URL do endpoint HTTP

  Escolha o URL do endpoint HTTP entre as opções a seguir na lista suspensa.

  - Logs do New Relic - EUA
  - Métricas do New Relic - EUA
  - Métricas do New Relic - UE

Autenticação

Você pode optar por inserir a chave da API diretamente ou recuperar o segredo AWS Secrets Manager para acessar o New Relic.

- Chave de API

  Insira sua chave de licença, que é uma sequência hexadecimal de 40 caracteres, nas configurações da sua conta New Relic One. Você precisa dessa chave de API para permitir a entrega de dados do Firehose para esse endpoint.

- Secret

  Selecione um segredo AWS Secrets Manager que contenha a chave de API da New Relic. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

Codificação de conteúdo

O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo de uma solicitação antes de enviá-la ao destino. Escolha GZIP ou Desabilitado para habilitar/desabilitar a codificação de conteúdo da solicitação.

Período de novas tentativas

Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o endpoint HTTP da New Relic.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido.

Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Defina as configurações de destino para o Snowflake

Esta seção descreve as opções para usar o Snowflake como seu destino.

> **Note**
>
> A integração do Firehose com o Snowflake está disponível no Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon), Europa (Irlanda), Leste dos EUA (Ohio), Ásia-Pacífico (Tóquio), Europa (Frankfurt), Ásia-Pacífico (Cingapura), Ásia-Pacífico (Seul) e Ásia-Pacífico (Sydney). Regiões da AWS

Configurações de conexão

- Forneça os valores para os seguintes campos:

  URL da conta Snowflake

  Especifique o URL da conta do Snowflake. Por exemplo: `xy12345.us-east-1.aws.snowflakecomputing.com`. Consulte a documentação do Snowflake para

saber como determinar o URL da sua conta. Observe que você não deve especificar o número da porta, enquanto o protocolo (https://) é opcional.

Autenticação

Você pode optar por inserir o login do usuário, a chave privada e a senha manualmente ou recuperar o segredo para acessar o Snowflake. AWS Secrets Manager

- Login de usuário

  Especifique o usuário do Snowflake a ser usado para carregar dados. Certifique-se de que o usuário tenha acesso para inserir dados na tabela Snowflake.

- Chave privada

  Especifique a chave privada do usuário usada para autenticação com o Snowflake. Verifique se a chave privada está no `PKCS8` formato. Não inclua cabeçalho e rodapé do PEM como parte dessa chave. Se a chave estiver dividida em várias linhas, remova as quebras de linha.

- Senha

  Especifique a senha para descriptografar a chave privada criptografada. Você pode deixar esse campo vazio se a chave privada não estiver criptografada. Para obter informações, consulte Usando a autenticação de pares de chaves e a rotação de chaves.

- Secret

  Selecione um segredo AWS Secrets Manager que contenha as credenciais do Snowflake. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

Configuração da função

Usar a função padrão do Snowflake — Se essa opção for selecionada, o Firehose não passará nenhuma função para o Snowflake. A função padrão é assumida para carregar dados. Certifique-se de que a função padrão tenha permissão para inserir dados na tabela do Snowflake.

Use a função personalizada do Snowflake — Insira uma função não padrão do Snowflake a ser assumida pelo Firehose ao carregar dados na tabela do Snowflake.

Conectividade Snowflake

As opções são privadas ou públicas.

ID VPCE privada (opcional)

O VPCE ID do Firehose para se conectar de forma privada com o Snowflake. O formato do ID é com.amazonaws.vpce. *[região] .vpce-svc- [id].* Para obter mais informações, consulte AWS PrivateLink & Snowflake.

> **Note**
>
> Certifique-se de que sua rede Snowflake permita o acesso ao Firehose. Para obter uma lista das IDs de VPCE que você pode usar, consulte o. Acesso ao Snowflake em VPC

Configurar o banco de dados:

- Você deve especificar as configurações a seguir para usar o Snowflake como destino para seu stream do Firehose.
  - Banco de dados Snowflake — Todos os dados no Snowflake são mantidos em bancos de dados.
  - Esquema Snowflake — Cada banco de dados consiste em um ou mais esquemas, que são agrupamentos lógicos de objetos de banco de dados, como tabelas e visualizações
  - Tabela Snowflake — Todos os dados no Snowflake são armazenados em tabelas de banco de dados, estruturadas logicamente como coleções de colunas e linhas.

Opções de carregamento de dados para sua tabela Snowflake

- Use chaves JSON como nomes de colunas
- Use colunas VARIANT
  - Nome da coluna de conteúdo — especifique um nome de coluna na tabela, onde os dados brutos devem ser carregados.
  - Nome da coluna de metadados (opcional) — Especifique um nome de coluna na tabela, onde as informações de metadados devem ser carregadas.

Período de novas tentativas

Tempo de duração (0—7200 segundos) para o Firehose tentar novamente se a abertura do canal ou a entrega ao Snowflake falharem devido a problemas de serviço do Snowflake. O Firehose tenta novamente com um recuo exponencial até que a duração da nova tentativa termine. Se você definir a duração da nova tentativa para 0 (zero) segundos, o Firehose não tentará novamente em caso de falhas do Snowflake e encaminhará os dados para o bucket de erros do Amazon S3.

Defina as configurações de destino para o Splunk

Esta seção descreve as opções para usar o Splunk como destino.

> **Note**
>
> O Firehose entrega dados para clusters Splunk configurados com Classic Load Balancer ou Application Load Balancer.

- Forneça os valores para os seguintes campos:

  Splunk cluster endpoint

  Para determinar o endpoint, consulte Configurar o Amazon Data Firehose para enviar dados para a plataforma Splunk na documentação do Splunk.

  Splunk endpoint type

  Escolha `Raw endpoint` na maioria dos casos. Escolha `Event endpoint` se você pré-processou seus dados usando AWS Lambda para enviar dados para índices diferentes por tipo de evento. Para obter informações sobre qual endpoint usar, consulte Configurar o Amazon Data Firehose para enviar dados para a plataforma Splunk na documentação do Splunk.

  Autenticação

  Você pode optar por inserir o token de autenticação diretamente ou recuperar o segredo AWS Secrets Manager para acessar o Splunk.

  - Token de autenticação

    Para configurar um endpoint Splunk que possa receber dados do Amazon Data Firehose, consulte Visão geral da instalação e configuração do complemento Splunk para Amazon

Data Firehose na documentação do Splunk. Salve o token obtido do Splunk ao configurar o endpoint para esse stream do Firehose e adicione-o aqui.

- Secret

  Selecione um segredo AWS Secrets Manager que contenha o token de autenticação do Splunk. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

HEC acknowledgement timeout

Especifique quanto tempo o Amazon Data Firehose espera pela confirmação do índice do Splunk. Se o Splunk não enviar a confirmação antes que o tempo limite seja atingido, o Amazon Data Firehose considerará isso uma falha na entrega de dados. Em seguida, o Amazon Data Firehose tenta novamente ou faz backup dos dados no seu bucket do Amazon S3, dependendo do valor da duração da nova tentativa que você definir.

Período de novas tentativas

Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o Splunk.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação da Splunk. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o Splunk (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e aguarda uma confirmação do Splunk.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho do buffer recomendado para o destino varia de acordo com o provedor de serviços.

Defina as configurações de destino para o Splunk Observability Cloud

Esta seção descreve as opções para usar a Splunk Observability Cloud como destino. Para obter mais informações, consulte https://docs.splunk.com/observability/en/gdi/ get-data-in /connect/aws/aws-apiconfig.html # connect-to-aws-using - the-splunk-observability-cloud -api.

- Forneça os valores para os seguintes campos:

  URL do endpoint de ingestão na nuvem

  Você pode encontrar o URL de ingestão de dados em tempo real da Splunk Observability Cloud em Profile > Organizations > Real-time Data Ingest Endpoint no console do Splunk Observability.

  Autenticação

  Você pode optar por inserir o token de acesso diretamente ou recuperar o segredo AWS Secrets Manager para acessar o Splunk Observability Cloud.

  - Token de acesso

    Copie seu token de acesso ao Splunk Observability com o escopo de autorização INGEST dos Tokens de Acesso em Configurações no console do Splunk Observability.

  - Secret

    Selecione um segredo AWS Secrets Manager que contenha o token de acesso para o Splunk Observability Cloud. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

Codificação de conteúdo

O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo de uma solicitação antes de enviá-la ao destino. Escolha GZIP ou Desabilitado para habilitar/desabilitar a codificação de conteúdo da solicitação.

Período de novas tentativas

Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o endpoint HTTP selecionado.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino varia de acordo com o provedor de serviços.

Defina as configurações de destino para o Sumo Logic

Esta seção descreve as opções para usar o Sumo Logic como destino. Para obter mais informações, consulte https://www.sumologic.com.

- Forneça os valores para os seguintes campos:

  URL do endpoint HTTP

  Especifique a URL para o endpoint HTTP no seguinte formato: `https://deployment name.sumologic.net/receiver/v1/kinesis/dataType/access token`. A origem deve ser uma URL HTTPS.

  Codificação de conteúdo

  O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo de uma solicitação antes de enviá-la ao destino. Escolha GZIP ou Desabilitado para habilitar/desabilitar a codificação de conteúdo da solicitação.

  Período de novas tentativas

  Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para o Sumo Logic.

  Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

  Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado para o destino do Elastic varia de acordo com o provedor de serviços.

Defina as configurações de destino para a Elastic

Esta seção descreve as opções para usar o Elastic como destino.

- Forneça os valores para os seguintes campos:

  URL do endpoint do Elastic

  Especifique a URL para o endpoint HTTP no seguinte formato: `https://<cluster-id>.es.<region>.aws.elastic-cloud.com`. A origem deve ser uma URL HTTPS.

  Autenticação

  Você pode optar por inserir a chave de API diretamente ou recuperar o segredo AWS Secrets Manager para acessar a Elastic.

  - Chave de API

    Entre em contato com a Elastic para obter a chave de API necessária para permitir a entrega de dados do Firehose para o serviço deles.

- Secret

  Selecione um segredo AWS Secrets Manager que contenha a chave de API da Elastic. Se você não encontrar seu segredo na lista suspensa, crie um em. AWS Secrets Manager Para ter mais informações, consulte Autentique-se com o AWS Secrets Manager Amazon Data Firehose.

Codificação de conteúdo

O Amazon Data Firehose usa codificação de conteúdo para compactar o corpo de uma solicitação antes de enviá-la ao destino. Escolha GZIP (que é o selecionado por padrão) ou Desabilitado para habilitar/desabilitar a codificação de conteúdo da solicitação.

Período de novas tentativas

Especifique por quanto tempo o Amazon Data Firehose tenta enviar dados novamente para a Elastic.

Depois de enviar os dados, o Amazon Data Firehose primeiro espera por uma confirmação do endpoint HTTP. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o endpoint HTTP (seja a tentativa inicial ou uma nova tentativa), ele reinicia o contador de tempo limite de confirmação e espera por uma confirmação do endpoint HTTP.

Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o período de tempo limite da confirmação seja atingido. Se o tempo limite de confirmação expirar, o Amazon Data Firehose determinará se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Se você não quiser que o Amazon Data Firehose tente enviar dados novamente, defina esse valor como 0.

Parâmetros: opcional

O Amazon Data Firehose inclui esses pares de valores-chave em cada chamada HTTP. Esses parâmetros ajudam a identificar e organizar os destinos.

Dicas de armazenamento em buffer

O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los ao destino especificado. O tamanho de buffer recomendado quando o destino é O Elastic é de 1 MiB.

Definir configurações avançadas e de backup

Este tópico descreve como definir o backup e as configurações avançadas para seu stream do Firehose.

Definir configurações de backup

O Amazon Data Firehose usa o Amazon S3 para fazer backup de todos os dados ou somente os dados falhados que ele tenta entregar ao destino escolhido.

> **Important**
>
> - As configurações de backup só são suportadas se a origem do seu stream do Firehose for Direct PUT ou Kinesis Data Streams.
> - O recurso de buffer zero está disponível somente para os destinos do aplicativo e não está disponível para o destino de backup do Amazon S3.

Você pode especificar as configurações de backup do S3 para seu stream do Firehose se tiver feito uma das seguintes opções:

- Se você definir o Amazon S3 como destino para seu stream do Firehose e optar por especificar uma função AWS Lambda para transformar registros de dados ou se optar por converter formatos de registro de dados para seu stream do Firehose.
- Se você definir o Amazon Redshift como destino para seu stream do Firehose e optar por especificar uma função AWS Lambda para transformar registros de dados.

- Se você definir qualquer um dos seguintes serviços como destino para seu stream do Firehose: Amazon OpenSearch Service, Datadog, Dynatrace, HTTP Endpoint, LogicMonitor MongoDB Cloud, New Relic, Splunk ou Sumo Logic.

A seguir estão as configurações de backup do seu stream do Firehose.

- Backup do registro de origem no Amazon S3: se o S3 ou o Amazon Redshift for o destino selecionado, essa configuração indicará se você deseja habilitar o backup dos dados da fonte ou mantê-lo desabilitado. Se qualquer outro serviço compatível (exceto o S3 ou o Amazon Redshift) estiver definido como seu destino selecionado, essa configuração indicará se você deseja fazer backup de todos os dados da fonte ou apenas dos dados com falha.
- Bucket de backup S3 - este é o bucket S3 em que o Amazon Data Firehose faz backup de seus dados.
- Prefixo do bucket de backup S3 - esse é o prefixo em que o Amazon Data Firehose faz backup de seus dados.
- Prefixo da saída de erros do bucket de backup do S3: todos os dados com falha são copiados nesse prefixo da saída de erros do bucket do S3.
- Dicas de armazenamento em buffer, compactação e criptografia para backup — o Amazon Data Firehose usa o Amazon S3 para fazer backup de todos os dados ou somente os dados falhados que tenta entregar ao destino escolhido. O Amazon Data Firehose armazena em buffer os dados recebidos antes de entregá-los (fazer backup) para o Amazon S3. Você pode escolher um tamanho de buffer de 1 a 128 MiBs e um intervalo de buffer de 60 a 900 segundos. A condição que é satisfeita primeiro aciona a entrega de dados ao Amazon S3. Se você habilitar a transformação de dados, o intervalo de buffer se aplica desde o momento em que os dados transformados são recebidos pelo Amazon Data Firehose até a entrega dos dados para o Amazon S3. Se a entrega de dados para o destino ficar aquém da gravação de dados no stream do Firehose, o Amazon Data Firehose aumentará o tamanho do buffer dinamicamente para recuperá-lo. Essa ação ajuda a garantir que todos os dados sejam entregues no destino.
- Compressão S3 - escolha compressão de dados Snappy compatível com GZIP, Snappy, Zip ou Hadoop, ou nenhuma compressão de dados. A compactação Snappy compatível com Snappy, Zip e Hadoop não está disponível para o stream do Firehose com o Amazon Redshift como destino.
- Formato de extensão de arquivo S3 (opcional) — Especifique um formato de extensão de arquivo para objetos entregues ao bucket de destino do Amazon S3. Se você habilitar esse recurso, a extensão de arquivo especificada substituirá as extensões de arquivo padrão anexadas pelos recursos de conversão de formato de dados ou compactação do S3, como .parquet ou .gz.

Verifique se você configurou a extensão de arquivo correta ao usar esse recurso com a conversão de formato de dados ou a compactação S3. A extensão do arquivo deve começar com um ponto (.) e pode conter caracteres permitidos: 0-9a-z! -_.*' (). A extensão do arquivo não pode exceder 128 caracteres.

- O Firehose oferece suporte à criptografia do lado do servidor Amazon S3 AWS Key Management Service com (SSE-KMS) para criptografar dados entregues no Amazon S3. Você pode optar por usar o tipo de criptografia padrão especificado no bucket S3 de destino ou criptografar com uma chave da lista de AWS KMS chaves que você possui. Se você criptografar os dados com AWS KMS chaves, poderá usar a chave AWS gerenciada padrão (aws/s3) ou uma chave gerenciada pelo cliente. Para obter mais informações, consulte Proteção de dados usando criptografia do lado do servidor com chaves AWS gerenciadas pelo KMS (SSE-KMS).

Definir as configurações avançadas

A seção a seguir contém detalhes sobre as configurações avançadas do seu stream do Firehose.

- Criptografia do lado do servidor - O Amazon Data Firehose oferece suporte à criptografia do lado do servidor do Amazon S3 AWS com o Key Management Service (AWS KMS) para criptografar dados entregues no Amazon S3. Para obter mais informações, consulte Proteção de dados usando criptografia do lado do servidor com chaves gerenciadas pelo KMS (AWS SSE-KMS).
- Registro de erros - o Amazon Data Firehose registra erros relacionados ao processamento e à entrega. Além disso, quando a transformação de dados está ativada, ela pode registrar invocações do Lambda e enviar erros de entrega de dados para o Logs. CloudWatch Para obter mais informações, consulte Monitoramento do Amazon Data Firehose usando CloudWatch registros.

  > **Important**
  >
  > Embora opcional, é altamente recomendável ativar o registro de erros do Amazon Data Firehose durante a criação do stream do Firehose. Essa prática garante que você possa acessar os detalhes do erro em caso de falhas no processamento de registros ou na entrega.

- Permissões - O Amazon Data Firehose usa funções do IAM para todas as permissões que o stream do Firehose precisa. Você pode escolher criar uma nova função na qual as permissões necessárias são atribuídas automaticamente ou escolher uma função existente criada para o Amazon Data Firehose. A função é usada para conceder ao Firehose acesso a vários serviços,

incluindo seu bucket do S3, chave AWS KMS (se a criptografia de dados estiver ativada) e função Lambda (se a transformação de dados estiver ativada). O console talvez crie um perfil com espaços reservados. Para obter mais informações, consulte O que é IAM?.

- Tags - Você pode adicionar tags para organizar seus AWS recursos, monitorar custos e controlar o acesso.

  Se você especificar tags na `CreateDeliveryStream` ação, o Amazon Data Firehose executará uma autorização adicional na `firehose:TagDeliveryStream` ação para verificar se os usuários têm permissão para criar tags. Se você não fornecer essa permissão, as solicitações para criar novos streams do Firehose com tags de recursos do IAM falharão, conforme a seguir`AccessDeniedException`.

```
AccessDeniedException
User: arn:aws:sts::x:assumed-role/x/x is not authorized to perform:
 firehose:TagDeliveryStream on resource: arn:aws:firehose:us-east-1:x:deliverystream/
x with an explicit deny in an identity-based policy.
```

  O exemplo a seguir demonstra uma política que permite aos usuários criar um stream do Firehose e aplicar tags.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "firehose:CreateDeliveryStream",
            "Resource": "*",
            }
        },
        {
            "Effect": "Allow",
            "Action": "firehose:TagDeliveryStream",
            "Resource": "*",
            }
        }
    ]
}
```

Depois de escolher suas configurações avançadas e de backup, revise suas escolhas e escolha Create Firehose stream.

O novo stream do Firehose leva alguns minutos no estado Creating antes de ser disponibilizado. Depois que seu stream do Firehose estiver em um estado ativo, você poderá começar a enviar dados do seu produtor para ele.

Entenda as dicas de buffer

O Amazon Data Firehose armazena em buffer os dados de streaming recebidos na memória até um determinado tamanho (tamanho do buffer) e por um determinado período de tempo (intervalo de armazenamento em buffer) antes de entregá-los aos destinos especificados. Você usaria dicas de buffer quando quiser entregar arquivos de tamanho ideal para o Amazon S3 e obter melhor desempenho dos aplicativos de processamento de dados ou para ajustar a taxa de entrega do Firehose de acordo com a velocidade de destino.

Você pode configurar o tamanho do buffer e o intervalo do buffer ao criar novos streams do Firehose ou atualizar o tamanho do buffer e o intervalo de buffer nos streams existentes do Firehose. O tamanho do buffer é medido em MBs e o intervalo de buffer é medido em segundos. Contudo, se especificar um valor para um deles, você também deverá fornecer um valor para o outro. A primeira condição de buffer satisfeita aciona o Firehose para entregar os dados. Se você não configurar os valores de buffer, os valores padrão serão usados.

Você pode configurar dicas de buffer do Firehose por meio dos SDKs, ou. AWS Management Console AWS Command Line Interface AWS Para streams existentes, você pode reconfigurar dicas de buffer com um valor adequado aos seus casos de uso usando a opção Editar no console ou usando a API. UpdateDestination Para novos streams, você pode configurar dicas de buffer como parte da criação de um novo stream usando o console ou usando a API. CreateDeliveryStream Para ajustar o tamanho do buffer, defina `SizeInMBs` e `IntervalInSeconds` no `DestinationConfiguration` parâmetro específico de destino da API CreateDeliveryStreamou UpdateDestination.

> **Note**
>
> - Para atender às latências mais baixas dos casos de uso em tempo real, você pode usar a dica de intervalo de buffer zero. Quando você configura o intervalo de armazenamento em buffer como zero segundos, o Firehose não armazena dados em buffer e os entrega

em alguns segundos. Antes de alterar as dicas de buffer para um valor menor, consulte o fornecedor as dicas de buffer recomendadas do Firehose para seus destinos.

- O recurso de buffer zero está disponível somente para os destinos do aplicativo e não está disponível para o destino de backup do Amazon S3.

> **Note**
>
> O Firehose usa o upload de várias partes para o destino S3 quando você configura um intervalo de tempo de buffer inferior a 60 segundos para oferecer latências mais baixas. Devido ao upload de várias partes para o destino do S3, você verá algum aumento nos custos da PUT API do S3 se escolher um intervalo de tempo de buffer menor que 60 segundos.

Para intervalos de dicas de buffer e valores padrão específicos do destino, consulte a tabela a seguir:

Destination (Destino)	Tamanho do buffer em MB (padrão entre parênteses)	Intervalo de buffer em segundos (padrão entre parênteses)
S3	1-128 (5)	0-900 (300)
Redshift	1-128 (5)	0-900 (300)
OpenSearch Sem servidor	1-100 (5)	0-900 (300)
OpenSearch	1-100 (5)	0-900 (300)
Splunk	1-5 (5)	0-60 (60)
Datadog	1-4 (4)	0-900 (60)
Coralogix	1-64 (6)	0-900 (60)
Dynatrace	1-64 (5)	0-900 (60)

Destination (Destino)	Tamanho do buffer em MB (padrão entre parênteses)	Intervalo de buffer em segundos (padrão entre parênteses)
Elastic	1	0-900 (60)
Honeycomb	1-64 (15)	0-900 (60)
Ponto final HTTP	1-64 (5)	0-900 (60)
LogicMonitor	1-64 (5)	0-900 (60)
Logzio	1-64 (5)	0-900 (60)
MongoDB	1-16 (5)	0-900 (60)
Nova relíquia	1-64 (5)	0-900 (60)
SumoLogic	1-64 (1)	0-900 (60)
Splunk Observability Cloud	1-64 (1)	0-900 (60)

Teste o stream do Firehose com dados de amostra

Você pode usar o AWS Management Console para ingerir dados simulados de ações. O console executa um script em seu navegador para colocar registros de amostra em seu stream do Firehose. Isso permite que você teste a configuração do seu stream do Firehose sem precisar gerar seus próprios dados de teste.

Este é um exemplo dos dados simulados:

```
{"TICKER_SYMBOL":"QXZ","SECTOR":"HEALTHCARE","CHANGE":-0.05,"PRICE":84.51}
```

Observe que as cobranças padrão do Amazon Data Firehose se aplicam quando seu stream do Firehose transmite os dados, mas não há cobrança quando os dados são gerados. Para interromper essas cobranças, você pode parar o fluxo de exemplo no console a qualquer momento.

Conteúdos

Pré-requisitos

Antes de começar, crie um stream do Firehose. Para ter mais informações, consulte Crie um stream do Firehose.

Testar usando o Amazon S3 como destino

Use o procedimento a seguir para testar seu stream do Firehose usando o Amazon Simple Storage Service (Amazon S3) como destino.

Para testar um stream do Firehose usando o Amazon S3

1. Abra o console do Firehose em https://console.aws.amazon.com/firehose/.

2. Escolha um stream ativo do Firehose. O stream do Firehose precisa estar no status Ativo para que você possa começar a enviar dados.
3. Em Test with demo data, escolha Start sending demo data para gerar dados de exemplo do índice de ações.
4. Siga as instruções na tela para verificar se os dados estão sendo entregues ao bucket do S3. Observe que pode demorar alguns minutos para que os novos objetos apareçam no bucket, com base na configuração de armazenamento em buffer do bucket.
5. Quando o teste for concluído, escolha Stop sending demo data para cessar a cobrança de uso.

Testar usando o Amazon Redshift como destino

Use o procedimento a seguir para testar seu stream do Firehose usando o Amazon Redshift como destino.

Para testar um stream do Firehose usando o Amazon Redshift

1. Seu stream do Firehose espera que uma tabela esteja presente em seu cluster do Amazon Redshift. Conecte-se ao Amazon Redshift por meio de uma interface SQL e execute a instrução a seguir para criar uma tabela que aceite a amostra de dados.

```
create table firehose_test_table
(
 TICKER_SYMBOL varchar(4),
 SECTOR varchar(16),
 CHANGE float,
 PRICE float
);
```

2. Abra o console do Firehose em https://console.aws.amazon.com/firehose/.
3. Escolha um stream ativo do Firehose. O stream do Firehose precisa estar no status Ativo para que você possa começar a enviar dados.
4. Edite os detalhes do destino do seu stream do Firehose para apontar para a tabela recém-criada`firehose_test_table`.
5. Em Test with demo data, escolha Start sending demo data para gerar dados de exemplo do índice de ações.

6. Siga as instruções na tela para verificar se os dados estão sendo entregues na tabela. Observe que pode demorar alguns minutos para que novas linhas apareçam na tabela, com base na configuração de armazenamento em buffer.
7. Quando o teste for concluído, escolha Stop sending demo data para cessar a cobrança de uso.
8. Edite os detalhes do destino do seu stream do Firehose para apontar para outra tabela.
9. (Opcional) Exclua a tabela `firehose_test_table`.

Teste usando o OpenSearch serviço como destino

Use o procedimento a seguir para testar seu stream do Firehose usando o Amazon OpenSearch Service como destino.

Para testar um stream do Firehose usando o Service OpenSearch

1. Abra o console do Firehose em https://console.aws.amazon.com/firehose/.
2. Escolha um stream ativo do Firehose. O stream do Firehose precisa estar no status Ativo para que você possa começar a enviar dados.
3. Em Test with demo data, escolha Start sending demo data para gerar dados de exemplo do índice de ações.
4. Siga as instruções na tela para verificar se os dados estão sendo entregues ao seu domínio do OpenSearch Serviço. Para obter mais informações, consulte Pesquisando documentos em um domínio OpenSearch de serviço no Amazon OpenSearch Service Developer Guide.
5. Quando o teste for concluído, escolha Stop sending demo data para cessar a cobrança de uso.

Teste usando o Splunk como destino

Use o procedimento a seguir para testar seu stream do Firehose usando o Splunk como destino.

Para testar um stream do Firehose usando o Splunk

1. Abra o console do Firehose em https://console.aws.amazon.com/firehose/.
2. Escolha um stream ativo do Firehose. O stream do Firehose precisa estar no status Ativo para que você possa começar a enviar dados.
3. Em Test with demo data, escolha Start sending demo data para gerar dados de exemplo do índice de ações.

4. Verifique se os dados estão sendo entregues para o seu índice do Splunk. Exemplo de termos de pesquisa no Splunk são `sourcetype="aws:firehose:json"` e `index="name-of-your-splunk-index"`. Para obter mais informações sobre como pesquisar eventos no Splunk, consulte Pesquisar manual na documentação do Splunk.

   Se os dados de teste não aparecerem no índice do Splunk, verifique se há eventos com falha no bucket do Amazon S3. Consulte também Dados não entregues ao Splunk.
5. Quando concluir o teste, escolha Stop sending demo data para cessar a cobrança de uso.

Enviar dados para um stream do Firehose

Você pode enviar dados para seu stream do Firehose a partir de fontes como o Kinesis Data Stream, o Amazon MSK, o Kinesis Agent ou a API Amazon Data Firehose usando o SDK. AWS Você também pode usar Amazon CloudWatch Logs, CloudWatch Events ou AWS IoT como sua fonte de dados. Se você é novo no Amazon Data Firehose, reserve um tempo para se familiarizar com os conceitos e a terminologia apresentados em. O que é o Amazon Data Firehose?

> **Note**
>
> Alguns AWS serviços só podem enviar mensagens e eventos para um stream do Firehose que esteja na mesma região. Se seu stream do Firehose não aparecer como uma opção quando você estiver configurando um destino para Amazon CloudWatch Logs, CloudWatch Events AWS IoT, ou verifique se seu stream do Firehose está na mesma região que seus outros serviços.

Tópicos

Escrevendo no Amazon Data Firehose usando o Kinesis Data Streams

Você pode configurar o Amazon Kinesis Data Streams para enviar informações para um stream do Firehose.

> **Important**
>
> Se você usar a Kinesis Producer Library (KPL) para gravar dados em um fluxo de dados do Kinesis, poderá usar agregação para combinar os registros gravados. Se você então usar esse stream de dados como fonte para seu stream do Firehose, o Amazon Data Firehose desagregará os registros antes de entregá-los ao destino. Se você configurar seu stream do Firehose para transformar os dados, o Amazon Data Firehose desagregará os registros antes de entregá-los. AWS Lambda Para obter mais informações, consulte Developing Amazon Kinesis Data Streams Producers Using the Kinesis Producer Library e Aggregation.

1. Faça login AWS Management Console e abra o console do Amazon Data Firehose em https://console.aws.amazon.com/firehose/.
2. Escolha Create Firehose stream. Na página Name and source (Nome e origem), forneça valores para os seguintes campos:

   Nome do stream do Firehose

   O nome do seu stream do Firehose.

   Origem

   Escolha Kinesis stream para configurar um stream do Firehose que usa um stream de dados do Kinesis como fonte de dados. Em seguida, você pode usar o Amazon Data Firehose para ler dados facilmente de um stream de dados existente e carregá-los nos destinos.

   Para usar um streaming de dados do Kinesis como fonte, escolha um fluxo existente na lista Fluxo do Kinesis ou escolha Criar novo para criar um novo fluxo de dados do Kinesis. Após criar um novo ´fluxo, selecione Atualizar para atualizar a lista Fluxo do Kinesis. Se você tiver um grande número de fluxos, filtre a lista com a opção Filter by name.

   > **Note**
   >
   > Quando você configura um stream de dados do Kinesis como origem de um stream do Firehose, o Amazon Data `PutRecord` Firehose e as operações são desativados. `PutRecordBatch` Para adicionar dados ao seu stream do Firehose nesse caso, use o Kinesis Data Streams and operations. `PutRecord` `PutRecords`

O Amazon Data Firehose começa a ler dados da `LATEST` posição do seu stream do Kinesis. Para obter mais informações sobre as posições do Kinesis Data Streams, consulte. GetShardIterator

O Amazon Data Firehose chama a operação do Kinesis Data GetRecordsStreams uma vez por segundo para cada fragmento. No entanto, quando o backup completo está ativado, o Firehose chama a operação do Kinesis Data `GetRecords` Streams duas vezes por segundo para cada fragmento, um para o destino de entrega principal e outro para o backup completo.

Mais de um stream do Firehose pode ser lido do mesmo stream do Kinesis. Outras aplicações do Kinesis (consumidores) também podem ler o mesmo fluxo. Cada chamada de qualquer stream do Firehose ou de outro aplicativo de consumo é contabilizada no limite geral de limitação do fragmento. Para evitar a limitação, planeje seus aplicativos cuidadosamente. Para obter mais informações sobre os limites do Kinesis Data Streams, consulte Limites do Amazon Kinesis Data Streams.

3. Escolha Next para avançar até a página Configurar a transformação de registros e a conversão de formatos.

Escrevendo no Amazon Data Firehose usando o Amazon MSK

Você pode configurar o Amazon MSK para enviar informações para um stream do Firehose.

1. Faça login AWS Management Console e abra o console do Amazon Data Firehose em https://console.aws.amazon.com/firehose/.
2. Escolha Create Firehose stream.

   Na seção Escolher destino da página, forneça valores para os seguintes campos:

   Origem

   Escolha o Amazon MSK para configurar um stream do Firehose que usa o Amazon MSK como fonte de dados. Você pode escolher entre clusters provisionados pelo MSK e clusters do MSK Sem Servidor. Em seguida, você pode usar o Amazon Data Firehose para ler dados facilmente de um cluster e tópico específicos do Amazon MSK e carregá-los no destino S3 especificado.

Destination (Destino)

Escolha o Amazon S3 como destino para seu stream do Firehose.

Na seção Configurações da fonte da página, forneça valores para os seguintes campos:

Conectividade com o cluster do Amazon MSK

Escolha a opção Agentes privados de bootstrap (recomendado) ou Agentes públicos de bootstrap de acordo com a configuração do cluster. Os agentes de bootstrap são o que o cliente Apache Kafka usa como ponto de partida para se conectar ao cluster. Os corretores de bootstrap públicos são destinados ao acesso público externo AWS, enquanto os corretores de bootstrap privados são destinados ao acesso interno. AWS Para obter mais informações sobre o Amazon MSK, consulte Amazon Managed Streaming for Apache Kafka.

Para se conectar a um cluster do Amazon MSK provisionado ou sem servidor por meio de agentes privados de bootstrap, o cluster deve atender a todos os requisitos a seguir.

- O cluster deve estar ativo.
- O cluster deve ter o IAM como um dos métodos de controle de acesso.
- A conectividade privada de várias VPCs deve estar habilitada para o método de controle de acesso do IAM.
- Você deve adicionar a esse cluster uma política baseada em recursos que conceda ao diretor do serviço Amazon Data Firehose a permissão para invocar a API do Amazon MSK. CreateVpcConnection

Para se conectar a um cluster do Amazon MSK provisionado por meio de agentes de bootstrap públicos, o cluster deve atender a todos os requisitos a seguir.

- O cluster deve estar ativo.
- O cluster deve ter o IAM como um dos métodos de controle de acesso.
- O cluster deve ser acessível ao público.

Cluster do Amazon MSK

Para o mesmo cenário de conta, especifique o ARN do cluster Amazon MSK de onde seu stream do Firehose lerá os dados.

Para um cenário entre contas, consulte Entrega entre contas da Amazon MSK.

Tópico

Especifique o tópico do Apache Kafka do qual você deseja que seu stream do Firehose consuma dados. Depois que o stream do Firehose for criado, você não poderá atualizar esse tópico.

Na seção Nome do stream Firehose da página, forneça valores para os seguintes campos:

Nome do stream Firehose

Especifique o nome do seu stream do Firehose.

3. Em seguida, você pode concluir a etapa opcional de configuração da transformação de registros e da conversão de formato de registros. Para ter mais informações, consulte Configurar a transformação de registros e a conversão de formatos.

Escrevendo para o Amazon Data Firehose usando o Kinesis Agent

O agente do Amazon Kinesis é um aplicativo de software Java independente que serve como uma implementação de referência para mostrar como você pode coletar e enviar dados para o Firehose. O agente monitora continuamente um conjunto de arquivos e envia novos dados para seu stream do Firehose. O agente mostra como você pode lidar com a rotação de arquivos, fazer o checkpoint e tentar novamente em caso de falhas. Mostra como você pode fornecer seus dados de maneira confiável, oportuna e simples. Também mostra como você pode emitir CloudWatch métricas para melhor monitorar e solucionar problemas no processo de streaming. Para saber mais, awslabs/amazon-kinesis-agent.

Por padrão, os registros são analisados em cada arquivo com base no caractere de nova linha (`'\n'`). No entanto, o agente também pode ser configurado para analisar registros de várias linhas (consulte Configurações do agente).

Você pode instalar o agente em ambientes de servidor baseados no Linux, como servidores web, servidores de log e servidores de banco de dados. Depois de instalar o agente, configure-o especificando os arquivos a serem monitorados e o stream do Firehose para os dados. Depois que o agente é configurado, ele coleta dados dos arquivos de forma durável e os envia de forma confiável para o stream do Firehose.

Tópicos

Pré-requisitos

- O sistema operacional deve ser o Amazon Linux, ou o Red Hat Enterprise Linux versão 7 ou posterior.
- O agente versão 2.0.0 ou posterior é executado usando o JRE versão 1.8 ou posterior. O agente versão1.1x é executado usando o JRE versão 1.7 ou posterior.
- Se você estiver usando o Amazon EC2; para executar o agente, inicie a instância do EC2.
- A função ou AWS as credenciais do IAM que você especificar devem ter permissão para realizar a operação do Amazon Data PutRecordBatchFirehose para que o agente envie dados para seu stream do Firehose. Se você ativar o CloudWatch monitoramento para o agente, a permissão para realizar a CloudWatch PutMetricDataoperação também será necessária. Para obter mais informações, consulte, Controle de acesso com o Amazon Data FirehoseMonitorar a integridade do Kinesis Agent, e Autenticação e controle de acesso para a Amazon CloudWatch.

Credenciais

Gerencie suas AWS credenciais usando um dos seguintes métodos:

- Crie um provedor de credenciais personalizadas. Para obter detalhes, consulte the section called “Provedores de credenciais personalizados”.
- Especifique uma função do IAM ao executar a instância do EC2.

- Especifique AWS as credenciais ao configurar o agente (veja as entradas para `awsAccessKeyId` e `awsSecretAccessKey` na tabela de configuração abaixothe section called “Configurações do agente”).
- Edite `/etc/sysconfig/aws-kinesis-agent` para especificar sua AWS região e chaves de AWS acesso.
- Se sua instância do EC2 estiver em uma AWS conta diferente, crie uma função do IAM para fornecer acesso ao serviço Amazon Data Firehose. Especifique essa função ao configurar o agente (consulte AssumeroLearn e IdassumeRoleExternal). Use um dos métodos anteriores para especificar AWS as credenciais de um usuário na outra conta que tenha permissão para assumir essa função.

Provedores de credenciais personalizados

É possível criar um provedor de credenciais personalizadas e fornecer seu nome de classe e caminho jar ao Kinesis Agent nas seguintes configurações: `userDefinedCredentialsProvider.classname` e `userDefinedCredentialsProvider.location`. Para obter as descrições dessas duas configurações, consulte the section called “Configurações do agente”.

Para criar um provedor de credenciais personalizadas, defina uma classe que implemente a interface `AWS CredentialsProvider`, como a do exemplo a seguir.

```
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;

public class YourClassName implements AWSCredentialsProvider {
    public YourClassName() {
    }

    public AWSCredentials getCredentials() {
        return new BasicAWSCredentials("key1", "key2");
    }

    public void refresh() {
    }
}
```

Sua classe deve ter um construtor que não aceite argumentos.

AWS invoca o método de atualização periodicamente para obter credenciais atualizadas. Se você quiser que seu provedor de credenciais forneça credenciais diferentes ao longo da vida útil, inclua código para atualizar as credenciais neste método. Também é possível deixar esse método vazio se quiser um provedor de credenciais que venda credenciais estáticas (sem alteração).

Download e instalação do agente

Primeiro, conecte-se à instância. Para obter mais informações, consulte Connect to Your Instance no Guia do usuário do Amazon EC2. Se você tiver problemas para se conectar, consulte Solução de problemas de conexão com sua instância no Guia do usuário do Amazon EC2.

Em seguida, instale o agente usando um dos métodos a seguir.

- Para configurar o agente a partir dos repositórios do Amazon Linux

  Esse método só funciona para instâncias do Amazon Linux. Use o seguinte comando:

  ```
  sudo yum install -y aws-kinesis-agent
  ```

  O agente versão 2.0.0 ou posterior é instalado em computadores com o sistema operacional Amazon Linux 2 (AL2). Essa versão do agente requer o Java versão 1.8 ou posterior. Se a versão Java requerida ainda não estiver presente, o processo de instalação do agente a instalará. Para obter mais informações sobre o Amazon Linux 2, consulte https://aws.amazon.com/amazon-linux-2/.
- Para configurar o agente a partir dos repositórios do Amazon S3

  Esse método funciona para o Red Hat Enterprise Linux e para instâncias do Amazon Linux 2, pois instala o agente a partir do repositório disponível publicamente. Use o comando a seguir para baixar e instalar a versão mais recente do agente versão 2.x.x:

  ```
  sudo yum install -y https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-agent-latest.amzn2.noarch.rpm
  ```

  Para instalar uma versão específica do agente, especifique o número da versão no comando. Por exemplo, o comando a seguir instala o agente versão 2.0.1.

```
sudo yum install -y https://streaming-data-agent.s3.amazonaws.com/aws-kinesis-
agent-2.0.1-1.amzn1.noarch.rpm
```

Se você tiver o Java 1.7 e não quiser atualizá-lo, poderá baixar o agente versão 1.x.x que é compatível com o Java 1.7. Por exemplo, para baixar o agente v1.1.6, você pode usar o seguinte comando:

```
sudo yum install -y https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-
agent-1.1.6-1.amzn1.noarch.rpm
```

O agente v1.x.x mais recente pode ser baixado usando o seguinte comando:

```
sudo yum install -y https://s3.amazonaws.com/streaming-data-agent/aws-kinesis-agent-
latest.amzn1.noarch.rpm
```

- Para configurar o agente a partir do GitHub repositório

  1. Primeiro, certifique-se de que a versão do Java requerida esteja instalada, dependendo da versão do agente.
  2. Faça o download do agente do awslabs/ repo amazon-kinesis-agent GitHub .
  3. Instale o agente navegando até o diretório de download e executando o comando a seguir:

     ```
     sudo ./setup --install
     ```

- Para configurar o agente em um contêiner do Docker

  O Kinesis Agent também pode ser executado em um contêiner por meio da base de contêineres amazonlinux. Use o Dockerfile a seguir e depois execute o `docker build`.

  ```
  FROM amazonlinux
  ```

```
RUN yum install -y aws-kinesis-agent which findutils
COPY agent.json /etc/aws-kinesis/agent.json

CMD ["start-aws-kinesis-agent"]
```

Configuração e inicialização do agente

Para configurar e iniciar o agente

1. Abra e edite o arquivo de configuração (como superusuário, se você estiver usando permissões padrão de acesso a arquivos): `/etc/aws-kinesis/agent.json`

   Nesse arquivo de configuração, especifique os arquivos (`"filePattern"`) dos quais o agente coleta dados e o nome do stream do Firehose `"deliveryStream"` () para o qual o agente envia dados. O nome do arquivo é um padrão, e o agente reconhece os rodízios de arquivos. Você só pode fazer o rodízio de arquivos ou criar novos arquivos uma vez por segundo, no máximo. O agente usa o registro de data e hora de criação do arquivo para determinar quais arquivos devem ser rastreados e incluídos no stream do Firehose. A criação de novos arquivos ou o rodízio de arquivos em uma frequência superior a uma vez por segundo não permite que o agente faça a distinção entre eles corretamente.

   ```
   {
      "flows": [
           {
               "filePattern": "/tmp/app.log*",
               "deliveryStream": "yourdeliverystream"
           }
      ]
   }
   ```

   A AWS região padrão é `us-east-1`. Se você estiver usando outra região, adicione a configuração `firehose.endpoint` ao arquivo de configuração, especificando o endpoint para a sua região. Para ter mais informações, consulte Configurações do agente.

2. Inicie o agente manualmente:

   ```
   sudo service aws-kinesis-agent start
   ```

3. (Opcional) Configure o agente para ser iniciado durante o startup do sistema:

```
sudo chkconfig aws-kinesis-agent on
```

Agora o agente está sendo executado como um serviço do sistema em segundo plano. Ele monitora continuamente os arquivos especificados e envia dados para o stream especificado do Firehose. A atividade do agent é registrada em `/var/log/aws-kinesis-agent/aws-kinesis-agent.log`.

Configurações do agente

O agente oferece suporte a duas configurações obrigatórias, `filePattern` e `deliveryStream`, além das configurações opcionais de recursos adicionais. É possível especificar configurações obrigatórios e opcionais em `/etc/aws-kinesis/agent.json`.

Sempre que você alterar o arquivo de configuração, deverá interromper e iniciar o agente, usando os seguintes comandos:

```
sudo service aws-kinesis-agent stop
sudo service aws-kinesis-agent start
```

Se desejar, você pode usar o seguinte comando:

```
sudo service aws-kinesis-agent restart
```

Estas são as configurações gerais.

Definição da configuração	Descrição
`assumeRoleARN`	O nome de recurso da Amazon (ARN) da função a ser assumida pelo usuário. Para obter mais informações, consulte Delegar acesso entre AWS contas usando funções do IAM no Guia do usuário do IAM.
`assumeRoleExternalId`	Um identificador opcional que determina quem pode assumir a função. Para obter mais informações, consulte Como usar um ID externo no Guia do usuário do IAM.

Definição da configuração	Descrição
`awsAccessKeyId`	AWS ID da chave de acesso que substitui as credenciais padrão. Essa configuração tem precedência sobre todos os outros provedores de credenciais.
`awsSecretAccessKey`	AWS chave secreta que substitui as credenciais padrão. Essa configuração tem precedência sobre todos os outros provedores de credenciais.
`cloudwatch.emitMetrics`	Permite que o agente emita métricas para, CloudWatch se definidas (verdadeiras). Padrão: True
`cloudwatch.endpoint`	O endpoint regional para CloudWatch. Padrão: `monitoring.us-east-1.amazonaws.com`
`firehose.endpoint`	O endpoint regional do Amazon Data Firehose. Padrão: `firehose.us-east-1.amazonaws.com`
`sts.endpoint`	O endpoint regional do AWS Security Token Service. Padrão: `https://sts.amazonaws.com`
`userDefinedCredentialsProvider.classname`	Se você definir um provedor de credenciais personalizadas, forneça seu nome de classe totalmente qualificado usando essa configuração. Não inclua `.class` no final do nome da classe.
`userDefinedCredentialsProvider.location`	Se você definir um provedor de credenciais personalizadas, use essa configuração para especificar o caminho absoluto do jar que contém o provedor de credenciais personalizadas. O agente também procura o arquivo jar no seguinte local: `/usr/share/aws-kinesis-agent/lib/`.

Estas são as configurações de fluxo.

Definição da configuração	Descrição
`aggregatedRecordSizeBytes`	Para fazer com que o agente agregue registros e depois os coloque no stream do Firehose em uma operação, especifique essa configuração. Defina-o com o tamanho que você deseja que o registro agregado tenha antes que o agente o coloque no stream do Firehose. Padrão: 0 (sem agregação)
`dataProcessingOptions`	A lista de opções de processamento aplicadas a cada registro analisado antes de ser enviado ao stream do Firehose. As opções de processamento são executadas na ordem especificada. Para ter mais informações, consulte Usar o agente para pré-processar os dados.
`deliveryStream`	[Obrigatório] O nome do stream do Firehose.
`filePattern`	[Obrigatório] Um glob para os arquivos que precisam ser monitorados pelo agente. Qualquer arquivo que corresponda a esse padrão é selecionado pelo agente automaticamente e monitorado. Para todos os arquivos correspondentes a esse padrão, conceda permissão de leitura a `aws-kinesis-agent-user` . Para o diretório que contém os arquivos, conceda permissões de leitura e execução a `aws-kinesis-agent-user` . **Important** O agente seleciona qualquer arquivo que corresponda a esse padrão. Para garantir que o agente não selecione registros não intencionais, escolha esse padrão cuidadosamente.
`initialPosition`	A posição em que o arquivo começou a ser analisado. Os valores válidos são `START_OF_FILE` e `END_OF_FILE` . Padrão: `END_OF_FILE`
`maxBufferAgeMillis`	O tempo máximo, em milissegundos, durante o qual o agente armazena dados em buffer antes de enviá-los para o stream do Firehose.

Definição da configuração	Descrição
	Intervalo de valores: 1.000 a 900.000 (1 segundo a 15 minutos) Padrão: 60.000 (1 minuto)
`maxBufferSizeBytes`	O tamanho máximo, em bytes, para o qual o agente armazena dados em buffer antes de enviá-los para o stream do Firehose. Intervalo de valores: 1 a 4.194.304 (4 MB) Padrão: 4.194.304 (4 MB)
`maxBufferSizeRecords`	O número máximo de registros para os quais o agente armazena dados em buffer antes de enviá-los para o stream do Firehose. Intervalo de valores: 1 a 500 Padrão: 500
`minTimeBetweenFilePollsMillis`	O intervalo de tempo, em milissegundos, em que o agente consulta e analisa os arquivos monitorados em busca de novos dados. Intervalo de valores: 1 ou mais Padrão: 100
`multiLineStartPattern`	O padrão de identificação do início de um registro. Um registro é composto por uma linha que corresponde ao padrão e pelas linhas subsequentes que não correspondem ao padrão. Os valores válidos são expressões regulares. Por padrão, cada nova linha nos arquivos de log é analisada como um único registro.
`skipHeaderLines`	O número de linhas em que o agente ignorará a análise no início dos arquivos monitorados. Intervalo de valores: 0 ou mais Padrão: 0 (zero)

Definição da configuração	Descrição
`truncatedRecordTerminator`	A string que o agente usa para truncar um registro analisado quando o tamanho do registro excede o limite de tamanho do registro do Amazon Data Firehose. (1,000 KB) Padrão: `'\n'` (nova linha)

Monitoramento de vários diretórios de arquivos e gravação em vários streams

Ao especificar vários fluxos de configurações, você pode configurar o agente para monitorar vários diretórios de arquivos e enviar dados a vários streams. No exemplo de configuração a seguir, o agente monitora dois diretórios de arquivos e envia dados para um stream de dados do Kinesis e um stream do Firehose, respectivamente. Você pode especificar endpoints diferentes para o Kinesis Data Streams e o Amazon Data Firehose para que seu stream de dados e seu stream do Firehose não precisem estar na mesma região.

```
{
    "cloudwatch.emitMetrics": true,
    "kinesis.endpoint": "https://your/kinesis/endpoint",
    "firehose.endpoint": "https://your/firehose/endpoint",
    "flows": [
        {
            "filePattern": "/tmp/app1.log*",
            "kinesisStream": "yourkinesisstream"
        },
        {
            "filePattern": "/tmp/app2.log*",
            "deliveryStream": "yourfirehosedeliverystream"
        }
    ]
}
```

Para obter informações mais detalhadas sobre o uso do agente com o Amazon Kinesis Data Streams, consulte Writing to Amazon Kinesis Data Streams with Kinesis Agent.

Usar o agente para pré-processar os dados

O agente pode pré-processar os registros analisados dos arquivos monitorados antes de enviá-los para o stream do Firehose. Você pode habilitar esse recurso adicionando a configuração `dataProcessingOptions` ao fluxo de arquivos. Um ou mais opções de processamento podem ser adicionadas e serão executadas na ordem especificada.

O agente oferece suporte às seguintes opções de processamento. Como o agente é de código aberto, você pode desenvolver e estender ainda mais suas opções de processamento. Você pode baixar o agente em Kinesis Agent.

Opções de processamento

`SINGLELINE`

Converte um registro de várias linhas em um registro de única linha removendo caracteres de nova linha, e espaços à esquerda e à direita.

```
{
    "optionName": "SINGLELINE"
}
```

`CSVTOJSON`

Converte um registro com formato separado por delimitador em um registro com formato JSON.

```
{
    "optionName": "CSVTOJSON",
    "customFieldNames": [ "field1", "field2", ... ],
    "delimiter": "yourdelimiter"
}
```

`customFieldNames`

[Obrigatório] Os nomes de campo usados como chaves em cada par de valores de chave JSON. Por exemplo, se você especificar `["f1", "f2"]`, o registro "v1, v2" será convertido em `{"f1":"v1","f2":"v2"}`.

`delimiter`

A string usada como delimitador no registro. O padrão é uma vírgula (,).

LOGTOJSON

Converte um registro com formato de log em um registro com formato JSON. Os formatos de log compatíveis são Apache Common Log, Apache Combined Log, Apache Error Log e RFC3164 Syslog.

```
{
    "optionName": "LOGTOJSON",
    "logFormat": "logformat",
    "matchPattern": "yourregexpattern",
    "customFieldNames": [ "field1", "field2", … ]
}
```

logFormat

[Obrigatório] O formato da entrada de log. Os valores possíveis são:

- `COMMONAPACHELOG`: o formato do Apache Common Log. Cada entrada de log tem o seguinte padrão: `"%{host} %{ident} %{authuser} [%{datetime}] \"%{request}\" %{response} %{bytes}"`.
- `COMBINEDAPACHELOG`: o formato do Apache Combined Log. Cada entrada de log tem o seguinte padrão: `"%{host} %{ident} %{authuser} [%{datetime}] \"%{request}\" %{response} %{bytes} %{referrer} %{agent}"`.
- `APACHEERRORLOG`: o formato do Apache Error Log. Cada entrada de log tem o seguinte padrão: `"[%{timestamp}] [%{module}:%{severity}] [pid %{processid}:tid %{threadid}] [client: %{client}] %{message}"`.
- `SYSLOG`: o formato do RFC3164 Syslog. Cada entrada de log tem o seguinte padrão: `"%{timestamp} %{hostname} %{program}[%{processid}]: %{message}"`.

matchPattern

Substitui o padrão do formato de log especificado. Use esta configuração para extrair valores de entradas de log, caso elas tenham um formato personalizado. Se você especificar `matchPattern`, também deverá especificar `customFieldNames`.

customFieldNames

Os nomes de campo personalizados usados como chaves em cada par de valores de chave JSON. Você pode usar essa configuração para definir nomes de campo para valores extraídos de `matchPattern` ou substituir os nomes de campo padrão de formatos de log predefinidos.

Example : Configuração LOGTOJSON

Aqui está um exemplo de uma configuração `LOGTOJSON` para uma entrada Apache Common Log convertida em formato JSON:

```
{
    "optionName": "LOGTOJSON",
    "logFormat": "COMMONAPACHELOG"
}
```

Antes da conversão:

```
64.242.88.10 - - [07/Mar/2004:16:10:02 -0800] "GET /mailman/listinfo/hsdivision
 HTTP/1.1" 200 6291
```

Depois da conversão:

```
{"host":"64.242.88.10","ident":null,"authuser":null,"datetime":"07/
Mar/2004:16:10:02 -0800","request":"GET /mailman/listinfo/hsdivision
 HTTP/1.1","response":"200","bytes":"6291"}
```

Example : Configuração LOGTOJSON com campos personalizados

Aqui está outro exemplo de configuração `LOGTOJSON`:

```
{
    "optionName": "LOGTOJSON",
    "logFormat": "COMMONAPACHELOG",
    "customFieldNames": ["f1", "f2", "f3", "f4", "f5", "f6", "f7"]
}
```

Com essa configuração, a mesma entrada Apache Common Log do exemplo anterior é convertida em formato JSON, da seguinte forma:

```
{"f1":"64.242.88.10","f2":null,"f3":null,"f4":"07/Mar/2004:16:10:02 -0800","f5":"GET /
mailman/listinfo/hsdivision HTTP/1.1","f6":"200","f7":"6291"}
```

Example : Conversão da entrada Apache Common Log

A configuração de fluxo a seguir converte uma entrada Apache Common Log em um registro de linha única no formato JSON:

```
{
    "flows": [
        {
            "filePattern": "/tmp/app.log*",
            "deliveryStream": "my-delivery-stream",
            "dataProcessingOptions": [
                {
                    "optionName": "LOGTOJSON",
                    "logFormat": "COMMONAPACHELOG"
                }
            ]
        }
    ]
}
```

Example : Conversão de registros de várias linhas

A configuração de fluxo a seguir analisa registros de várias linha cuja primeira linha começa com "`[SEQUENCE=`". Cada registro é convertido primeiro em um registro de única linha. Em seguida, os valores são extraídos do registro com base em um delimitador por tabulações. Os valores extraídos são mapeados para os valores `customFieldNames` especificados, a fim de formar um registro de linha única no formato JSON.

```
{
    "flows": [
        {
            "filePattern": "/tmp/app.log*",
            "deliveryStream": "my-delivery-stream",
            "multiLineStartPattern": "\\[SEQUENCE=",
            "dataProcessingOptions": [
                {
                    "optionName": "SINGLELINE"
                },
                {
                    "optionName": "CSVTOJSON",
                    "customFieldNames": [ "field1", "field2", "field3" ],
                    "delimiter": "\\t"
                }
            ]
        }
    ]
}
```

Example : Configuração LOGTOJSON com padrão de correspondência

Aqui está um exemplo de configuração LOGTOJSON referente a uma entrada Apache Common Log convertida em formato JSON, com o último campo (bytes) omitido:

```
{
    "optionName": "LOGTOJSON",
    "logFormat": "COMMONAPACHELOG",
    "matchPattern": "^([\\d.]+) (\\S+) (\\S+) \\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(.
+?)\" (\\d{3})",
    "customFieldNames": ["host", "ident", "authuser", "datetime", "request",
 "response"]
}
```

Antes da conversão:

```
123.45.67.89 - - [27/Oct/2000:09:27:09 -0400] "GET /java/javaResources.html HTTP/1.0"
 200
```

Depois da conversão:

```
{"host":"123.45.67.89","ident":null,"authuser":null,"datetime":"27/Oct/2000:09:27:09
 -0400","request":"GET /java/javaResources.html HTTP/1.0","response":"200"}
```

Comandos da CLI do agente

Inicie automaticamente o agente durante o startup do sistema:

```
sudo chkconfig aws-kinesis-agent on
```

Verifique o status do agente:

```
sudo service aws-kinesis-agent status
```

Interrompa o agente:

```
sudo service aws-kinesis-agent stop
```

Leia o arquivo de log do agente a partir deste local:

```
/var/log/aws-kinesis-agent/aws-kinesis-agent.log
```

Desinstale o agente:

```
sudo yum remove aws-kinesis-agent
```

Perguntas frequentes

Existe um Kinesis Agent para Windows?

O Kinesis Agent para Windows é um software diferente das plataformas do Kinesis Agent para Linux.

Por que o Kinesis Agent está ficando mais lento e/ou aumentando os `RecordSendErrors`?

Isso geralmente ocorre devido ao controle de utilização do Kinesis. Verifique a `WriteProvisionedThroughputExceeded` métrica do Kinesis Data Streams `ThrottledRecords` ou a métrica dos streams do Firehose. Qualquer aumento de 0 nessas métricas indica que os limites do fluxo precisam ser aumentados. Para obter mais informações, consulte Limites do Kinesis Data Stream e Streams do Firehose.

Depois de descartar o controle de utilização como causa, verifique se o Kinesis Agent está configurado para seguir um número grande de arquivos pequenos. Há um atraso quando o Kinesis Agent exibe os dados do final de um arquivo novo, portanto, o Kinesis Agent deveria estar exibindo os dados do final de um pequeno número de arquivos maiores. Tente consolidar os arquivos de log em arquivos maiores.

Por que estou recebendo exceções `java.lang.OutOfMemoryError` ?

O Kinesis Agent não tem memória suficiente para lidar com a workload atual. Tente aumentar `JAVA_START_HEAP` e `JAVA_MAX_HEAP` no `/usr/bin/start-aws-kinesis-agent` e reiniciar o agente.

Por que estou recebendo exceções `IllegalStateException : connection pool shut down`?

O Kinesis Agent não tem conexões suficientes para lidar com a workload atual. Tente aumentar `maxConnections` e `maxSendingThreads` nas configurações gerais do agente em `/etc/aws-kinesis/agent.json`. O valor padrão para esses campos é 12 vezes o número de processadores

de runtime disponíveis. Consulte AgentConfiguration.java para saber mais sobre as configurações avançadas do agente.

Como posso depurar outro problema com o Kinesis Agent?

Os logs do nível DEBUG podem ser habilitados em `/etc/aws-kinesis/log4j.xml`.

Como devo configurar o Kinesis Agent?

Quanto menor o `maxBufferSizeBytes`, mais frequentemente o Kinesis Agent enviará dados. Isso pode ser bom, pois diminui o tempo de entrega dos registros, mas também aumenta as solicitações por segundo feitas ao Kinesis.

Por que o Kinesis Agent está enviando registros duplicados?

Isso ocorre devido a uma configuração incorreta da exibição dos dados do final dos arquivos. Certifique-se de que cada `fileFlow's filePattern` corresponda a apenas um arquivo. Isso também pode ocorrer se o modo `logrotate` que está sendo usado estiver no modo `copytruncate`. Tente mudar o modo para o modo padrão ou criar para evitar duplicações. Para obter mais informações sobre como lidar com registros duplicados, consulte Handling Duplicate Records.

Escrevendo no Amazon Data Firehose com o SDK AWS

Você pode usar a API Amazon Data Firehose para enviar dados para um stream do Firehose usando o SDK para AWS Java, .NET, Node.js, Python ou Ruby. Se você é novo no Amazon Data Firehose, reserve um tempo para se familiarizar com os conceitos e a terminologia apresentados em. O que é o Amazon Data Firehose? Para obter mais informações, consulte Comece a desenvolver usando a Amazon Web Services.

Esses exemplos não representam um código pronto para produção, pois não verificam todas as exceções possíveis nem abrangem todas as considerações de segurança ou de performance possíveis.

A API Amazon Data Firehose oferece duas operações para enviar dados para seu stream do Firehose: e. PutRecordPutRecordBatch `PutRecord()`envia um registro de dados em uma chamada e `PutRecordBatch()` pode enviar vários registros de dados em uma chamada.

Tópicos

- Operações de gravação única usando PutRecord

Operações de gravação única usando PutRecord

A colocação de dados requer somente o nome do stream Firehose e um buffer de bytes (<=1000 KB). Como o Amazon Data Firehose agrupa vários registros antes de carregar o arquivo no Amazon S3, talvez você queira adicionar um separador de registros. Para colocar dados, um registro por vez, em um stream do Firehose, use o código a seguir:

```
PutRecordRequest putRecordRequest = new PutRecordRequest();
putRecordRequest.setDeliveryStreamName(deliveryStreamName);

String data = line + "\n";

Record record = new Record().withData(ByteBuffer.wrap(data.getBytes()));
putRecordRequest.setRecord(record);

// Put record into the DeliveryStream
firehoseClient.putRecord(putRecordRequest);
```

Para mais contexto de código, consulte o exemplo de código incluído no AWS SDK. Para obter informações sobre a sintaxe de solicitação e resposta, consulte o tópico relevante em Firehose API Operations.

Operações de gravação em lote usando PutRecordBatch

A colocação de dados requer apenas o nome do stream Firehose e uma lista de registros. Como o Amazon Data Firehose agrupa vários registros antes de carregar o arquivo no Amazon S3, talvez você queira adicionar um separador de registros. Para colocar registros de dados em lotes em um stream do Firehose, use o código a seguir:

```
PutRecordBatchRequest putRecordBatchRequest = new PutRecordBatchRequest();
putRecordBatchRequest.setDeliveryStreamName(deliveryStreamName);
putRecordBatchRequest.setRecords(recordList);

// Put Record Batch records. Max No.Of Records we can put in a
// single put record batch request is 500
firehoseClient.putRecordBatch(putRecordBatchRequest);

recordList.clear();
```

Para mais contexto de código, consulte o exemplo de código incluído no AWS SDK. Para obter informações sobre a sintaxe de solicitação e resposta, consulte o tópico relevante em Firehose API Operations.

Gravando no Amazon Data Firehose usando registros CloudWatch

CloudWatch Os eventos de registros podem ser enviados para o Firehose usando filtros de CloudWatch assinatura. Para obter mais informações, consulte Filtros de assinatura com o Amazon Data Firehose.

CloudWatch Os eventos de registros são enviados para o Firehose no formato gzip compactado. Se você quiser entregar eventos de log descompactados para destinos do Firehose, você pode usar o recurso de descompactação no Firehose para descompactar automaticamente os registros. CloudWatch

> **Important**
>
> Atualmente, o Firehose não suporta a entrega de CloudWatch registros para o destino do Amazon OpenSearch Service porque a Amazon CloudWatch combina vários eventos de log em um registro Firehose e o Amazon OpenSearch Service não pode aceitar vários eventos de log em um registro. Como alternativa, você pode considerar o uso do filtro de assinatura do Amazon OpenSearch Service em CloudWatch registros.

Descompressão de registros CloudWatch

Se você estiver usando o Firehose para entregar CloudWatch registros e quiser entregar dados descompactados para o destino do stream do Firehose, use o Firehose Data Format Conversion (Parquet, ORC) ou o particionamento dinâmico. Você deve ativar a descompressão para seu stream do Firehose.

Você pode ativar a descompressão usando o AWS Management Console, AWS Command Line Interface ou AWS SDKs.

> **Note**
>
> Se você ativar o recurso de descompressão em um stream, use esse stream exclusivamente para filtros de assinaturas do CloudWatch Logs, e não para Vended Logs. Se você ativar o recurso de descompressão em um stream usado para ingerir CloudWatch registros e

registros vendidos, a ingestão de registros vendidos no Firehose falhará. Esse recurso de descompressão é somente para CloudWatch registros.

Extração de mensagens após a descompressão dos registros CloudWatch

Ao ativar a descompressão, você também tem a opção de ativar a extração de mensagens. Ao usar a extração de mensagens, o Firehose filtra todos os metadados, como proprietário, grupo de registros, fluxo de registros e outros, dos registros descompactados do CloudWatch Logs e entrega somente o conteúdo dentro dos campos da mensagem. Se você estiver entregando dados para um destino do Splunk, deverá ativar a extração de mensagens para que o Splunk analise os dados. A seguir estão exemplos de saídas após a descompressão com e sem extração de mensagens.

Figura 1: Saída da amostra após a descompressão sem extração de mensagem:


```
{
 "owner": "111111111111",
 "logGroup": "CloudTrail/logs",
 "logStream": "111111111111_CloudTrail/logs_us-east-1",
 "subscriptionFilters": [
 "Destination"
 ],
 "messageType": "DATA_MESSAGE",
 "logEvents": [
 {
 "id": "31953106606966983378809025079804211143289615424298221568",
 "timestamp": 1432826855000,
 "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root1\"}"
 },
 {
 "id": "31953106606966983378809025079804211143289615424298221569",
 "timestamp": 1432826855000,
 "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root2\"}"
 },
 {
 "id": "31953106606966983378809025079804211143289615424298221570",
 "timestamp": 1432826855000,
 "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root3\"}"
 }
 ]
}
```

Figura 2: Saída da amostra após a descompressão com extração de mensagem:

```
{"eventVersion":"1.03","userIdentity":{"type":"Root1"}
{"eventVersion":"1.03","userIdentity":{"type":"Root2"}
{"eventVersion":"1.03","userIdentity":{"type":"Root3"}
```

Ativando e desativando a descompressão

Você pode ativar e desativar a descompressão usando o AWS Management Console, AWS Command Line Interface ou AWS SDKs.

Habilitando a descompressão em um novo fluxo de dados usando o AWS Management Console

Para habilitar a descompressão em um novo fluxo de dados usando o AWS Management Console

1. Faça login AWS Management Console e abra o console do Kinesis em https://console.aws.amazon.com/kinesis.
2. Escolha Amazon Data Firehose no painel de navegação.
3. Escolha Create Firehose stream.
4. Em Escolha a origem e o destino

   Origem

   A fonte do seu stream do Firehose. Escolha uma das seguintes fontes:

   - Direct PUT — Escolha essa opção para criar um stream do Firehose no qual os aplicativos produtores gravam diretamente. Para obter uma lista de AWS serviços, agentes e serviços de código aberto integrados ao Direct PUT no Firehose, consulte esta seção.
   - Stream do Kinesis: escolha essa opção para configurar um stream do Firehose que usa um stream de dados do Kinesis como fonte de dados. Em seguida, você pode usar o Firehose para ler dados facilmente de um stream de dados existente do Kinesis e carregá-los nos destinos. Para obter mais informações, consulte Gravando no Firehose usando o Kinesis Data Streams

   Destination (Destino)

   O destino do seu stream do Firehose. Escolha uma das seguintes opções:

   - Amazon S3

- Splunk

5. Em Firehose stream name, insira um nome para seu stream.
6. (Opcional) Em Transformar registros:
   - Na seção Descompactar registros de origem do Amazon CloudWatch Logs, escolha Ativar descompressão.
   - Se você quiser usar a extração de mensagens após a descompactação, escolha Ativar extração de mensagens.

Habilitando a descompressão em um fluxo de dados existente usando o AWS Management Console

Se você tiver um stream do Firehose com uma função Lambda para realizar a descompressão, poderá substituí-lo pelo recurso de descompressão do Firehose. Antes de continuar, revise o código da função Lambda para confirmar se ele só executa a descompressão ou a extração de mensagens. A saída da função Lambda deve ser semelhante aos exemplos mostrados na Figura 1 ou Figura 2 na seção anterior. Se a saída for semelhante, você poderá substituir a função Lambda usando as etapas a seguir.

1. Substitua sua função Lambda atual por esse esquema. A nova função Lambda do blueprint detecta automaticamente se os dados recebidos estão compactados ou descompactados. Ele só executa a descompressão se os dados de entrada estiverem compactados.
2. Ative a descompressão usando a opção Firehose integrada para descompressão.
3. Ative CloudWatch as métricas para seu stream do Firehose, caso ainda não esteja ativado. Monitore a métrica CloudWatchProcessorLambda _ IncomingCompressedData e espere até que essa métrica mude para zero. Isso confirma que todos os dados de entrada enviados para sua função Lambda estão descompactados e que a função Lambda não é mais necessária.
4. Remova a transformação de dados do Lambda porque você não precisa mais dela para descompactar seu stream.

Desativando a descompressão usando o AWS Management Console

Para desativar a descompressão em um fluxo de dados usando o AWS Management Console

1. Faça login AWS Management Console e abra o console do Kinesis em https://console.aws.amazon.com/kinesis.
2. Escolha Amazon Data Firehose no painel de navegação.
3. Escolha o stream do Firehose que você deseja editar.
4. Na página de detalhes do stream do Firehose, escolha a guia Configuração.
5. Na seção Transformar e converter registros, escolha Editar.
6. Em Descompactar registros de origem do Amazon CloudWatch Logs, desmarque Ativar descompressão e escolha Salvar alterações.

Perguntas frequentes

O que acontece com os dados de origem em caso de erro durante a descompressão?

Se o Amazon Data Firehose não conseguir descompactar o registro, o registro será entregue como está (em formato compactado) para o bucket S3 de erro que você especificou durante a criação do stream do Firehose. Junto com o registro, o objeto entregue também inclui código de erro e mensagem de erro, e esses objetos serão entregues a um prefixo de bucket do S3 chamado. `decompression-failed` O Firehose continuará processando outros registros após uma falha na descompressão de um registro.

O que acontece com os dados de origem em caso de erro no pipeline de processamento após a descompressão bem-sucedida?

Se o Amazon Data Firehose cometer erros nas etapas de processamento após a descompactação, como particionamento dinâmico e conversão de formato de dados, o registro será entregue em formato compactado para o bucket S3 de erro que você especificou durante a criação do stream do Firehose. Junto com o registro, o objeto entregue também inclui código de erro e mensagem de erro.

Como você é informado em caso de erro ou exceção?

Em caso de erro ou exceção durante a descompactação, se você configurar o Logs, o Firehose CloudWatch registrará as mensagens CloudWatch de erro no Logs. Além disso, o Firehose envia métricas para CloudWatch métricas que você pode monitorar. Opcionalmente, você também pode criar alarmes com base nas métricas emitidas pelo Firehose.

O que acontece quando **`put`** as operações não vêm do CloudWatch Logs?

Quando `puts` o cliente não vem do CloudWatch Logs, a seguinte mensagem de erro é retornada:

```
Put to Firehose failed for AccountId: <accountID>, FirehoseName: <firehosename> because
 the request is not originating from allowed source types.
```

Quais métricas o Firehose emite para o recurso de descompressão?

O Firehose emite métricas para descompressão de cada registro. Você deve selecionar o período (1 min), a estatística (soma), o intervalo de datas para obter o número de `DecompressedRecords` fracassados ou bem-sucedidos ou fracassados ou `DecompressedBytes` bem-sucedidos. Para ter mais informações, consulte CloudWatch Métricas de descompressão de registros.

Escrevendo para o Amazon Data Firehose usando eventos CloudWatch

Você pode configurar CloudWatch a Amazon para enviar eventos para um stream do Firehose adicionando um destino a uma regra de CloudWatch eventos.

Para criar um destino para uma regra de CloudWatch eventos que envia eventos para um stream existente do Firehose

1. Faça login no AWS Management Console e abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Escolha Criar Regra.
3. Na página Etapa 1: Criar regra, em Targets, escolha Add target e, em seguida, escolha Firehose stream.
4. Escolha um stream existente do Firehose.

Para obter mais informações sobre a criação de regras de CloudWatch eventos, consulte Getting Started with Amazon CloudWatch Events.

Escrevendo para o Amazon Data Firehose usando AWS IoT

Você pode configurar AWS IoT para enviar informações para um stream do Firehose adicionando uma ação.

Para criar uma ação que envie eventos para um stream existente do Firehose

1. Ao criar uma regra do console do AWS IoT na página Create a rule (Criar uma regra), em Set one or more actions (Definir uma ou mais ações), selecione Add action (Adicionar ação).
2. Escolha Enviar mensagens para um fluxo do Amazon Kinesis Firehose.
3. Escolha Configurar ação.
4. Em Nome do fluxo, escolha um stream existente do Firehose.
5. Em Separator, escolha um caractere separador a ser inserido entre os registros.
6. Em Nome do perfil do IAM, escolha um perfil do IAM ou escolha Criar um novo perfil.
7. Selecione Adicionar ação.

Para obter mais informações sobre a criação de regras da AWS IoT, consulte AWS IoT Rule Tutorials.

Segurança no Amazon Data Firehose

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

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isto como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. A eficácia da nossa segurança é regularmente testada e verificada por auditores de terceiros como parte dos Programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam ao Data Firehose, consulte AWS Services in Scope by Compliance Program.
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, como a confidencialidade de seus dados, os requisitos da sua organização, leis e regulamentos aplicáveis.

Essa documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Data Firehose. Os tópicos a seguir mostram como configurar o Data Firehose para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que podem ajudá-lo a monitorar e proteger seus recursos do Data Firehose.

Tópicos

Proteção de dados no Amazon Data Firehose

O Amazon Data Firehose criptografa todos os dados em trânsito usando o protocolo TLS. Além disso, para dados armazenados em armazenamento provisório durante o processamento, o Amazon Data Firehose criptografa os dados AWS Key Management Serviceusando e verifica a integridade dos dados usando a verificação de soma de verificação.

Se você tiver dados confidenciais, poderá ativar a criptografia de dados do lado do servidor ao usar o Amazon Data Firehose. Como fazer isso depende da fonte dos seus dados.

> **Note**
>
> 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 disponíveis, consulte Federal Information Processing Standard (FIPS) 140-2.

Criptografia no lado do servidor tendo o Kinesis Data Streams como fonte de dados

Quando você envia dados de seus produtores de dados para seu stream de dados, o Kinesis Data Streams criptografa seus dados AWS Key Management Service usando AWS KMS uma chave () antes de armazená-los em repouso. Quando seu stream do Firehose lê os dados do seu stream de dados, o Kinesis Data Streams primeiro descriptografa os dados e depois os envia para o Amazon Data Firehose. O Amazon Data Firehose armazena os dados na memória com base nas dicas de buffer que você especificar. Em seguida, entrega-o aos destinos sem armazenar os dados não criptografados em repouso.

Para obter informações sobre como habilitar a criptografia no lado do servidor para o Kinesis Data Streams, consulte Using Server-Side Encryption no Amazon Kinesis Data Streams Developer Guide.

Criptografia do lado do servidor com Direct PUT ou outras fontes de dados

Se você enviar dados para seu stream do Firehose usando PutRecordou PutRecordBatch, ou se você enviar os dados usando AWS IoT Amazon CloudWatch Logs ou CloudWatch Events, você pode ativar a criptografia do lado do servidor usando a operação. StartDeliveryStreamEncryption

Para parar server-side-encryption, use a StopDeliveryStreamEncryptionoperação.

Você também pode ativar o SSE ao criar o stream do Firehose. Para fazer isso, especifique DeliveryStreamEncryptionConfigurationInputquando você invoca. CreateDeliveryStream

Quando a CMK é do tipoCUSTOMER_MANAGED_CMK, se o serviço Amazon Data Firehose não conseguir descriptografar registros por causa de KMSNotFoundException a, KMSInvalidStateException a, a ou KMSAccessDeniedException a, KMSDisabledException o serviço espera até 24 horas (o período de retenção) para que você resolva o problema. Se o problema persistir depois do período de retenção, o serviço ignorará os registros que passaram pelo período de retenção e não puderam ser descriptografados, e descartará os dados. O Amazon Data Firehose fornece as quatro CloudWatch métricas a seguir que você pode usar para rastrear as quatro AWS KMS exceções:

- KMSKeyAccessDenied
- KMSKeyDisabled
- KMSKeyInvalidState
- KMSKeyNotFound

Para obter mais informações sobre essas quatro métricas, consulte the section called "Monitoramento com CloudWatch métricas".

Important

Para criptografar seu stream do Firehose, use CMKs simétricas. O Amazon Data Firehose não oferece suporte a CMKs assimétricas. Para obter informações sobre CMKs simétricas e assimétricas, consulte Sobre CMKs simétricas e assimétricas no guia do desenvolvedor. AWS Key Management Service

Note

Quando você usa uma chave gerenciada pelo cliente (CUSTOMER_MANAGED_CMK) para ativar a criptografia do lado do servidor (SSE) para seu stream do Firehose, o serviço Firehose define um contexto de criptografia sempre que usa sua chave. Como esse contexto de criptografia representa uma ocorrência em que uma chave pertencente à sua AWS conta foi usada, ela é registrada como parte dos registros de AWS CloudTrail eventos da sua

AWS conta. Esse contexto de criptografia é gerado pelo sistema pelo serviço Firehose. Seu aplicativo não deve fazer nenhuma suposição sobre o formato ou o conteúdo do contexto de criptografia definido pelo serviço Firehose.

Controle de acesso com o Amazon Data Firehose

As seções a seguir abordam como controlar o acesso de e para seus recursos do Amazon Data Firehose. As informações que eles abordam incluem como conceder acesso ao seu aplicativo para que ele possa enviar dados para o stream do Firehose. Eles também descrevem como você pode conceder ao Amazon Data Firehose acesso ao seu bucket do Amazon Simple Storage Service (Amazon S3), ao cluster do Amazon Redshift ou ao cluster do OpenSearch Amazon Service, bem como às permissões de acesso necessárias se você usar Datadog, Dynatrace, MongoDB, New Relic, Splunk ou Sumo LogicMonitor Logic como seu destino. Por fim, você encontrará neste tópico orientações sobre como configurar o Amazon Data Firehose para que ele possa entregar dados a um destino que pertença a uma conta diferente AWS . A tecnologia para gerenciar todas essas formas de acesso é AWS Identity and Access Management (IAM). Para obter mais informações sobre o IAM, consulte O que é o IAM?.

Conteúdo

- Conceda ao Amazon Data Firehose acesso a um destino Snowflake
- Acesso ao Snowflake em VPC
- Conceda ao Amazon Data Firehose acesso a um destino de endpoint HTTP
- Entrega entre contas da Amazon MSK
- Entrega entre contas a um destino do Amazon S3
- Entrega entre contas para um destino OpenSearch de serviço
- Uso de tags para controle de acesso

Conceda ao seu aplicativo acesso aos recursos do Amazon Data Firehose

Para dar ao seu aplicativo acesso ao stream do Firehose, use uma política semelhante a este exemplo. Você pode ajustar as operações de API individuais às quais concede acesso modificando a seção `Action` ou conceder acesso a todas as operações com `"firehose:*"`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "firehose:DeleteDeliveryStream",
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:UpdateDestination"
            ],
            "Resource": [
                "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"
            ]
        }
    ]
}
```

Conceda ao Amazon Data Firehose acesso ao seu cluster privado do Amazon MSK

Se a origem do seu stream do Firehose for um cluster privado do Amazon MSK, use uma política semelhante a este exemplo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "Service": [
          "firehose.amazonaws.com"
        ]
    },
    "Effect": "Allow",
    "Action": [
      "kafka:CreateVpcConnection"
    ],
    "Resource": "cluster-arn"
    }
  ]
}
```

Permita que o Amazon Data Firehose assuma uma função do IAM

Esta seção descreve as permissões e políticas que concedem ao Amazon Data Firehose acesso para ingerir, processar e entregar dados da origem ao destino.

> **Note**
>
> Se você usar o console para criar um stream do Firehose e escolher a opção de criar uma nova função, AWS anexará a política de confiança necessária à função. Se você quiser que o Amazon Data Firehose use uma função do IAM existente ou crie uma função por conta própria, anexe as seguintes políticas de confiança a essa função para que o Amazon Data Firehose possa assumi-la. Edite as políticas para substituir o ID da *conta pelo ID* da sua AWS conta. Para obter informações sobre como modificar a relação de confiança de uma função, consulte Modificar uma função.

O Amazon Data Firehose usa uma função do IAM para todas as permissões que o stream do Firehose precisa para processar e entregar dados. Certifique-se de que as seguintes políticas de confiança estejam associadas a essa função para que o Amazon Data Firehose possa assumi-la.

```
{
 "Version": "2012-10-17",
```

```
 "Statement": [{
  "Sid": "",
  "Effect": "Allow",
  "Principal": {
   "Service": "firehose.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
   "StringEquals": {
    "sts:ExternalId": "account-id"
   }
  }
 }]
}
```

Essa política usa a chave de contexto de `sts:ExternalId` condição para garantir que somente as atividades do Amazon Data Firehose originadas da sua AWS conta possam assumir essa função do IAM. Para obter mais informações, consulte O problema de "confused deputy" no Guia do usuário do IAM.

Se você escolher o Amazon MSK como fonte para seu stream do Firehose, deverá especificar outra função do IAM que conceda ao Amazon Data Firehose permissões para ingerir dados de origem do cluster Amazon MSK especificado. Certifique-se de que as seguintes políticas de confiança estejam associadas a essa função para que o Amazon Data Firehose possa assumi-la.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "Service": [
          "firehose.amazonaws.com"
        ]
      },
      "Effect": "Allow",
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Certifique-se de que essa função que concede ao Amazon Data Firehose permissões para ingerir dados de origem do cluster Amazon MSK especificado conceda as seguintes permissões:

```
{
   "Version": "2012-10-17",
   "Statement": [{
        "Effect":"Allow",
        "Action": [
           "kafka:GetBootstrapBrokers",
           "kafka:DescribeCluster",
           "kafka:DescribeClusterV2",
           "kafka-cluster:Connect"
         ],
         "Resource": "CLUSTER-ARN"
       },
       {
         "Effect":"Allow",
         "Action": [
           "kafka-cluster:DescribeTopic",
           "kafka-cluster:DescribeTopicDynamicConfiguration",
           "kafka-cluster:ReadData"
         ],
         "Resource": "TOPIC-ARN"
       }]
}
```

Conceda acesso ao Amazon Data Firehose AWS Glue para conversão de formato de dados

Se seu stream do Firehose realizar a conversão do formato de dados, o Amazon Data Firehose fará referência às definições de tabela armazenadas em. AWS Glue Para dar ao Amazon Data Firehose o acesso necessário AWS Glue, adicione a seguinte declaração à sua política. Para obter informações sobre como encontrar o ARN da tabela, consulte Especificando ARNs de recursos do AWS Glue.

```
[{
    "Effect": "Allow",
    "Action": [
        "glue:GetTable",
        "glue:GetTableVersion",
```

```
        "glue:GetTableVersions"
    ],
    "Resource": "table-arn"
}, {
    "Sid": "GetSchemaVersion",
    "Effect": "Allow",
    "Action": [
        "glue:GetSchemaVersion"
    ],
    "Resource": ["*"]
}]
```

A política recomendada para obter esquemas do registro de esquemas não tem restrições de recursos. Para obter mais informações, consulte exemplos de IAM para desserializadores no Guia do AWS Glue desenvolvedor.

> **Note**
>
> Atualmente, não AWS Glue é suportado nas regiões de Israel (Tel Aviv), Ásia-Pacífico (Jacarta) ou Oriente Médio (EAU). Se você estiver trabalhando com o Amazon Data Firehose na região Ásia-Pacífico (Jacarta) ou na região do Oriente Médio (EAU), certifique-se de dar acesso ao Amazon Data Firehose AWS Glue em uma das regiões onde há suporte no momento. AWS Glue Há suporte para interoperabilidade entre regiões entre Data Firehose e. AWS Glue Para obter mais informações sobre as regiões em AWS Glue que há suporte, consulte https://docs.aws.amazon.com/general/latest/gr/glue.html

Conceda ao Amazon Data Firehose acesso a um destino do Amazon S3

Quando você está usando um destino do Amazon S3, o Amazon Data Firehose entrega dados para seu bucket do S3 e, opcionalmente, pode usar uma AWS KMS chave que você possui para criptografia de dados. Se o registro de erros estiver ativado, o Amazon Data Firehose também enviará erros de entrega de dados para seu grupo de CloudWatch registros e streams. Você precisa ter uma função do IAM ao criar um stream do Firehose. O Amazon Data Firehose assume essa função do IAM e obtém acesso ao bucket, à chave e ao grupo de CloudWatch registros e fluxos especificados.

Use a seguinte política de acesso para permitir que o Amazon Data Firehose acesse seu bucket e sua chave do S3. AWS KMS Se você não tiver o bucket do S3, adicione `s3:PutObjectAcl` à lista

de ações do Amazon S3. Isso concede ao proprietário do bucket acesso total aos objetos entregues pelo Amazon Data Firehose.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStream",
                "kinesis:GetShardIterator",
                "kinesis:GetRecords",
                "kinesis:ListShards"
            ],
            "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
        },
        {
           "Effect": "Allow",
           "Action": [
               "kms:Decrypt",
               "kms:GenerateDataKey"
           ],
           "Resource": [
               "arn:aws:kms:region:account-id:key/key-id"
           ],
           "Condition": {
               "StringEquals": {
```

```
                    "kms:ViaService": "s3.region.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:log-
stream-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": [
                "arn:aws:lambda:region:account-id:function:function-name:function-
version"
            ]
        }
    ]
}
```

A política acima também tem uma declaração que permite o acesso ao Amazon Kinesis Data Streams. Se você não usar o Kinesis Data Streams como fonte de dados, poderá remover essa instrução. Se você usar o Amazon MSK como sua fonte, poderá substituir essa declaração pela seguinte:

```
{
   "Sid":"",
   "Effect":"Allow",
   "Action":[
      "kafka:GetBootstrapBrokers",
      "kafka:DescribeCluster",
```

```
      "kafka:DescribeClusterV2",
      "kafka-cluster:Connect"
   ],
   "Resource":"arn:aws:kafka:{{mskClusterRegion}}:{{mskClusterAccount}}:cluster/
{{mskClusterName}}/{{clusterUUID}}"
},
{
   "Sid":"",
   "Effect":"Allow",
   "Action":[
      "kafka-cluster:DescribeTopic",
      "kafka-cluster:DescribeTopicDynamicConfiguration",
      "kafka-cluster:ReadData"
   ],
   "Resource":"arn:aws:kafka:{{mskClusterRegion}}:{{mskClusterAccount}}:topic/
{{mskClusterName}}/{{clusterUUID}}/{{mskTopicName}}"
},
{
   "Sid":"",
   "Effect":"Allow",
   "Action":[
      "kafka-cluster:DescribeGroup"
   ],
   "Resource":"arn:aws:kafka:{{mskClusterRegion}}:{{mskClusterAccount}}:group/
{{mskClusterName}}/{{clusterUUID}}/*"
}
```

Para obter mais informações sobre como permitir que outros AWS serviços acessem seus AWS recursos, consulte Como criar uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

Para saber como conceder ao Amazon Data Firehose acesso a um destino do Amazon S3 em outra conta, consulte. the section called “Entrega entre contas a um destino do Amazon S3”

Conceda ao Amazon Data Firehose acesso a um destino do Amazon Redshift

Consulte o seguinte ao conceder acesso ao Amazon Data Firehose ao usar um destino do Amazon Redshift.

Tópicos

- Perfil do IAM e políticas de acesso padrão

- Acesso da VPC a um cluster provisionado pelo Amazon Redshift ou a um grupo de trabalho do Amazon Redshift sem servidor

Perfil do IAM e políticas de acesso padrão

Quando você está usando um destino do Amazon Redshift, o Amazon Data Firehose entrega dados para seu bucket do S3 como um local intermediário. Opcionalmente, ele pode usar qualquer AWS KMS chave que você possua para criptografia de dados. Em seguida, o Amazon Data Firehose carrega os dados do bucket S3 para seu cluster provisionado do Amazon Redshift ou grupo de trabalho Amazon Redshift Serverless. Se o registro de erros estiver ativado, o Amazon Data Firehose também enviará erros de entrega de dados para seu grupo de CloudWatch registros e streams. O Amazon Data Firehose usa o nome de usuário e a senha especificados do Amazon Redshift para acessar seu cluster provisionado ou grupo de trabalho Amazon Redshift Serverless e usa uma função do IAM para acessar o bucket, a chave, o grupo de logs e os streams especificados. CloudWatch Você precisa ter uma função do IAM ao criar um stream do Firehose.

Use a seguinte política de acesso para permitir que o Amazon Data Firehose acesse seu bucket e sua chave do S3. AWS KMS Se você não possui o bucket do S3, adicione-o `s3:PutObjectAcl` à lista de ações do Amazon S3, que concede ao proprietário do bucket acesso total aos objetos entregues pelo Amazon Data Firehose. A política acima também tem uma instrução que concede acesso ao Amazon Kinesis Data Streams. Se você não usar o Kinesis Data Streams como fonte de dados, poderá remover essa instrução.

```
{
"Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": [
                "arn:aws:kms:region:account-id:key/key-id"
            ],
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "s3.region.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStream",
                "kinesis:GetShardIterator",
                "kinesis:GetRecords",
                "kinesis:ListShards"
            ],
            "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:log-
stream-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "lambda:InvokeFunction",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": [
                "arn:aws:lambda:region:account-id:function:function-name:function-
version"
            ]
        }
    ]
}
```

Para obter mais informações sobre como permitir que outros AWS serviços acessem seus AWS recursos, consulte Como criar uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

Acesso da VPC a um cluster provisionado pelo Amazon Redshift ou a um grupo de trabalho do Amazon Redshift sem servidor

Se o cluster provisionado do Amazon Redshift ou um grupo de trabalho do Amazon Redshift sem servidor estiver em uma nuvem privada virtual (VPC), ele deve ser acessível publicamente com um endereço IP público. Além disso, conceda ao Amazon Data Firehose acesso ao seu cluster provisionado do Amazon Redshift ou ao grupo de trabalho Amazon Redshift Serverless desbloqueando os endereços IP do Amazon Data Firehose. Atualmente, o Amazon Data Firehose usa um bloco CIDR para cada região disponível:

- `13.58.135.96/27` para Leste dos EUA (Ohio)
- `52.70.63.192/27` para Leste dos EUA (Norte da Virgínia)
- `13.57.135.192/27` para Oeste dos EUA (N. da Califórnia)
- `52.89.255.224/27` para Oeste dos EUA (Oregon)
- `18.253.138.96/27`para AWS GovCloud (Leste dos EUA)
- `52.61.204.160/27`para AWS GovCloud (Oeste dos EUA)
- `35.183.92.128/27` para Canadá (Central)
- `40.176.98.192/27`por Canada West (Calgary)
- `18.162.221.32/27` para Ásia-Pacífico (Hong Kong)
- `13.232.67.32/27` para Ásia-Pacífico (Mumbai)
- `18.60.192.128/27` para Ásia-Pacífico (Hyderabad)

- 13.209.1.64/27 para Ásia-Pacífico (Seul)
- 13.228.64.192/27 para Ásia-Pacífico (Singapura)
- 13.210.67.224/27 para Ásia-Pacífico (Sydney)
- 108.136.221.64/27 para Ásia-Pacífico (Jacarta)
- 13.113.196.224/27 para Ásia-Pacífico (Tóquio)
- 13.208.177.192/27 para Ásia-Pacífico (Osaka)
- 52.81.151.32/27 para China (Pequim)
- 161.189.23.64/27 para China (Ningxia)
- 16.62.183.32/27 para Europa (Zurique)
- 35.158.127.160/27 para Europa (Frankfurt)
- 52.19.239.192/27 para Europa (Irlanda)
- 18.130.1.96/27 para Europa (Londres)
- 35.180.1.96/27 para Europa (Paris)
- 13.53.63.224/27 para Europa (Estocolmo)
- 15.185.91.0/27 para Oriente Médio (Bahrein)
- 18.228.1.128/27 para América do Sul (São Paulo)
- 15.161.135.128/27 para Europa (Milão)
- 13.244.121.224/27 para África (Cidade do Cabo)
- 3.28.159.32/27 para Oriente Médio (Emirados Árabes Unidos)
- 51.16.102.0/27 para Israel (Tel Aviv)
- 16.50.161.128/27 para Ásia-Pacífico (Melbourne)

Para obter mais informações sobre como desbloquear endereços IP, consulte a etapa Autorizar o acesso ao cluster no Guia de conceitos básicos do Amazon Redshift.

Conceda ao Amazon Data Firehose acesso a um destino de serviço público OpenSearch

Quando você está usando um destino de OpenSearch serviço, o Amazon Data Firehose entrega dados para seu cluster de OpenSearch serviços e, simultaneamente, faz backup de todos os documentos falhados ou de todos os documentos em seu bucket do S3. Se o registro de erros

estiver ativado, o Amazon Data Firehose também enviará erros de entrega de dados para seu grupo de CloudWatch registros e streams. O Amazon Data Firehose usa uma função do IAM para acessar o domínio de OpenSearch serviço, o bucket do S3, a AWS KMS chave, o grupo de CloudWatch registros e os fluxos especificados. Você precisa ter uma função do IAM ao criar um stream do Firehose.

Use a seguinte política de acesso para permitir que o Amazon Data Firehose acesse seu bucket, domínio de OpenSearch serviço e chave do S3. AWS KMS Se você não possui o bucket do S3, adicione-o `s3:PutObjectAcl` à lista de ações do Amazon S3, que concede ao proprietário do bucket acesso total aos objetos entregues pelo Amazon Data Firehose. A política acima também tem uma instrução que concede acesso ao Amazon Kinesis Data Streams. Se você não usar o Kinesis Data Streams como fonte de dados, poderá remover essa instrução.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
           "Effect": "Allow",
           "Action": [
               "kms:Decrypt",
               "kms:GenerateDataKey"
           ],
           "Resource": [
               "arn:aws:kms:region:account-id:key/key-id"
           ],
           "Condition": {
               "StringEquals": {
```

```
                    "kms:ViaService": "s3.region.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
                }
            }
        },
        {
           "Effect": "Allow",
           "Action": [
               "es:DescribeDomain",
               "es:DescribeDomains",
               "es:DescribeDomainConfig",
               "es:ESHttpPost",
               "es:ESHttpPut"
           ],
          "Resource": [
              "arn:aws:es:region:account-id:domain/domain-name",
              "arn:aws:es:region:account-id:domain/domain-name/*"
          ]
       },
       {
          "Effect": "Allow",
          "Action": [
              "es:ESHttpGet"
          ],
          "Resource": [
              "arn:aws:es:region:account-id:domain/domain-name/_all/_settings",
              "arn:aws:es:region:account-id:domain/domain-name/_cluster/stats",
              "arn:aws:es:region:account-id:domain/domain-name/index-name*/
_mapping/type-name",
              "arn:aws:es:region:account-id:domain/domain-name/_nodes",
              "arn:aws:es:region:account-id:domain/domain-name/_nodes/stats",
              "arn:aws:es:region:account-id:domain/domain-name/_nodes/*/stats",
              "arn:aws:es:region:account-id:domain/domain-name/_stats",
              "arn:aws:es:region:account-id:domain/domain-name/index-name*/_stats",
              "arn:aws:es:region:account-id:domain/domain-name/"
          ]
       },
       {
          "Effect": "Allow",
          "Action": [
              "kinesis:DescribeStream",
```

```
                "kinesis:GetShardIterator",
                "kinesis:GetRecords",
                "kinesis:ListShards"
            ],
            "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:log-
stream-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": [
                "arn:aws:lambda:region:account-id:function:function-name:function-
version"
            ]
        }
    ]
}
```

Para obter mais informações sobre como permitir que outros AWS serviços acessem seus AWS recursos, consulte Como criar uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

Para saber como conceder ao Amazon Data Firehose acesso a um cluster de OpenSearch serviços em outra conta, consulte. the section called “Entrega entre contas para um destino OpenSearch de serviço”

Conceda ao Amazon Data Firehose acesso a um destino de OpenSearch serviço em uma VPC

Se o seu domínio de OpenSearch serviço estiver em uma VPC, certifique-se de conceder ao Amazon Data Firehose as permissões descritas na seção anterior. Além disso, você precisa conceder ao Amazon Data Firehose as seguintes permissões para permitir que ele acesse a VPC do seu domínio OpenSearch de serviço.

- `ec2:DescribeVpcs`
- `ec2:DescribeVpcAttribute`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeNetworkInterfaces`
- `ec2:CreateNetworkInterface`
- `ec2:CreateNetworkInterfacePermission`
- `ec2:DeleteNetworkInterface`

> **Important**
>
> Não revogue essas permissões depois de criar o stream do Firehose. Se você revogar essas permissões, seu stream do Firehose será degradado ou deixará de fornecer dados ao OpenSearch seu domínio de serviço sempre que o serviço tentar consultar ou atualizar ENIs.

> **Important**
>
> Ao especificar sub-redes para entregar dados ao destino em uma VPC privada, verifique se você tem um número suficiente de endereços IP livres nas sub-redes escolhidas. Se não houver um endereço IP gratuito disponível em uma sub-rede especificada, o Firehose não poderá criar ou adicionar ENIs para a entrega de dados na VPC privada, e a entrega será degradada ou falhará.

Ao criar ou atualizar seu stream do Firehose, você especifica um grupo de segurança para o Firehose usar ao enviar dados para seu domínio de serviço. OpenSearch Você pode usar o

mesmo grupo de segurança usado pelo domínio do OpenSearch Serviço ou um diferente. Se você especificar um grupo de segurança diferente, certifique-se de que ele permita tráfego HTTPS de saída para o grupo de segurança do domínio do OpenSearch Serviço. Além disso, certifique-se de que o grupo de segurança do domínio OpenSearch Service permita tráfego HTTPS do grupo de segurança que você especificou ao configurar seu stream do Firehose. Se você usa o mesmo grupo de segurança para o stream do Firehose e para o domínio OpenSearch Service, verifique se a regra de entrada do grupo de segurança permite tráfego HTTPS. Para obter mais informações sobre regras de grupo de segurança, consulte Regras de grupo de segurança na documentação da Amazon VPC.

Conceda ao Amazon Data Firehose acesso a um destino público OpenSearch sem servidor

Quando você está usando um destino OpenSearch sem servidor, o Amazon Data Firehose entrega dados para sua coleção OpenSearch sem servidor e, ao mesmo tempo, faz backup de todos os documentos falhados ou de todos os documentos em seu bucket do S3. Se o registro de erros estiver ativado, o Amazon Data Firehose também enviará erros de entrega de dados para seu grupo de CloudWatch registros e streams. O Amazon Data Firehose usa uma função do IAM para acessar a coleção OpenSearch Serverless, o bucket S3, o grupo e os fluxos de AWS KMS chaves e CloudWatch logs especificados. Você precisa ter uma função do IAM ao criar um stream do Firehose.

Use a seguinte política de acesso para permitir que o Amazon Data Firehose acesse seu bucket S3, domínio OpenSearch sem servidor e chave. AWS KMS Se você não possui o bucket do S3, adicione-o `s3:PutObjectAcl` à lista de ações do Amazon S3, que concede ao proprietário do bucket acesso total aos objetos entregues pelo Amazon Data Firehose. A política acima também tem uma instrução que concede acesso ao Amazon Kinesis Data Streams. Se você não usar o Kinesis Data Streams como fonte de dados, poderá remover essa instrução.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
```

```
                "s3:ListBucketMultipartUploads",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
           "Effect": "Allow",
           "Action": [
               "kms:Decrypt",
               "kms:GenerateDataKey"
           ],
           "Resource": [
               "arn:aws:kms:region:account-id:key/key-id"
           ],
           "Condition": {
               "StringEquals": {
                   "kms:ViaService": "s3.region.amazonaws.com"
               },
               "StringLike": {
                   "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
               }
           }
        },
       {
          "Effect": "Allow",
          "Action": [
              "kinesis:DescribeStream",
              "kinesis:GetShardIterator",
              "kinesis:GetRecords",
              "kinesis:ListShards"
          ],
          "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
       },
       {
          "Effect": "Allow",
          "Action": [
              "logs:PutLogEvents"
          ],
          "Resource": [
```

```
            "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:log-
stream-name"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "lambda:InvokeFunction",
            "lambda:GetFunctionConfiguration"
         ],
         "Resource": [
            "arn:aws:lambda:region:account-id:function:function-name:function-
version"
         ]
      },
      {
       "Effect": "Allow",
       "Action": "aoss:APIAccessAll",
       "Resource": "arn:aws:aoss:region:account-id:collection/collection-id"
     }
   ]
}
```

Além da política acima, você também deve configurar o Amazon Data Firehose para ter as seguintes permissões mínimas atribuídas em uma política de acesso a dados:

```
[
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/target-collection/target-index"
            ],
            "Permission":[
               "aoss:WriteDocument",
               "aoss:UpdateIndex",
               "aoss:CreateIndex"
            ]
         }
      ],
      "Principal":[
```

```
            "arn:aws:sts::account-id:assumed-role/firehose-delivery-role-name/*"
        ]
    }
]
```

Para obter mais informações sobre como permitir que outros AWS serviços acessem seus AWS recursos, consulte Como criar uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

Conceda ao Amazon Data Firehose acesso a um destino OpenSearch sem servidor em uma VPC

Se sua coleção OpenSearch Serverless estiver em uma VPC, certifique-se de conceder ao Amazon Data Firehose as permissões descritas na seção anterior. Além disso, você precisa conceder ao Amazon Data Firehose as seguintes permissões para permitir que ele acesse a VPC da sua OpenSearch coleção Serverless.

- `ec2:DescribeVpcs`
- `ec2:DescribeVpcAttribute`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeNetworkInterfaces`
- `ec2:CreateNetworkInterface`
- `ec2:CreateNetworkInterfacePermission`
- `ec2:DeleteNetworkInterface`

> **Important**
>
> Não revogue essas permissões depois de criar o stream do Firehose. Se você revogar essas permissões, seu stream do Firehose será degradado ou deixará de fornecer dados ao OpenSearch seu domínio de serviço sempre que o serviço tentar consultar ou atualizar ENIs.

> **Important**
>
> Ao especificar sub-redes para entregar dados ao destino em uma VPC privada, verifique se você tem um número suficiente de endereços IP livres nas sub-redes escolhidas. Se não houver um endereço IP gratuito disponível em uma sub-rede especificada, o Firehose não poderá criar ou adicionar ENIs para a entrega de dados na VPC privada, e a entrega será degradada ou falhará.

Ao criar ou atualizar seu stream do Firehose, você especifica um grupo de segurança para o Firehose usar ao enviar dados para sua coleção Serverless. OpenSearch Você pode usar o mesmo grupo de segurança que a coleção OpenSearch Serverless usa ou um diferente. Se você especificar um grupo de segurança diferente, certifique-se de que ele permita tráfego HTTPS de saída para o grupo de segurança da coleção OpenSearch Serverless. Além disso, certifique-se de que o grupo de segurança da coleção OpenSearch Serverless permita tráfego HTTPS do grupo de segurança que você especificou ao configurar seu stream do Firehose. Se você usa o mesmo grupo de segurança para o stream do Firehose e para a coleção OpenSearch Serverless, verifique se a regra de entrada do grupo de segurança permite tráfego HTTPS. Para obter mais informações sobre regras de grupo de segurança, consulte Regras de grupo de segurança na documentação da Amazon VPC.

Conceda ao Amazon Data Firehose acesso a um destino Splunk

Quando você usa um destino Splunk, o Amazon Data Firehose entrega dados para seu endpoint do Splunk HTTP Event Collector (HEC). Ele também faz backup desses dados no bucket do Amazon S3 que você especificar e, opcionalmente, você pode usar uma AWS KMS chave que você possui para a criptografia do lado do servidor do Amazon S3. Se o registro de erros estiver ativado, o Firehose enviará erros de entrega de dados para seus fluxos de CloudWatch registro. Você também pode usar AWS Lambda para transformação de dados.

Se você usa um balanceador de AWS carga, certifique-se de que seja um Classic Load Balancer ou um Application Load Balancer. Além disso, habilite sessões fixas com base na duração com a expiração de cookies desativada para o Classic Load Balancer e a expiração é definida como máxima (7 dias) para o Application Load Balancer. Para obter informações sobre como fazer isso, consulte Duration-Based Session Stickiness for Classic Load Balancer ou an Application Load Balancer.

Você deve ter uma função do IAM ao criar um stream do Firehose. O Firehose assume essa função do IAM e obtém acesso ao bucket, à chave, ao grupo de CloudWatch registros e aos fluxos especificados.

Use a seguinte política de acesso para permitir que o Amazon Data Firehose acesse seu bucket do S3. Se você não possui o bucket do S3, adicione-o `s3:PutObjectAcl` à lista de ações do Amazon S3, que concede ao proprietário do bucket acesso total aos objetos entregues pelo Amazon Data Firehose. Essa política também concede ao Amazon Data Firehose acesso CloudWatch para registro de erros e transformação de AWS Lambda dados. A política também tem uma instrução que permite o acesso ao Amazon Kinesis Data Streams. Se você não usar o Kinesis Data Streams como fonte de dados, poderá remover essa instrução. O Amazon Data Firehose não usa o IAM para acessar o Splunk. Para acessar o Splunk, ele usa o token HEC.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
           "Effect": "Allow",
           "Action": [
               "kms:Decrypt",
               "kms:GenerateDataKey"
           ],
           "Resource": [
               "arn:aws:kms:region:account-id:key/key-id"
           ],
           "Condition": {
```

```
                "StringEquals": {
                    "kms:ViaService": "s3.region.amazonaws.com"
                },
                "StringLike": {
                    "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:DescribeStream",
                "kinesis:GetShardIterator",
                "kinesis:GetRecords",
                "kinesis:ListShards"
            ],
            "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": [
                "arn:aws:lambda:region:account-id:function:function-name:function-
version"
            ]
        }
    ]
}
```

Para obter mais informações sobre como permitir que outros AWS serviços acessem seus AWS recursos, consulte Como criar uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

Acesso ao Splunk no VPC

Se a plataforma do Splunk estiver em uma VPC, ela será acessível ao público com um endereço IP público. Além disso, conceda ao Amazon Data Firehose acesso à sua plataforma Splunk desbloqueando os endereços IP do Amazon Data Firehose. Atualmente, o Amazon Data Firehose usa os seguintes blocos CIDR.

- `18.216.68.160/27, 18.216.170.64/27, 18.216.170.96/27` para Leste dos EUA (Ohio)
- `34.238.188.128/26, 34.238.188.192/26, 34.238.195.0/26` para Leste dos EUA (Norte da Virgínia)
- `13.57.180.0/26` para Oeste dos EUA (N. da Califórnia)
- `34.216.24.32/27, 34.216.24.192/27, 34.216.24.224/27` para Oeste dos EUA (Oregon)
- `18.253.138.192/26`para AWS GovCloud (Leste dos EUA)
- `52.61.204.192/26`para AWS GovCloud (Oeste dos EUA)
- `18.162.221.64/26` para Ásia-Pacífico (Hong Kong)
- `13.232.67.64/26` para Ásia-Pacífico (Mumbai)
- `13.209.71.0/26` para Ásia-Pacífico (Seul)
- `13.229.187.128/26` para Ásia-Pacífico (Singapura)
- `13.211.12.0/26` para Ásia-Pacífico (Sydney)
- `13.230.21.0/27, 13.230.21.32/27` para Ásia-Pacífico (Tóquio)
- `51.16.102.64/26` para Israel (Tel Aviv)
- `35.183.92.64/26` para Canadá (Central)
- `40.176.98.128/26`por Canada West (Calgary)
- `18.194.95.192/27, 18.194.95.224/27, 18.195.48.0/27` para Europa (Frankfurt)
- `34.241.197.32/27, 34.241.197.64/27, 34.241.197.96/27` para Europa (Irlanda)
- `18.130.91.0/26` para Europa (Londres)
- `35.180.112.0/26` para Europa (Paris)
- `13.53.191.0/26` para Europa (Estocolmo)

- 15.185.91.64/26 para Oriente Médio (Bahrein)
- 18.228.1.192/26 para América do Sul (São Paulo)
- 15.161.135.192/26 para Europa (Milão)
- 13.244.165.128/26 para África (Cidade do Cabo)
- 13.208.217.0/26 para Ásia-Pacífico (Osaka)
- 52.81.151.64/26 para China (Pequim)
- 161.189.23.128/26 para China (Ningxia)
- 108.136.221.128/26 para Ásia-Pacífico (Jacarta)
- 3.28.159.64/26 para Oriente Médio (Emirados Árabes Unidos)
- 51.16.102.64/26 para Israel (Tel Aviv)
- 16.62.183.64/26 para Europa (Zurique)
- 18.60.192.192/26 para Ásia-Pacífico (Hyderabad)
- 16.50.161.192/26 para Ásia-Pacífico (Melbourne)

Acesso ao Snowflake ou ao endpoint HTTP

Não há um subconjunto de intervalos de endereços AWS IP específicos para o Amazon Data Firehose quando o destino é um endpoint HTTP ou clusters públicos do Snowflake.

Para adicionar o Firehose a uma lista de permissões para clusters públicos do Snowflake ou aos seus endpoints públicos de HTTP ou HTTPS, adicione todos os intervalos de endereços AWS IP atuais às suas regras de entrada.

> **Note**
>
> As notificações nem sempre são provenientes de endereços IP na mesma AWS região do tópico associado. Você deve incluir o intervalo AWS de endereços IP para todas as regiões.

Conceda ao Amazon Data Firehose acesso a um destino Snowflake

Quando você usa o Snowflake como destino, o Firehose entrega dados para uma conta do Snowflake usando o URL da sua conta do Snowflake. Ele também faz backup dos dados de erro no bucket do Amazon Simple Storage Service que você especifica e, opcionalmente, você pode usar

uma AWS Key Management Service chave que você possui para a criptografia do lado do servidor do Amazon S3. Se o registro de erros estiver ativado, o Firehose enviará erros de entrega de dados para seus fluxos de CloudWatch registros.

Você precisa ter uma função do IAM antes de criar um stream do Firehose. O Firehose assume essa função do IAM e obtém acesso ao bucket, à chave, ao grupo e aos fluxos de CloudWatch registros especificados. Use a política de acesso a seguir para permitir que o Firehose acesse seu bucket do S3. Se você não é proprietário do bucket S3, adicione `s3:PutObjectAcl` à lista de ações do Amazon Simple Storage Service, que concede ao proprietário do bucket acesso total aos objetos entregues pelo Firehose. Essa política também concede ao Firehose acesso CloudWatch para registro de erros. A política também tem uma instrução que permite o acesso ao Amazon Kinesis Data Streams. Se você não usar o Kinesis Data Streams como fonte de dados, poderá remover essa instrução. O Firehose não usa o IAM para acessar o Snowflake. Para acessar o Snowflake, ele usa o URL e o ID de voz da sua conta do Snowflake no PrivateLink caso de um cluster privado.

```
{
"Version": "2012-10-17",
    "Statement":
    [
        {
"Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
"Effect": "Allow",
           "Action": [
               "kms:Decrypt",
               "kms:GenerateDataKey"
           ],
           "Resource": [
               "arn:aws:kms:region:account-id:key/key-id"
```

```
            ],
            "Condition": {
"StringEquals": {
"kms:ViaService": "s3.region.amazonaws.com"
                },
                "StringLike": {
"kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/prefix*"
                }
            }
        },
        {
"Effect": "Allow",
            "Action": [
                "kinesis:DescribeStream",
                "kinesis:GetShardIterator",
                "kinesis:GetRecords",
                "kinesis:ListShards"
            ],
            "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
        },
        {
"Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:*"
            ]
        }
    ]
}
```

Para obter mais informações sobre como permitir que outros AWS serviços acessem seus AWS recursos, consulte Como criar uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

Acesso ao Snowflake em VPC

Se o cluster do Snowflake tiver um link privado habilitado, o Firehose usará VPC Endpoints para entregar dados ao seu cluster privado sem passar pela Internet pública. Para isso, crie regras de rede do Snowflake para permitir a entrada do seguinte `AwsVpceIds` no cluster em que Região da

AWS seu cluster está. Para obter mais informações, consulte Criação de regras de rede no Guia do usuário do Snowflake.

IDs de endpoint de VPC a serem usados com base nas regiões em que seu cluster está

Região da AWS	**VPCE IDs**
Leste dos EUA (Ohio)	vpce-0d96cafcd96a50aeb vpce-0cec34343d48f537b
Leste dos EUA (Norte da Virgínia)	vpce-0b4d7e8478e141ba8 vpce-0b75cd681fb507352 vpce-01c03e63820ec00d8 vpce-0c2cfc51dc2882422 vpce-06ca862f019e4e056 vpce-020cda0cfa63f8d1c vpce-0b80504a1a783cd70 vpce-0289b9ff0b5259a96 vpce-0d7add8628bd69a12 vpce-02bfb5966cc59b2af vpce-09e707674af878bf2 vpce-049b52e96cc1a2165 vpce-0bb6c7b7a8a86cdbb vpce-03b22d599f51e80f3 vpce-01d60dc60fc106fe1 vpce-0186d20a4b24ecbef vpce-0533906401a36e416

Região da AWS	**VPCE IDs**
	vpce-05111fb13d396710e vpce-0694613f4fbd6f514 vpce-09b21cb25fe4cc4f4 vpce-06029c3550e4d2399 vpce-00961862a21b033da vpce-01620b9ae33273587 vpce-078cf4ec226880ac9 vpce-0d711bf076ce56381 vpce-066b7e13cbfca6f6e vpce-0674541252d9ccc26 vpce-03540b88dedb4b000 vpce-0b1828e79ad394b95 vpce-0dc0e6f001fb1a60d vpce-0d8f82e71a244098a vpce-00e374d9e3f1af5ce vpce-0c1e3d6631ddb442f

Região da AWS	**VPCE IDs**
Oeste dos EUA (Oregon)	vpce-0f60f72da4cd1e4e7 vpce-0c60d21eb8b1669fd vpce-01c4e3e29afdafbef vpce-0cc6bf2a88da139de vpce-0797e08e169e50662 vpce-033cbe480381b5c0e vpce-00debbdd8f9eb10a5 vpce-08ec2f386c809e889 vpce-0856d14310857b545
Europa (Frankfurt)	vpce-068dbb7d71c9460fb vpce-0a7a7f095942d4ec9
Europa (Irlanda)	vpce-06857e59c005a6276 vpce-04390f4f8778b75f2 vpce-011fd2b1f0aa172fd
Ásia-Pacífico (Tóquio)	vpce-06369e5258144e68a vpce-0f2363cdb8926fbe8
Ásia-Pacífico (Singapura)	vpce-049cd46cce7a12d52 vpce-0e8965a1a4bdb8941
Ásia-Pacífico (Seul)	vpce-0aa444d9001e1faa1 vpce-04a49d4dcfd02b884

Região da AWS	**VPCE IDs**
Ásia-Pacífico (Sydney)	vpce-048a60a182c52be63 vpce-03c19949787fd1859

Conceda ao Amazon Data Firehose acesso a um destino de endpoint HTTP

Você pode usar o Amazon Data Firehose para entregar dados para qualquer destino de endpoint HTTP. O Amazon Data Firehose também faz backup desses dados no bucket do Amazon S3 que você especificar e, opcionalmente, você pode usar AWS KMS uma chave que você possui para a criptografia do lado do servidor do Amazon S3. Se o registro de erros estiver ativado, o Amazon Data Firehose enviará erros de entrega de dados para seus fluxos de CloudWatch log. Você também pode usar AWS Lambda para transformação de dados.

Você precisa ter uma função do IAM ao criar um stream do Firehose. O Amazon Data Firehose assume essa função do IAM e obtém acesso ao bucket, à chave e ao grupo de CloudWatch registros e fluxos especificados.

Use a seguinte política de acesso para permitir que o Amazon Data Firehose acesse o bucket S3 que você especificou para backup de dados. Se você não possui o bucket do S3, adicione-o `s3:PutObjectAcl` à lista de ações do Amazon S3, que concede ao proprietário do bucket acesso total aos objetos entregues pelo Amazon Data Firehose. Essa política também concede ao Amazon Data Firehose acesso CloudWatch para registro de erros e transformação de AWS Lambda dados. A política também tem uma instrução que permite o acesso ao Amazon Kinesis Data Streams. Se você não usar o Kinesis Data Streams como fonte de dados, poderá remover essa instrução.

> **Important**
>
> O Amazon Data Firehose não usa o IAM para acessar destinos de endpoints HTTP pertencentes a provedores de serviços terceirizados compatíveis, incluindo Datadog, Dynatrace, LogicMonitor MongoDB, New Relic, Splunk ou Sumo Logic. Para acessar um destino de endpoint HTTP especificado de propriedade de um provedor de serviços terceirizado compatível, entre em contato com esse provedor de serviços para obter a chave de API ou a chave de acesso necessária para permitir a entrega de dados para esse serviço do Amazon Data Firehose.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
           "Effect": "Allow",
           "Action": [
               "kms:Decrypt",
               "kms:GenerateDataKey"
           ],
           "Resource": [
               "arn:aws:kms:region:account-id:key/key-id"
           ],
           "Condition": {
               "StringEquals": {
                   "kms:ViaService": "s3.region.amazonaws.com"
               },
               "StringLike": {
                   "kms:EncryptionContext:aws:s3:arn": "arn:aws:s3:::bucket-name/
prefix*"
               }
           }
        },
        {
           "Effect": "Allow",
           "Action": [
               "kinesis:DescribeStream",
               "kinesis:GetShardIterator",
```

```
                "kinesis:GetRecords",
                "kinesis:ListShards"
            ],
            "Resource": "arn:aws:kinesis:region:account-id:stream/stream-name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:log-group:log-group-name:log-stream:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": [
                "arn:aws:lambda:region:account-id:function:function-name:function-
version"
            ]
        }
    ]
}
```

Para obter mais informações sobre como permitir que outros AWS serviços acessem seus AWS recursos, consulte Como criar uma função para delegar permissões a um AWS serviço no Guia do usuário do IAM.

> **Important**
>
> Atualmente, o Amazon Data Firehose NÃO oferece suporte à entrega de dados para endpoints HTTP em uma VPC.

Entrega entre contas da Amazon MSK

Ao criar um stream do Firehose a partir da sua conta do Firehose (por exemplo, Conta B) e sua origem é um cluster MSK em outra AWS conta (Conta A), você deve ter as seguintes configurações em vigor.

Conta A:

1. No console do Amazon MSK, escolha o cluster provisionado e depois escolha Propriedades.
2. Em Configurações de rede, escolha Editar e ative a Conectividade de várias VPCs.
3. Em Configurações de segurança, escolha Editar política do cluster.

    a. Se o cluster ainda não tiver uma política configurada, marque Incluir entidade principal do serviço Firehose e Habilitar a entrega do S3 entre contas do Firehose. Isso AWS Management Console gerará automaticamente uma política com as permissões apropriadas.

    b. Se o cluster já tiver uma política configurada, adicione as seguintes permissões à política existente:

```
{
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::arn:role/mskaasTestDeliveryRole"
      },
      "Action": [
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2",
        "kafka-cluster:Connect"
      ],
      "Resource": "arn:aws:kafka:us-east-1:arn:cluster/DO-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20"  // ARN of the
 cluster
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::arn:role/mskaasTestDeliveryRole"
      },
      "Action": [
        "kafka-cluster:DescribeTopic",
```

```
        "kafka-cluster:DescribeTopicDynamicConfiguration",
        "kafka-cluster:ReadData"
      ],
      "Resource": "arn:aws:kafka:us-east-1:arn:topic/DO-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20/*"//topic of the
 cluster
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::233450236687:role/mskaasTestDeliveryRole"
      },
      "Action": "kafka-cluster:DescribeGroup",
      "Resource": "arn:aws:kafka:us-east-1:arn:group/DO-NOT-TOUCH-mskaas-
provisioned-privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20/*" //topic of
 the cluster
    },
 }
```

4. Em Entidade principal da AWS , insira o ID da entidade principal da Conta B.
5. Em Tópico, especifique o tópico do Apache Kafka do qual você deseja que seu stream do Firehose consuma dados. Depois que o stream do Firehose for criado, você não poderá atualizar esse tópico.
6. Selecione Save changes (Salvar alterações)

Conta B:

1. No console do Firehose, escolha Criar stream do Firehose usando a Conta B.
2. Em Fonte, escolha Amazon Managed Streaming for Apache.
3. Em Configurações da fonte, para o cluster do Amazon Managed Streaming for Apache Kafka, insira o ARN do cluster do Amazon MSK na Conta A.
4. Em Tópico, especifique o tópico do Apache Kafka do qual você deseja que seu stream do Firehose consuma dados. Depois que o stream do Firehose for criado, você não poderá atualizar esse tópico.
5. Em Nome do stream de entrega, especifique o nome do seu stream Firehose.

Na Conta B, ao criar seu stream do Firehose, você deve ter uma função do IAM (criada por padrão ao usar o AWS Management Console) que conceda ao stream do Firehose acesso de “leitura” ao cluster Amazon MSK entre contas para o tópico configurado.

Veja a seguir o que é configurado pelo AWS Management Console:

```
{
    "Sid": "",
    "Effect": "Allow",
    "Action": [
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2",
        "kafka-cluster:Connect"
        ],
    "Resource": "arn:aws:kafka:us-east-1:arn:cluster/DO-NOT-TOUCH-mskaas-provisioned-
privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20/*" //topic of the cluster
    },
    {
    "Sid": "",
    "Effect": "Allow",
    "Action": [
        "kafka-cluster:DescribeTopic",
        "kafka-cluster:DescribeTopicDynamicConfiguration",
        "kafka-cluster:ReadData"
    ],
    "Resource": "arn:aws:kafka:us-east-1:arn:topic/DO-NOT-TOUCH-mskaas-provisioned-
privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20/mskaas_test_topic" //topic of the
 cluster
    },
    {
    "Sid": "",
    "Effect": "Allow",
    "Action": [
        "kafka-cluster:DescribeGroup"
    ],
    "Resource": "arn:aws:kafka:us-east-1:arn:group/DO-NOT-TOUCH-mskaas-provisioned-
privateLink/xxxxxxxxx-2f3a-462a-ba09-xxxxxxxxxx-20/*" //topic of the cluster
    },
 }
```

Em seguida, você pode concluir a etapa opcional de configuração da transformação de registros e da conversão de formato de registros. Para ter mais informações, consulte Configurar a transformação de registros e a conversão de formatos.

Entrega entre contas a um destino do Amazon S3

Você pode usar as APIs AWS CLI ou as APIs do Amazon Data Firehose para criar um stream do Firehose em uma conta com AWS um destino do Amazon S3 em uma conta diferente. O procedimento a seguir mostra um exemplo de configuração de um stream do Firehose de propriedade da conta A para entregar dados a um bucket do Amazon S3 de propriedade da conta B.

1. Crie uma função do IAM na conta A usando as etapas descritas em Conceder acesso ao Firehose a um destino do Amazon S3.

   > **Note**
   >
   > O bucket do Amazon S3 especificado na política de acesso padrão pertence à conta B neste caso. Certifique-se de adicionar `s3:PutObjectAcl` à lista de ações do Amazon S3 na política de acesso, que concede à conta B acesso total aos objetos entregues pelo Amazon Data Firehose. Essa permissão é necessária para a entrega entre contas. O Amazon Data Firehose define o cabeçalho x-amz-acl "" na solicitação como ""bucket-owner-full-control.

2. Para permitir o acesso no perfil do IAM criado anteriormente, crie uma política de bucket do S3 na conta B. O código a seguir é um exemplo de política de bucket. Para obter mais informações, consulte Usar políticas de buckets e do usuário.

```
{

    "Version": "2012-10-17",
    "Id": "PolicyID",
    "Statement": [
        {
            "Sid": "StmtID",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::accountA-id:role/iam-role-name"
            },
            "Action": [
                "s3:AbortMultipartUpload",
```

```
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        }
    ]
}
```

3. Crie um stream do Firehose na conta A usando a função do IAM que você criou na etapa 1.

Entrega entre contas para um destino OpenSearch de serviço

Você pode usar as AWS CLI APIs do Amazon Data Firehose para criar um stream do Firehose em uma AWS conta com um destino de OpenSearch serviço em outra conta. O procedimento a seguir mostra um exemplo de como você pode criar um stream do Firehose na conta A e configurá-lo para entregar dados a um destino de OpenSearch serviço pertencente à conta B.

1. Criar um perfil do IAM na conta A usando as etapas descritas em the section called “Conceda ao Amazon Data Firehose acesso a um destino de serviço público OpenSearch ”.
2. Para permitir o acesso da função do IAM que você criou na etapa anterior, crie uma política de OpenSearch serviço na conta B. O seguinte JSON é um exemplo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::Account-A-ID:role/firehose_delivery_role "
      },
      "Action": "es:ESHttpGet",
      "Resource": [
```

```
        "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_all/
_settings",
        "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_cluster/
stats",
        "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/roletest*/
_mapping/roletest",
        "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_nodes",
        "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_nodes/
stats",
        "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_nodes/*/
stats",
        "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/_stats",
        "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/roletest*/
_stats",
        "arn:aws:es:us-east-1:Account-B-ID:domain/cross-account-cluster/"
      ]
    }
  ]
}
```

3. Crie um stream do Firehose na conta A usando a função do IAM que você criou na etapa 1. Ao criar o stream do Firehose, use as APIs AWS CLI ou as APIs do Amazon Data Firehose e especifique o `ClusterEndpoint` campo em vez de Service. `DomainARN` OpenSearch

> **Note**
>
> Para criar um stream do Firehose em uma AWS conta com um destino de OpenSearch serviço em uma conta diferente, você deve usar as APIs AWS CLI ou as APIs do Amazon Data Firehose. Você não pode usar o AWS Management Console para criar esse tipo de configuração entre contas.

Uso de tags para controle de acesso

Você pode usar o `Condition` elemento opcional (ou `Condition` bloco) em uma política do IAM para ajustar o acesso às operações do Amazon Data Firehose com base nas chaves e valores das tags. As subseções a seguir descrevem como fazer isso para as diferentes operações do Amazon Data Firehose. Para saber mais sobre o uso do elemento `Condition` e as operações que você pode usar com ele, consulte Elementos de política JSON do IAM: condição.

CreateDeliveryStream

Para a operação `CreateDeliveryStream`, use a chave de condição `aws:RequestTag`. No exemplo a seguir, `MyKey` e `MyValue` representam a chave e o valor correspondente de uma tag. Para obter mais informações, consulte Conceitos básicos de tags.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "firehose:CreateDeliveryStream",
            "firehose:TagDeliveryStream"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/MyKey": "MyValue"
            }
        }
    }]
}
```

TagDeliveryStream

Para a operação `TagDeliveryStream`, use a chave de condição `aws:TagKeys`. No exemplo a seguir, `MyKey` é um exemplo de chave de tag.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "firehose:TagDeliveryStream",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": "MyKey"
                 }
            }
        }
    ]
```

```
}
```

UntagDeliveryStream

Para a operação `UntagDeliveryStream`, use a chave de condição `aws:TagKeys`. No exemplo a seguir, `MyKey` é um exemplo de chave de tag.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "firehose:UntagDeliveryStream",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": "MyKey"
                 }
            }
        }
    ]
}
```

ListDeliveryStreams

Não é possível usar controle de acesso com base em tags com `ListDeliveryStreams`.

Outras operações do Amazon Data Firehose

Para todas as operações do Amazon Data Firehose`CreateDeliveryStream`, exceto,`TagDeliveryStream`, e `UntagDeliveryStreamListDeliveryStreams`, use a chave de `aws:RequestTag` condição. No exemplo a seguir, `MyKey` e `MyValue` representam a chave e o valor correspondente de uma tag.

`ListDeliveryStreams`, use a chave de `firehose:ResourceTag` condição para controlar o acesso com base nas tags desse stream do Firehose.

No exemplo a seguir, `MyKey` e `MyValue` representam a chave e o valor correspondente de uma tag. A política só se aplicaria aos fluxos do Data Firehose com uma tag nomeada `MyKey` com um valor de. `MyValue` Para obter mais informações sobre como controlar o acesso com base em tags

de recursos, consulte Como controlar o acesso a AWS recursos usando tags no Guia do usuário do IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
            "Effect": "Deny",
            "Action": "firehose:DescribeDeliveryStream",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                      "firehose:ResourceTag/MyKey": "MyValue"
                  }
            }
        }
    ]
}
```

Autentique-se com o AWS Secrets Manager Amazon Data Firehose

O Amazon Data Firehose se integra AWS Secrets Manager para fornecer acesso seguro aos seus segredos e automatizar a rotação de credenciais. Essa integração permite que o Firehose recupere um segredo do Secrets Manager em tempo de execução para se conectar aos destinos de streaming mencionados anteriormente e fornecer seus fluxos de dados. Com isso, seus segredos não são visíveis em texto simples durante o fluxo de trabalho de criação do stream, AWS Management Console nem nos parâmetros da API. Ele fornece uma prática segura para gerenciar seus segredos e dispensa você de atividades complexas de gerenciamento de credenciais, como a configuração de funções personalizadas do Lambda para gerenciar rotações de senhas.

Para obter mais informações, consulte o AWS Secrets Manager Guia de usuário do .

Entenda os segredos

Um segredo pode ser uma senha, um conjunto de credenciais, como um nome de usuário e senha, um token OAuth ou outras informações secretas que você armazena em formato criptografado no Secrets Manager.

Para cada destino, você deve especificar o par de valores-chave secretos no formato JSON correto, conforme mostrado na seção a seguir. O Amazon Data Firehose não conseguirá se conectar ao seu destino se seu segredo não tiver o formato JSON correto de acordo com o destino.

Formato de segredo para o cluster provisionado do Amazon Redshift e o grupo de trabalho Amazon Redshift Serverless

```
{
    "username":  "<username>",
    "password":  "<password>"
}
```

Formato secreto para o Splunk

```
{
    "hec_token":  "<hec token>"
}
```

Formato secreto para Snowflake

```
{
    "user":  "<user>",
    "private_key":  "<private_key>",
    "key_passphrase":  "<passphrase>" // optional
}
```

Formato secreto para endpoint HTTP, Coralogix, Datadog, Dynatrace, Elastic, Honeycomb, Logz.io, MongoDB Cloud e New Relic LogicMonitor

```
{
    "api_key":  "<apikey>"
}
```

Criar um segredo

Para criar um segredo, siga as etapas em Criar um AWS Secrets Manager segredo no Guia do AWS Secrets Manager usuário.

Use o segredo

Recomendamos que você use AWS Secrets Manager para armazenar suas credenciais ou chaves para se conectar a destinos de streaming, como Amazon Redshift, endpoint HTTP, Snowflake, Splunk, Coralogix, Datadog, Dynatrace, Elastic, Honeycomb, Logz.io, MongoDB Cloud e New Relic. LogicMonitor

Você pode configurar a autenticação com o Secrets Manager para esses destinos por meio do AWS Management Console no momento da criação do stream do Firehose. Para ter mais informações, consulte Definir configurações de destino. Como alternativa, você também pode usar as operações CreateDeliveryStreame da UpdateDestinationAPI para configurar a autenticação com o Secrets Manager.

O Firehose armazena os segredos em cache com uma criptografia e os usa em todas as conexões com os destinos. Ele atualiza o cache a cada 10 minutos para garantir que as credenciais mais recentes sejam usadas.

Você pode optar por desativar a capacidade de recuperar segredos do Secrets Manager a qualquer momento durante o ciclo de vida do stream. Se você não quiser usar o Secrets Manager para recuperar segredos, você pode usar o nome de usuário/senha ou a chave de API.

> **Note**
>
> Embora não haja custo adicional para esse recurso no Firehose, você será cobrado pelo acesso e pela manutenção do Secrets Manager. Para obter mais informações, consulte a página AWS Secrets Managerde preços.

Conceda acesso ao Firehose para recuperar o segredo

Para que o Firehose recupere um segredo AWS Secrets Manager, você deve fornecer ao Firehose as permissões necessárias para acessar o segredo e a chave que criptografa seu segredo.

Ao usar AWS Secrets Manager para armazenar e recuperar segredos, há algumas opções de configuração diferentes, dependendo de onde o segredo está armazenado e de como é criptografado.

- Se o segredo estiver armazenado na mesma AWS conta da sua função do IAM e estiver criptografado com a chave AWS gerenciada padrão (`aws/secretsmanager`), a função do IAM

que a Firehose presume só precisará de `secretsmanager:GetSecretValue` permissão sobre o segredo.

```
// secret role policy
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "Secret ARN"
        }
    ]
}
```

Para obter mais informações sobre as políticas do IAM, consulte Exemplos de políticas de permissões para AWS Secrets Manager.

- Se o segredo estiver armazenado na mesma conta da função, mas criptografado com uma chave gerenciada pelo cliente (CMK), a função precisará de ambas `secretsmanager:GetSecretValue` e de `kms:Decrypt` permissões. A política da CMK também precisa permitir que a função do IAM seja `kms:Decrypt` executada.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "Secret ARN"
        },
        {
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "KMSKeyARN"
        }
    ]
}
```

- Se o segredo estiver armazenado em uma AWS conta diferente da sua função e for criptografado com a chave AWS gerenciada padrão, essa configuração não será possível, pois o Secrets Manager não permite acesso entre contas quando o segredo é criptografado com a chave AWS gerenciada.

- Se o segredo for armazenado em uma conta diferente e criptografado com uma CMK, o papel do IAM precisará de `secretsmanager:GetSecretValue` permissão sobre o segredo e `kms:Decrypt` permissão na CMK. A política de recursos do segredo e a política de CMK na outra conta também precisam permitir que a função do IAM tenha as permissões necessárias. Para obter mais informações, consulte Acesso entre contas.

Gire o segredo

A rotação é quando você atualiza periodicamente um segredo. Você pode configurar AWS Secrets Manager para alternar automaticamente o segredo para você em uma programação especificada por você. Dessa forma, você pode substituir segredos de longo prazo por segredos de curto prazo. Isso ajuda a reduzir o risco de comprometimento. Para obter mais informações, consulte Rotacionar AWS Secrets Manager segredos no Guia do AWS Secrets Manager usuário.

Gerencie funções do IAM por meio do console Amazon Data Firehose

O Amazon Data Firehose é um serviço totalmente gerenciado que fornece dados de streaming em tempo real para destinos. Você também pode configurar o Firehose para transformar e converter o formato dos seus dados antes da entrega. Para usar esses recursos, primeiro você deve fornecer funções do IAM para conceder permissões ao Firehose ao criar ou editar um stream do Firehose. O Firehose usa essa função do IAM para todas as permissões que o stream do Firehose precisa.

Por exemplo, considere um cenário em que você cria um stream do Firehose que entrega dados para o Amazon S3, e esse stream do Firehose tem registros de origem do Transform com o recurso ativado. AWS Lambda Nesse caso, você deve fornecer funções do IAM para conceder permissões ao Firehose para acessar o bucket do S3 e invocar a função Lambda, conforme mostrado a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "lambdaProcessing",
        "Effect": "Allow",
        "Action": ["lambda:InvokeFunction", "lambda:GetFunctionConfiguration"],
        "Resource": "arn:aws:lambda:us-east-1:<account id>:function:<lambda function
 name>:<lambda function version>"
    }, {
        "Sid": "s3Permissions",
```

```
        "Effect": "Allow",
        "Action": ["s3:AbortMultipartUpload", "s3:GetBucketLocation", "s3:GetObject",
 "s3:ListBucket", "s3:ListBucketMultipartUploads", "s3:PutObject"],
        "Resource": ["arn:aws:s3:::<bucket name>", "arn:aws:s3:::<bucket name>/*"]
    }]
}
```

O console Firehose permite que você escolha como deseja fornecer essas funções. Você pode escolher uma das opções a seguir.

- Escolha uma função existente do IAM
- Crie uma nova função do IAM a partir do console

Escolha uma função existente do IAM

Você pode escolher entre uma função existente do IAM. Com essa opção, certifique-se de que a função do IAM escolhida tenha uma política de confiança adequada e as permissões necessárias para sua origem e destino. Para ter mais informações, consulte Controle de acesso com o Amazon Data Firehose.

Crie uma nova função do IAM a partir do console

Como alternativa, você também pode usar o console Firehose para criar uma nova função em seu nome.

Quando o Firehose cria uma função do IAM em seu nome, a função inclui automaticamente todas as políticas de permissão e confiança que concedem as permissões necessárias com base na configuração do stream do Firehose.

Por exemplo, se você não habilitou Transformar registros de origem com o AWS Lambda recurso, o console gerará a seguinte declaração na política de permissão.

```
{
  "Sid": "lambdaProcessing",
  "Effect": "Allow",
   "Action": [
     "lambda:InvokeFunction",
     "lambda:GetFunctionConfiguration"
   ],
   "Resource": "arn:aws:lambda:us-east-1:<account id>:function:
%FIREHOSE_POLICY_TEMPLATE_PLACEHOLDER%"
```

```
}
```

> **Note**
>
> É seguro ignorar as declarações de política contidas nela`%FIREHOSE_POLICY_TEMPLATE_PLACEHOLDER%`, pois elas não concedem permissões sobre nenhum recurso.

O console cria e edita fluxos de trabalho de stream do Firehose também cria uma política de confiança e a anexa à função do IAM. A política de confiança permite que o Firehose assuma a função do IAM. Veja a seguir um exemplo de uma política de confiança.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "firehoseAssume",
        "Effect": "Allow",
        "Principal": {
            "Service": "firehose.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }]
}
```

> **Important**
>
> - Você deve evitar usar a mesma função do IAM gerenciada pelo console para vários streams do Firehose. Caso contrário, a função do IAM pode se tornar excessivamente permissiva ou resultar em erros.
> - Para usar declarações de política diferentes em uma política de permissão de uma função do IAM gerenciada pelo console, você pode criar sua própria função do IAM e copiar as declarações de política para uma política de permissão anexada à nova função. Para anexar a função ao stream do Firehose, selecione a opção Escolher função do IAM existente no acesso ao serviço.
> - O console gerencia qualquer função do IAM que contenha a string service-role em seu ARN. Ao escolher a opção de função do IAM existente, certifique-se de selecionar uma

função do IAM sem a string da função de serviço em seu ARN para que o console não faça nenhuma alteração nela.

Etapas para criar uma função do IAM a partir do console

1. Abra o console do Firehose em https://console.aws.amazon.com/firehose/.
2. Escolha Create Firehose stream.
3. Escolha uma origem e um destino. Para ter mais informações, consulte Crie um stream do Firehose.
4. Escolha as configurações de destino. Para ter mais informações, consulte Definir configurações de destino.
5. Em Configurações avançadas, para acesso ao serviço, escolha Criar ou atualizar a função do IAM.

> **Note**
>
> Essa é uma opção padrão. Para usar uma função existente, selecione a opção Escolher função do IAM existente. O console Firehose não fará nenhuma alteração em sua própria função.

6. Escolha Create Firehose stream.

Editar a função do IAM no console

Quando você edita um stream do Firehose, o Firehose atualiza a política de permissão correspondente de acordo com as alterações de configuração e permissão.

Por exemplo, quando você edita o stream do Firehose e ativa o AWS Lambda recurso Transform source records with using the latest version of Lambda function as`exampleLambdaFunction`, você obtém a seguinte declaração de política na política de permissão.

```
{
  "Sid": "lambdaProcessing",
  "Effect": "Allow",
  "Action": [
    "lambda:InvokeFunction",
```

```
    "lambda:GetFunctionConfiguration"
  ],
  "Resource": "arn:aws:lambda:us-east-1:<account id>:function:exampleLambdaFunction:
$LATEST"
}
```

> **Important**
>
> Uma função do IAM gerenciada pelo console foi projetada para ser autônoma. Não recomendamos que você modifique a política de permissão ou a política de confiança fora do console.

Editar a função do IAM no console

1. Abra o console do Firehose em https://console.aws.amazon.com/firehose/.
2. Escolha Streams do Firehose e escolha o nome de um stream do Firehose que você deseja atualizar.
3. Na guia Configuração, na seção Acesso ao servidor, escolha Editar.
4. Atualize a opção de função do IAM.

   > **Note**
   >
   > Por padrão, o console sempre atualiza uma função do IAM com a função de serviço padrão em seu ARN. Ao escolher a opção de função do IAM existente, certifique-se de selecionar uma função do IAM sem a string da função de serviço em seu ARN para que o console não faça nenhuma alteração nela.

5. Escolha Salvar alterações.

Monitoramento do Amazon Data Firehose

O Amazon Data Firehose fornece funcionalidade de monitoramento para seus streams do Firehose. Para ter mais informações, consulte Monitorar.

Validação de conformidade para Amazon Data Firehose

Auditores terceirizados avaliam a segurança e a conformidade do Amazon Data Firehose como parte de AWS vários programas de conformidade. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

Para obter uma lista de AWS serviços no escopo de programas de conformidade específicos, consulte AWS Serviços no escopo por programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixando relatórios no AWS Artifact.

Sua responsabilidade de conformidade ao usar o Data Firehose é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade da sua empresa e pelas leis e regulamentações aplicáveis. Se o uso do Data Firehose estiver sujeito à conformidade com padrões como HIPAA, PCI ou FedRAMP, fornece recursos para ajudar a: AWS

- Guias de início rápido sobre segurança e conformidade — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos focados em segurança e conformidade em. AWS
- Documento técnico sobre arquitetura para segurança e conformidade com a HIPAA — Este whitepaper descreve como as empresas podem usar para criar aplicativos compatíveis com a HIPAA. AWS
- AWS Recursos de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- AWS Config— Esse AWS serviço avalia se suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub— Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno, AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Resiliência no Amazon Data Firehose

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as Zonas

de Disponibilidade, é possível projetar e operar aplicações e bancos de dados que executem o failover automaticamente entre as Zonas de Disponibilidade sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenter tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Além da infraestrutura AWS global, o Data Firehose oferece vários recursos para ajudar a suportar suas necessidades de resiliência e backup de dados.

Recuperação de desastres

O Amazon Data Firehose é executado em modo sem servidor e cuida da degradação do host, da disponibilidade da zona de disponibilidade e de outros problemas relacionados à infraestrutura por meio da migração automática. Quando isso acontece, o Amazon Data Firehose garante que o stream do Firehose seja migrado sem perda de dados.

Segurança da infraestrutura no Amazon Data Firehose

Como um serviço gerenciado, o Amazon Data Firehose é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar o Firehose pela 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.

> **Note**
>
> Para solicitações HTTPS de saída, o Amazon Data Firehose usa uma biblioteca HTTP que seleciona automaticamente a versão mais alta do protocolo TLS suportada no lado de destino.

VPC endpoints (PrivateLink)

O Amazon Data Firehose fornece suporte para VPC endpoints (). PrivateLink Para ter mais informações, consulte Usando o Amazon Data Firehose com AWS PrivateLink.

Melhores práticas de segurança para o Amazon Data Firehose

O Amazon Data Firehose fornece vários recursos de segurança a serem considerados ao desenvolver e implementar suas próprias políticas de segurança. As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes no seu ambiente, trate-as como considerações úteis em vez de requisitos.

Implemente o acesso de privilégio mínimo

Ao conceder permissões, você decide quem está recebendo quais permissões para quais recursos do Amazon Data Firehose. Você habilita ações específicas que quer permitir nesses recursos. Portanto, você deve conceder somente as permissões necessárias para executar uma tarefa. A implementação do privilégio de acesso mínimo é fundamental para reduzir o risco de segurança e o impacto que pode resultar de erros ou usuários mal-intencionados.

Usar funções do IAM

Os aplicativos produtores e clientes devem ter credenciais válidas para acessar os streams do Firehose, e seu stream do Firehose deve ter credenciais válidas para acessar os destinos. Você não deve armazenar AWS credenciais diretamente em um aplicativo cliente ou em um bucket do Amazon S3. Essas são credenciais de longo prazo que não são automaticamente alternadas e podem ter um impacto comercial significativo se forem comprometidas.

Em vez disso, você deve usar uma função do IAM para gerenciar credenciais temporárias para que seus aplicativos de produtor e cliente acessem os streams do Firehose. Quando você usa uma

função, não precisa usar credenciais de longo prazo (como um nome de usuário e uma senha ou chaves de acesso) para acessar outros recursos.

Para obter mais informações, consulte os seguintes tópicos no Manual do usuário do IAM:

- Funções do IAM
- Cenários comuns para funções: usuários, aplicativos e serviços

Implemente a criptografia do lado do servidor em recursos dependentes

Dados em repouso e dados em trânsito podem ser criptografados no Amazon Data Firehose. Para obter mais informações, consulte Proteção de dados no Amazon Amazon Data Firehose.

Use CloudTrail para monitorar chamadas de API

O Amazon Data Firehose é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Amazon Data Firehose.

Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Amazon Data Firehose, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para ter mais informações, consulte the section called “Registrando chamadas de API do Amazon Data Firehose com AWS CloudTrail”.

Transformação de dados do Amazon Data Firehose

O Amazon Data Firehose pode invocar sua função Lambda para transformar os dados de origem recebidos e entregar os dados transformados aos destinos. Você pode ativar a transformação de dados do Amazon Data Firehose ao criar seu stream do Firehose.

Fluxo de transformação de dados

Quando você ativa a transformação de dados do Firehose, o Firehose armazena os dados recebidos em buffer. A dica de tamanho do buffer varia entre 0,2 MB e 3 MB. A dica padrão do tamanho do buffer do Lambda é de 1 MB para todos os destinos, exceto Splunk e Snowflake. Para Splunk e Snowflake, a dica de buffer padrão é 256 KB. A dica de intervalo de buffer do Lambda varia entre 0 e 900 segundos. A dica padrão de intervalo de buffer do Lambda é de sessenta segundos para todos os destinos, exceto o Snowflake. Para o Snowflake, o intervalo padrão de dica de buffer é de 30 segundos. Para ajustar o tamanho do buffer, defina o ProcessingConfigurationparâmetro da UpdateDestinationAPI CreateDeliveryStreamou com o ProcessorParameterchamado e. `BufferSizeInMBs` `IntervalInSeconds` Em seguida, o Firehose invoca a função Lambda especificada de forma assíncrona com cada lote armazenado em buffer usando o modo de invocação síncrona. AWS Lambda Os dados transformados são enviados do Lambda para o Firehose. O Firehose então o envia para o destino quando o tamanho do buffer de destino especificado ou o intervalo de buffer é atingido, o que ocorrer primeiro.

> **Important**
>
> O modo de invocação síncrona do Lambda tem um limite de tamanho de carga útil de 6 MB para ambas a solicitação e a resposta. Certifique-se de que o tamanho do armazenamento em buffer para envio da solicitação para a função seja menor que ou igual a 6 MB. Além disso, verifique se a resposta que sua função retorna não excede 6 MB.

Transformação de dados e modelo de status

Todos os registros transformados do Lambda devem conter os seguintes parâmetros, ou o Amazon Data Firehose os rejeitará e tratará isso como uma falha na transformação de dados.

Para o Kinesis Data Streams e o Direct PUT:

recordId

O ID do registro é passado do Amazon Data Firehose para o Lambda durante a invocação. O registro transformado deve conter o mesmo ID de registro. Qualquer incompatibilidade entre o ID do registro original e o ID do registro transformado é considerada uma falha na transformação de dados.

resultado

O status da transformação de dados do registro. Os valores possíveis são: Ok (o registro foi transformado com êxito), `Dropped` (o registro foi removido intencionalmente pela lógica de processamento), e `ProcessingFailed` (não foi possível transformar o registro). Se um registro tiver um status de Ok ou`Dropped`, o Amazon Data Firehose o considerará processado com sucesso. Caso contrário, o Amazon Data Firehose o considera processado sem sucesso.

data

A carga útil dos dados transformados, após a codificação base64.

Este é um exemplo de saída de resultados do Lambda:

```
 {
    "recordId": "<recordId from the Lambda input>",
    "result": "Ok",
    "data": "<Base64 encoded Transformed data>"
}
```

Para o Amazon MSK

recordId

O ID do registro é passado do Firehose para o Lambda durante a invocação. O registro transformado deve conter o mesmo ID de registro. Qualquer incompatibilidade entre o ID do registro original e o ID do registro transformado é considerada uma falha na transformação de dados.

resultado

O status da transformação de dados do registro. Os valores possíveis são: Ok (o registro foi transformado com êxito), `Dropped` (o registro foi removido intencionalmente pela lógica de processamento), e `ProcessingFailed` (não foi possível transformar o registro). Se um registro

tiver um status de `Ok` ou`Dropped`, o Firehose o considerará processado com sucesso. Caso contrário, o Firehose o considera processado sem sucesso.

KafkaRecordValue

A carga útil dos dados transformados, após a codificação base64.

Este é um exemplo de saída de resultados do Lambda:

```
{
    "recordId": "<recordId from the Lambda input>",
    "result": "Ok",
    "kafkaRecordValue": "<Base64 encoded Transformed data>"
}
```

Esquema do Lambda

Esses esquemas demonstram como você pode criar e usar funções AWS Lambda para transformar dados em seus fluxos de dados do Amazon Data Firehose.

Para ver as plantas que estão disponíveis no console AWS Lambda

1. Faça login no AWS Management Console e abra o AWS Lambda console em https://console.aws.amazon.com/lambda/.
2. Selecione Create function (Criar função) e Use a blueprint (Usar um esquema).
3. No campo Blueprints, pesquise a palavra-chave `firehose` para encontrar os blueprints Lambda do Amazon Data Firehose.

Lista de esquemas:

- Processar registros enviados para o stream do Amazon Data Firehose (Node.js, Python)

  Este esquema mostra um exemplo básico de como processar dados em seu stream de dados do Firehose usando AWS o Lambda.

  Data da versão mais recente: novembro de 2016.

  Notas da versão: nenhuma.
- CloudWatch Registros do processo enviados para o Firehose

Esse blueprint está obsoleto. Para obter informações sobre o processamento de CloudWatch registros enviados para o Firehose, consulte Gravando no Firehose usando registros. CloudWatch

- Converta registros de stream do Amazon Data Firehose no formato syslog em JSON (Node.js)

  Este esquema mostra como você pode converter os registros de entrada no formato RFC3164 Syslog em JSON.

  Data da versão mais recente: novembro de 2016.

  Notas da versão: nenhuma.

Para ver as plantas que estão disponíveis no AWS Serverless Application Repository

1. Acesse AWS Serverless Application Repository.
2. Escolha Procurar todas as aplicações.
3. No campo Applications (Aplicativos) procure a palavra-chave `firehose`.

Também é possível criar uma função do Lambda sem usar um esquema. Consulte Introdução ao AWS Lambda.

Tratamento de falhas de transformação de dados

Se a invocação da função Lambda falhar devido a um tempo limite de rede ou porque você atingiu o limite de invocação do Lambda, o Amazon Data Firehose repetirá a invocação três vezes por padrão. Se a invocação não for bem-sucedida, o Amazon Data Firehose ignorará esse lote de registros. Os registros ignorados são tratados como registros com falha no processamento. Você pode especificar ou substituir as opções de nova tentativa usando a API CreateDeliveryStreamou`UpdateDestination`. Para esse tipo de falha, você pode registrar erros de invocação no Amazon CloudWatch Logs. Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando registros CloudWatch .

Se o status da transformação de dados de um registro for`ProcessingFailed`, o Amazon Data Firehose tratará o registro como processado sem sucesso. Para esse tipo de falha, você pode emitir registros de erro para o Amazon CloudWatch Logs a partir da sua função Lambda. Para obter mais informações, consulte Como acessar o Amazon CloudWatch Logs AWS Lambda no Guia do AWS Lambda desenvolvedor.

Se a transformação de dados apresentar falha, os registros com falha de processamento serão entregues ao bucket do S3 na pasta `processing-failed`. Os registros têm o seguinte formato:

```
{
    "attemptsMade": "count",
    "arrivalTimestamp": "timestamp",
    "errorCode": "code",
    "errorMessage": "message",
    "attemptEndingTimestamp": "timestamp",
    "rawData": "data",
    "lambdaArn": "arn"
}
```

`attemptsMade`

O número de tentativas de solicitações de invocação.

`arrivalTimestamp`

A hora em que o registro foi recebido pelo Amazon Data Firehose.

`errorCode`

O código de erro de HTTP retornado pelo Lambda.

`errorMessage`

A mensagem de erro retornada pelo Lambda.

`attemptEndingTimestamp`

A vez em que o Amazon Data Firehose parou de tentar invocações do Lambda.

`rawData`

Os dados de registro com codificação base64.

`lambdaArn`

O nome do recurso da Amazon (ARN) da função do Lambda.

Duração de uma invocação do Lambda

O Amazon Data Firehose suporta um tempo de invocação do Lambda de até 5 minutos. Se sua função do Lambda levar mais de 5 minutos para ser concluída, você receberá o seguinte erro: O

Firehose encontrou erros de tempo limite ao chamar o Lambda. AWS O tempo limite máximo da função é de 5 minutos.

Para obter informações sobre o que o Amazon Data Firehose fará se esse erro ocorrer, consulte. the section called “Tratamento de falhas de transformação de dados”

Período de retenção de backup do registro de origem

O Amazon Data Firehose pode fazer backup de todos os registros não transformados em seu bucket S3 simultaneamente enquanto entrega os registros transformados ao destino. Você pode ativar o backup do registro de origem ao criar ou atualizar seu stream do Firehose. Não é possível desabilitar o período de retenção de backup do registro de origem após habilitá-lo.

Particionamento dinâmico no Amazon Data Firehose

O particionamento dinâmico permite particionar continuamente dados de streaming no Firehose usando chaves dentro dos dados (por exemplo, `customer_id` `outransaction_id`) e depois entregar os dados agrupados por essas chaves nos prefixos correspondentes do Amazon Simple Storage Service (Amazon S3). Isso facilita a execução de análises econômicas e de alto desempenho em dados de streaming no Amazon S3 usando vários serviços, como Amazon Athena, Amazon EMR, Amazon Redshift Spectrum e Amazon. QuickSight Além disso, o AWS Glue pode realizar trabalhos mais sofisticados de extração, transformação e carregamento (ETL) depois que os dados de streaming particionados dinamicamente são entregues ao Amazon S3, em casos de uso em que é necessário processamento adicional.

Particionar os dados minimiza a quantidade de dados digitalizados, otimiza a performance e reduz os custos de consultas de análise no Amazon S3. Também aumenta o acesso granular aos dados. Os streams Firehose são tradicionalmente usados para capturar e carregar dados no Amazon S3. Para particionar um conjunto de dados em streaming para análises baseadas no Amazon S3, você precisaria executar aplicações de particionamento entre buckets do Amazon S3 antes de disponibilizar os dados para análise, o que pode se tornar complicado ou caro.

Com o particionamento dinâmico, o Firehose agrupa continuamente dados em trânsito usando chaves de dados definidas de forma dinâmica ou estática e entrega os dados para prefixos individuais do Amazon S3 por chave. Isso reduz time-to-insight em minutos ou horas. Também reduz os custos e simplifica as arquiteturas.

Tópicos

Chaves de particionamento

Com o particionamento dinâmico, você cria conjuntos de dados direcionados a partir dos dados do S3 em streaming particionando os dados com base em chaves de particionamento. As chaves de particionamento permitem que você filtre os dados em streaming com base em valores específicos. Por exemplo, se você precisar filtrar os dados com base no ID do cliente e no país, poderá especificar o campo de dados de `customer_id` como uma chave de particionamento e o campo de dados de `country` como outra chave de particionamento. Em seguida, você especifica as expressões (usando os formatos compatíveis) para definir os prefixos de bucket do S3 aos quais os registros de dados particionados dinamicamente devem ser entregues.

Estes são os métodos aceitos para criar chaves de particionamento:

- Análise embutida - esse método usa o mecanismo de suporte integrado do Firehose, um analisador jq, para extrair as chaves para particionamento de registros de dados que estão no formato JSON. Atualmente, oferecemos suporte apenas à `jq 1.6` versão.
- AWS Função Lambda - esse método usa uma função AWS Lambda especificada para extrair e retornar os campos de dados necessários para o particionamento.

> **Important**
>
> Ao habilitar o particionamento dinâmico, você deve configurar pelo menos um desses métodos para particionar os dados. Você pode configurar qualquer um desses métodos para especificar as chaves de particionamento ou ambos ao mesmo tempo.

Criar chaves de particionamento com análise em linha

Para configurar a análise em linha como o método de particionamento dinâmico para os dados em streaming, você deve escolher os parâmetros de registro de dados a serem usados como chaves de particionamento e fornecer um valor para cada chave de particionamento especificada.

O exemplo de registro de dados a seguir mostra como você pode definir chaves de particionamento para ele com análise embutida. Observe que os dados devem ser codificados no formato Base64. Você também pode consultar o exemplo da CLI.

```
{
   "type": {
```

```
    "device": "mobile",
    "event": "user_clicked_submit_button"
  },
  "customer_id": "1234567890",
  "event_timestamp": 1565382027,   #epoch timestamp
  "region": "sample_region"
}
```

Por exemplo, você pode escolher particionar os dados com base no parâmetro `customer_id` ou no parâmetro `event_timestamp`. Isso significa que você deseja que o valor do parâmetro `customer_id` ou do parâmetro `event_timestamp` em cada registro seja usado para determinar o prefixo do S3 ao qual o registro deve ser entregue. Você também pode escolher um parâmetro aninhado, como `device` com uma expressão `.type.device`. A lógica de particionamento dinâmico pode depender de vários parâmetros.

Depois de selecionar os parâmetros dos dados para as chaves de particionamento, você mapeia cada parâmetro para uma expressão jq válida. A tabela a seguir mostra esse mapeamento de parâmetros para expressões jq:

Parâmetro	Expressão jq
`customer_id`	.customer_id
`device`	.type.device
`year`	.event_timestamp| strftime("%Y")
`month`	.event_timestamp| strftime("%m")
`day`	.event_timestamp| strftime("%d")
`hour`	.event_timestamp| strftime("%H")

Em tempo de execução, o Firehose usa a coluna direita acima para avaliar os parâmetros com base nos dados de cada registro.

Criar chaves de particionamento com uma função do AWS Lambda

Para registros de dados compactados ou criptografados, ou dados que estejam em qualquer formato de arquivo que não seja JSON, você pode usar a função AWS Lambda integrada com

seu próprio código personalizado para descompactar, descriptografar ou transformar os registros a fim de extrair e retornar os campos de dados necessários para o particionamento. Essa é uma expansão da função Lambda de transformação existente que está disponível atualmente com o Firehose. Você pode transformar, analisar e retornar os campos de dados que podem ser usados para particionamento dinâmico usando a mesma função do Lambda.

Veja a seguir um exemplo da função Lambda de processamento de stream do Firehose em Python que reproduz cada registro lido da entrada à saída e extrai as chaves de particionamento dos registros.

```
from __future__ import print_function
import base64
import json
import datetime

# Signature for all Lambda functions that user must implement
def lambda_handler(firehose_records_input, context):
    print("Received records for processing from DeliveryStream: " +
 firehose_records_input['deliveryStreamArn']
          + ", Region: " + firehose_records_input['region']
          + ", and InvocationId: " + firehose_records_input['invocationId'])

    # Create return value.
    firehose_records_output = {'records': []}

    # Create result object.
    # Go through records and process them

    for firehose_record_input in firehose_records_input['records']:
        # Get user payload
        payload = base64.b64decode(firehose_record_input['data'])
        json_value = json.loads(payload)

        print("Record that was received")
        print(json_value)
        print("\n")
        # Create output Firehose record and add modified payload and record ID to it.
        firehose_record_output = {}
        event_timestamp = datetime.datetime.fromtimestamp(json_value['eventTimestamp'])
        partition_keys = {"customerId": json_value['customerId'],
                          "year": event_timestamp.strftime('%Y'),
                          "month": event_timestamp.strftime('%m'),
```

```
                        "date": event_timestamp.strftime('%d'),
                        "hour": event_timestamp.strftime('%H'),
                        "minute": event_timestamp.strftime('%M')
                        }

        # Create output Firehose record and add modified payload and record ID to it.
        firehose_record_output = {'recordId': firehose_record_input['recordId'],
                                  'data': firehose_record_input['data'],
                                  'result': 'Ok',
                                  'metadata': { 'partitionKeys': partition_keys }}

        # Must set proper record ID
        # Add the record to the list of output records.

        firehose_records_output['records'].append(firehose_record_output)

    # At the end return processed records
    return firehose_records_output
```

Veja a seguir um exemplo da função Lambda de processamento de stream do Firehose em Go que reproduz cada registro lido da entrada à saída e extrai as chaves de particionamento dos registros.

```
package main

import (
 "fmt"
 "encoding/json"
 "time"
 "strconv"

 "github.com/aws/aws-lambda-go/events"
 "github.com/aws/aws-lambda-go/lambda"
)

type DataFirehoseEventRecordData struct {
 CustomerId string `json:"customerId"`
}

func handleRequest(evnt events.DataFirehoseEvent) (events.DataFirehoseResponse, error)
 {

 fmt.Printf("InvocationID: %s\n", evnt.InvocationID)
```

```
 fmt.Printf("DeliveryStreamArn: %s\n", evnt.DeliveryStreamArn)
 fmt.Printf("Region: %s\n", evnt.Region)

 var response events.DataFirehoseResponse

 for _, record := range evnt.Records {
  fmt.Printf("RecordID: %s\n", record.RecordID)
  fmt.Printf("ApproximateArrivalTimestamp: %s\n", record.ApproximateArrivalTimestamp)

  var transformedRecord events.DataFirehoseResponseRecord
  transformedRecord.RecordID = record.RecordID
  transformedRecord.Result = events.DataFirehoseTransformedStateOk
  transformedRecord.Data = record.Data

  var metaData events.DataFirehoseResponseRecordMetadata
  var recordData DataFirehoseEventRecordData
  partitionKeys := make(map[string]string)

  currentTime := time.Now()
  json.Unmarshal(record.Data, &recordData)
  partitionKeys["customerId"] = recordData.CustomerId
  partitionKeys["year"] = strconv.Itoa(currentTime.Year())
  partitionKeys["month"] = strconv.Itoa(int(currentTime.Month()))
  partitionKeys["date"] = strconv.Itoa(currentTime.Day())
  partitionKeys["hour"] = strconv.Itoa(currentTime.Hour())
  partitionKeys["minute"] = strconv.Itoa(currentTime.Minute())
  metaData.PartitionKeys = partitionKeys
  transformedRecord.Metadata = metaData

  response.Records = append(response.Records, transformedRecord)
 }

 return response, nil
}

func main() {
 lambda.Start(handleRequest)
}
```

Prefixo de bucket do Amazon S3 para particionamento dinâmico

Ao criar um stream do Firehose que usa o Amazon S3 como destino, você deve especificar um bucket do Amazon S3 onde o Firehose deve entregar seus dados. Os prefixos de bucket do Amazon S3 são usados para organizar os dados armazenados nos buckets do S3. Um prefixo de bucket do Amazon S3 é semelhante a um diretório que permite agrupar objetos semelhantes.

Com o particionamento dinâmico, os dados particionados são entregues nos prefixos especificados do Amazon S3. Se você não habilitar o particionamento dinâmico, especificar um prefixo de bucket do S3 para seu stream do Firehose é opcional. No entanto, se você optar por ativar o particionamento dinâmico, deverá especificar os prefixos de bucket do S3 para os quais o Firehose entrega dados particionados.

Em cada stream do Firehose em que você ativa o particionamento dinâmico, o valor do prefixo do bucket do S3 consiste em expressões com base nas chaves de particionamento especificadas para esse stream do Firehose. Usando novamente o exemplo de registro de dados acima, você pode criar o seguinte valor de prefixo do S3 que consiste em expressões com base nas chaves de particionamento definidas acima:

```
"ExtendedS3DestinationConfiguration": {
"BucketARN": "arn:aws:s3:::my-logs-prod",
"Prefix": "customer_id=!{partitionKeyFromQuery:customer_id}/
    device=!{partitionKeyFromQuery:device}/
    year=!{partitionKeyFromQuery:year}/
    month=!{partitionKeyFromQuery:month}/
    day=!{partitionKeyFromQuery:day}/
    hour=!{partitionKeyFromQuery:hour}/"
}
```

O Firehose avalia a expressão acima em tempo de execução. Ele agrupa os registros que correspondem à mesma expressão de prefixo S3 avaliada para um único conjunto de dados. O Firehose então entrega cada conjunto de dados ao prefixo S3 avaliado. A frequência de entrega do conjunto de dados para o S3 é determinada pela configuração do buffer de fluxo do Firehose. Assim sendo, o registro neste exemplo é entregue à seguinte chave de objeto do S3:

```
s3://my-logs-prod/customer_id=1234567890/device=mobile/year=2019/month=08/day=09/
hour=20/my-delivery-stream-2019-08-09-23-55-09-a9fa96af-e4e4-409f-bac3-1f804714faaa
```

Para o particionamento dinâmico, você deve usar o seguinte formato de expressão no prefixo de bucket do S3: `!{namespace:value}`, em que o namespace pode ser `partitionKeyFromQuery`, `partitionKeyFromLambda` ou ambos. Se estiver usando análise em linha para criar as chaves de particionamento para os dados da fonte, você deverá especificar um valor de prefixo de bucket do S3 consistindo em expressões especificadas no seguinte formato: `"partitionKeyFromQuery:keyID"`. Se estiver usando função do AWS Lambda para criar as chaves de particionamento para os dados da fonte, você deverá especificar um valor de prefixo de bucket de S3 que consista em expressões especificadas no seguinte formato: `"partitionKeyFromLambda:keyID"`.

> **Note**
>
> Você também pode especificar o valor do prefixo do bucket do S3 usando o formato de estilo hive, por exemplo customer_id=! {partitionKeyFromConsulta: customer_ID}.

Para obter mais informações, consulte “Escolha o Amazon S3 para seu destino” em Criação de um stream do Amazon Firehose e prefixos personalizados para objetos do Amazon S3.

Particionamento dinâmico de dados agregados

Você pode aplicar o particionamento dinâmico aos dados agregados (por exemplo, vários eventos, logs ou registros agregados em uma única chamada de API `PutRecord` e `PutRecordBatch`), mas esses dados devem primeiro ser desagregados. Você pode desagregar seus dados ativando a desagregação de vários registros, o processo de analisar os registros no stream do Firehose e separá-los.

A desagregação de vários registros pode ser do `JSON` tipo, o que significa que a separação dos registros é baseada em objetos JSON consecutivos. A desagregação também pode ser desse tipo`Delimited`, o que significa que a separação dos registros é realizada com base em um delimitador personalizado especificado. Esse delimitador personalizado deve ser uma string codificada na base 64. Por exemplo, se quiser usar a sequência de caracteres a seguir como seu delimitador personalizado####, você deve especificá-la no formato codificado em base 64, que a traduz para. `IyMjIw==`

> **Note**
>
> Ao desagregar registros JSON, certifique-se de que sua entrada ainda seja apresentada no formato JSON compatível. Os objetos JSON devem estar em uma única linha sem delimitador ou somente delimitados por nova linha (JSONL). Uma matriz de objetos JSON não é uma entrada válida.
> Estes são exemplos de entrada correta: `{"a":1}{"a":2}` and `{"a":1}\n{"a":2}`
> Este é um exemplo da entrada incorreta: `[{"a":1}, {"a":2}]`

Com dados agregados, quando você ativa o particionamento dinâmico, o Firehose analisa os registros e procura objetos JSON válidos ou registros delimitados em cada chamada de API com base no tipo de desagregação de vários registros especificado.

> **Important**
>
> Se os dados forem agregados, o particionamento dinâmico só poderá ser aplicado se os dados primeiro forem desagregados.

> **Important**
>
> Quando você usa o recurso de transformação de dados no Firehose, a desagregação será aplicada antes da transformação de dados. Os dados que chegam ao Firehose serão processados na seguinte ordem: Desagregação → Transformação de dados via Lambda → Chaves de particionamento.

Adicionar um novo delimitador de linha ao entregar dados ao S3

Você pode ativar o New Line Delimiter para adicionar um novo delimitador de linha entre registros em objetos que são entregues ao Amazon S3. Isso pode ser útil para analisar objetos no Amazon S3. Isso também é particularmente útil quando o particionamento dinâmico é aplicado a dados agregados porque a desagregação de vários registros (que deve ser aplicada aos dados agregados antes que possam ser particionados dinamicamente) remove novas linhas dos registros como parte do processo de análise.

Como habilitar o particionamento dinâmico

Você pode configurar o particionamento dinâmico para seus streams do Firehose por meio do Amazon Data Firehose Management Console, da CLI ou das APIs.

> **Important**
>
> Você pode ativar o particionamento dinâmico somente ao criar um novo stream do Firehose. Você não pode ativar o particionamento dinâmico para um stream existente do Firehose que não tenha o particionamento dinâmico já ativado.

Para obter etapas detalhadas sobre como habilitar e configurar o particionamento dinâmico por meio do console de gerenciamento do Firehose ao criar um novo stream do Firehose, consulte Criação de um stream do Amazon Firehose. Ao concluir a tarefa de especificar o destino do seu stream do Firehose, certifique-se de seguir as etapas na seção Escolha o Amazon S3 para seu destino, pois atualmente, o particionamento dinâmico só é suportado para streams do Firehose que usam o Amazon S3 como destino.

Depois que o particionamento dinâmico em um stream ativo do Firehose estiver ativado, você poderá atualizar a configuração adicionando novas chaves de particionamento ou removendo ou atualizando as existentes e as expressões do prefixo S3. Depois de atualizado, o Firehose começa a usar as novas chaves e as novas expressões de prefixo do S3.

> **Important**
>
> Depois de habilitar o particionamento dinâmico em um stream do Firehose, ele não pode ser desativado nesse stream do Firehose.

Tratamento de erros de particionamento dinâmico

Se o Amazon Data Firehose não conseguir analisar registros de dados em seu stream do Firehose ou não conseguir extrair as chaves de particionamento especificadas ou avaliar as expressões incluídas no valor do prefixo do S3, esses registros de dados serão entregues ao prefixo do bucket de erro do S3 que você deve especificar ao criar o stream do Firehose, no qual você ativa o particionamento dinâmico. O prefixo do bucket de erro do S3 contém todos os registros que o Firehose não consegue entregar ao destino especificado do S3. Esses registros são organizados de

acordo com o tipo de erro. Junto com o registro, o objeto entregue também inclui informações sobre o erro para ajudar a entender e resolver esse erro.

Você deve especificar um prefixo de bucket de erro do S3 para um stream do Firehose se quiser ativar o particionamento dinâmico para esse stream do Firehose. Se você não quiser ativar o particionamento dinâmico para um stream do Firehose, especificar um prefixo de bucket de erro do S3 é opcional.

Armazenamento em buffer de dados e particionamento dinâmico

O Amazon Data Firehose armazena os dados de streaming recebidos em um determinado tamanho e por um determinado período de tempo antes de entregá-los aos destinos especificados. Você pode configurar o tamanho do buffer e o intervalo do buffer ao criar novos streams do Firehose ou atualizar o tamanho do buffer e o intervalo do buffer nos streams existentes do Firehose. O tamanho do buffer é medido em MBs e o intervalo do buffer é medido em segundos.

Quando o particionamento dinâmico está habilitado, o Firehose armazena internamente os registros que pertencem a uma determinada partição com base na dica de buffer configurada (tamanho e horário) antes de entregar esses registros ao seu bucket do Amazon S3. Para fornecer objetos de tamanho máximo, o Firehose usa o buffer de vários estágios internamente. Portanto, o end-to-end atraso de um lote de registros pode ser 1,5 vezes o tempo de dica de buffer configurado. Isso afeta a atualização dos dados de um stream do Firehose.

A quantidade de partições ativas é o número total de partições ativas dentro do buffer de entrega. Por exemplo, se a consulta de particionamento dinâmico monta 3 partições por segundo e você tiver uma configuração de sugestão de buffer que aciona a entrega a cada 60 segundos, então, em média, você teria 180 partições ativas. Se o Firehose não puder entregar os dados em uma partição para um destino, essa partição será considerada ativa no buffer de entrega até que possa ser entregue.

Uma nova partição é criada quando um prefixo do S3 é avaliado como um novo valor com base nos campos de dados do registro e nas expressões do prefixo do S3. Um novo buffer é criado para cada partição ativa. Cada registro subsequente com o mesmo prefixo S3 avaliado é entregue a esse buffer.

Quando o buffer atinge o limite de tamanho do buffer ou o intervalo de tempo do buffer, o Firehose cria um objeto com os dados do buffer e o entrega ao prefixo especificado do Amazon S3. Depois que o objeto é entregue, o buffer dessa partição e da própria partição são excluídos e removidos da contagem de partições ativas.

O Firehose entrega cada dado do buffer como um único objeto quando o tamanho ou o intervalo do buffer são atendidos para cada partição separadamente. Quando o número de partições ativas atinge o limite de 500 por stream do Firehose, o restante dos registros no stream do Firehose é entregue ao prefixo do bucket de erro do S3 especificado (). activePartitionExceeded Você pode usar o formulário Amazon Data Firehose Limits para solicitar um aumento dessa cota para até 5.000 partições ativas por determinado stream do Firehose. Se precisar de mais partições, você pode criar mais streams do Firehose e distribuir as partições ativas entre elas.

Convertendo seu formato de registro de entrada no Firehose

O Amazon Data Firehose pode converter o formato dos seus dados de entrada de JSON para Apache Parquet ou Apache ORC antes de armazenar os dados no Amazon S3. Parquet e ORC são formatos de dados em colunas que economizam espaço e permitem consultas mais rápidas do que com os formatos orientados a linhas, como JSON. Se você quiser converter um formato de entrada diferente de JSON, como valores separados por vírgula (CSV) ou texto estruturado, você pode usá-lo AWS Lambda para transformá-lo em JSON primeiro. Para ter mais informações, consulte Transformação de dados.

Tópicos

Requisitos de conversão de formato de registro

O Amazon Data Firehose exige os três elementos a seguir para converter o formato dos dados do seu registro:

- Um desserializador para ler o JSON dos seus dados de entrada — Você pode escolher um dos dois tipos de desserializadores: Apache Hive JSON ou OpenX JSON. SerDe SerDe

> **Note**
>
> Ao combinar vários documentos JSON no mesmo registro, certifique-se de que sua entrada ainda seja apresentada no formato JSON compatível. Uma matriz de documentos JSON não é uma entrada válida.
> Por exemplo, essa é a entrada correta: `{"a":1}{"a":2}`
> E essa é a entrada incorreta: `[{"a":1}, {"a":2}]`

- Um esquema para determinar como interpretar esses dados: use o AWS Glue para criar um esquema no AWS Glue Data Catalog. Em seguida, o Amazon Data Firehose faz referência a esse esquema e o usa para interpretar seus dados de entrada. Você pode usar o mesmo esquema para configurar o Amazon Data Firehose e seu software de análise. Para obter mais informações, consulte Preenchendo o catálogo de dados do AWS Glue no Guia do AWS Glue desenvolvedor.

  > **Note**
  >
  > O esquema criado no Catálogo de AWS Glue Dados deve corresponder à estrutura de dados de entrada. Caso contrário, os dados convertidos não conterão atributos que não estejam especificados no esquema. Se você usar JSON aninhado, use um tipo STRUCT no esquema que espelha a estrutura dos dados JSON. Veja este exemplo para saber como lidar com JSON aninhado com um tipo STRUCT.

- Um serializador para converter os dados no formato de armazenamento colunar de destino (Parquet ou ORC) — Você pode escolher um dos dois tipos de serializadores: ORC ou Parquet. SerDe SerDe

> **Important**
>
> Se você habilitar a conversão do formato de registro, não poderá definir o destino do Amazon Data Firehose como Amazon OpenSearch Service, Amazon Redshift ou Splunk. Com a conversão de formato ativada, o Amazon S3 é o único destino que você pode usar para seu stream do Firehose.

Você pode converter o formato dos seus dados mesmo se você agregar seus registros antes de enviá-los para o Amazon Data Firehose.

Escolher o desserializador JSON

Escolha o OpenX JSON SerDe se o JSON de entrada contiver carimbos de data e hora nos seguintes formatos:

- yyyy-MM-dd'T'HH:mm:ss[.S]'Z', em que a fração pode ter até 9 dígitos, por exemplo, `2017-02-07T15:13:01.39256Z`.

- yyyy-[M]M-[d]d HH:mm:ss[.S], em que a fração pode ter até 9 dígitos, por exemplo, `2017-02-07 15:13:01.14`.
- Segundos a partir do ponto zero, por exemplo, `1518033528`.
- Milissegundos a partir do ponto zero, por exemplo, `1518033528123`.
- Segundos a partir do ponto zero com ponto flutuante, por exemplo, `1518033528.123`.

O OpenX JSON SerDe pode converter pontos (`.`) em sublinhados (). _ Ele também pode converter chaves JSON para minúsculas antes de desserializá-las. Para obter mais informações sobre as opções que estão disponíveis com esse desserializador por meio do Amazon Data Firehose, consulte OpenX. JsonSerDe

Se você não tiver certeza de qual desserializador escolher, use o OpenX JSON SerDe, a menos que tenha carimbos de data e hora que ele não suporta.

Se você tiver carimbos de data e hora em formatos diferentes dos listados anteriormente, use o Apache Hive JSON. SerDe Ao escolher esse desserializador, você poderá especificar os formatos de time stamp a serem usados. Para fazer isso, siga a sintaxe do padrão de string do formato Joda Time `DateTimeFormat`. Para obter mais informações, consulte Classe DateTimeFormat.

Você também pode usar o valor especial `millis` para analisar o time stamp em milissegundos de epoch. Se você não especificar um formato, o Amazon Data Firehose usa `java.sql.Timestamp::valueOf` por padrão.

O JSON do Hive SerDe não permite o seguinte:

- Pontos (`.`) em nomes de coluna.
- Campos cujo tipo é `uniontype`.
- Campos que têm tipos numéricos no esquema, mas que são strings no JSON. Por exemplo, se o esquema for (um int) e o JSON for`{"a":"123"}`, o Hive apresentará um erro SerDe .

O Hive SerDe não converte JSON aninhado em strings. Por exemplo, se você tiver `{"a": {"inner":1}}`, ele não tratará `{"inner":1}` como uma string.

Escolher o desserializador

O serializador que você escolhe depende de suas necessidades de negócios. Para saber mais sobre as duas opções de serializador, consulte ORC SerDe e Parquet. SerDe

Converter formato de registro de entrada (Console)

Você pode ativar a conversão de formato de dados no console ao criar ou atualizar um stream do Firehose. Com a conversão de formato de dados ativada, o Amazon S3 é o único destino que você pode configurar para o stream do Firehose. Além disso, a compactação do Amazon S3 será desabilitada quando você habilitar a conversão de formato. No entanto, a compactação Snappy ocorre automaticamente como parte do processo de conversão. O formato de enquadramento do Snappy que o Amazon Data Firehose usa nesse caso é compatível com o Hadoop. Isso significa que você pode usar os resultados da compactação Snappy e executar consultas nesses dados no Athena. Para o formato de enquadramento Snappy no qual o Hadoop se baseia, consulte .java. BlockCompressorStream

Para habilitar a conversão de formato de dados para um stream de dados do Firehose

1. Faça login no e abra o console do Amazon Data Firehose em https://console.aws.amazon.com/firehose/. AWS Management Console
2. Escolha um stream do Firehose para atualizar ou crie um novo stream do Firehose seguindo as etapas em. Crie um stream do Firehose
3. Em Convert record format (Converter formato do registro), defina Record format conversion (Conversão de formato do registro) como Enabled (Habilitado).
4. Selecione o formato de saída que você deseja. Para obter mais informações sobre as duas opções, consulte Apache Parquet e Apache ORC.
5. Escolha uma AWS Glue tabela para especificar um esquema para seus registros de origem. Defina a região, o banco de dados, a tabela e a versão da tabela.

Converter formato de registro de entrada (API)

Se você quiser que o Amazon Data Firehose converta o formato dos seus dados de entrada de JSON para Parquet ou ORC, especifique o DataFormatConversionConfigurationelemento opcional em ExtendedS3 ou em ExtendedS3. DestinationConfiguration DestinationUpdate Se você especificar DataFormatConversionConfiguration, as seguintes restrições se aplicam:

- Em BufferingHints, você não pode `SizeInMBs` definir um valor menor que 64 se você habilitar a conversão do formato de registro. Além disso, quando a conversão de formato não está ativada, o valor padrão é 5. O valor se torna 128 quando você a habilita.

- Você deve definir `CompressionFormat` em ExtendedS3 DestinationConfiguration ou em ExtendedS3 como. DestinationUpdate `UNCOMPRESSED` O valor padrão para `CompressionFormat` é `UNCOMPRESSED`. Portanto, você também pode deixá-lo não especificado em DestinationConfigurationExtendedS3. Os dados ainda são compactados como parte do processo de serialização, usando a compactação Snappy, por padrão. O formato de enquadramento do Snappy que o Amazon Data Firehose usa nesse caso é compatível com o Hadoop. Isso significa que você pode usar os resultados da compactação Snappy e executar consultas nesses dados no Athena. Para o formato de enquadramento Snappy no qual o Hadoop se baseia, consulte .java. BlockCompressorStream Quando você configurar o serializador, você poderá escolher outros tipos de compactação.

Gerenciar o erro de conversão de formato de registro

Quando o Amazon Data Firehose não consegue analisar ou desserializar um registro (por exemplo, quando os dados não correspondem ao esquema), ele os grava no Amazon S3 com um prefixo de erro. Se essa gravação falhar, o Amazon Data Firehose a tentará novamente para sempre, bloqueando outras entregas. Para cada registro com falha, o Amazon Data Firehose grava um documento JSON com o seguinte esquema:

```
{
  "attemptsMade": long,
  "arrivalTimestamp": long,
  "lastErrorCode": string,
  "lastErrorMessage": string,
  "attemptEndingTimestamp": long,
  "rawData": string,
  "sequenceNumber": string,
  "subSequenceNumber": long,
  "dataCatalogTable": {
    "catalogId": string,
    "databaseName": string,
    "tableName": string,
    "region": string,
    "versionId": string,
    "catalogArn": string
  }
}
```

Exemplo de conversão do formato do registro

Para obter um exemplo de como configurar a conversão do formato de registro com AWS CloudFormation, consulte AWS::DataFirehose:: DeliveryStream.

Usar o Amazon Managed Service for Apache Flink

Com o Amazon Managed Service for Apache Flink, você pode usar Java, Scala ou SQL para processar e analisar dados em streaming. O serviço permite que você crie e execute código Java em fontes de streaming para realizar análises de séries temporais, alimentar painéis em tempo real e criar métricas em tempo real.

Para ver um exemplo de integração com o Amazon Managed Service para Apache Flink, consulte Exemplo: Escrevendo no Amazon Data Firehose.

Neste exercício, você cria um aplicativo Apache Flink que tem um stream de dados do Kinesis como fonte e um stream do Firehose como coletor. Usando o coletor, você pode conferir a saída da aplicação em um bucket do Amazon S3.

Antes de começar, configure os pré-requisitos necessários:

- Componentes do serviço gerenciado para o aplicativo Apache Flink
- Pré-requisitos para concluir o exercício

Entenda a entrega de dados do Amazon Data Firehose

Depois que os dados são enviados para o stream do Firehose, eles são automaticamente enviados para o destino que você escolher.

> **Important**
>
> Se você usar a Kinesis Producer Library (KPL) para gravar dados em um fluxo de dados do Kinesis, poderá usar agregação para combinar os registros gravados. Se você então usar esse stream de dados como fonte para seu stream do Firehose, o Amazon Data Firehose desagregará os registros antes de entregá-los ao destino. Se você configurar seu stream do Firehose para transformar os dados, o Amazon Data Firehose desagregará os registros antes de entregá-los. AWS Lambda Para obter mais informações, consulte Developing Amazon Kinesis Data Streams Producers Using the Kinesis Producer Library e Aggregation no Amazon Kinesis Data Streams Developer Guide.

Tópicos

Configurar formato de entrega de dados

Para entrega de dados ao Amazon Simple Storage Service (Amazon S3), o Firehose concatena vários registros de entrada com base na configuração de buffer do seu stream do Firehose. Depois, entrega os ao Amazon S3; como um objeto do S3;. Por padrão, o Firehose concatena dados sem delimitadores. Se quiser ter novos delimitadores de linha entre os registros, você pode adicionar

novos delimitadores de linha ativando o recurso na configuração do console Firehose ou no parâmetro da API.

Para entrega de dados ao Amazon Redshift, o Firehose primeiro entrega os dados recebidos em seu bucket do S3 no formato descrito anteriormente. Em seguida, o Firehose emite um comando do Amazon COPY Redshift para carregar os dados do seu bucket do S3 para o cluster provisionado do Amazon Redshift ou para o grupo de trabalho Amazon Redshift Serverless. Certifique-se de que, após o Amazon Data Firehose concatenar vários registros recebidos em um objeto do Amazon S3, o objeto do Amazon S3 possa ser copiado para seu cluster provisionado do Amazon Redshift ou grupo de trabalho Amazon Redshift Serverless. Para obter mais informações, consulte Parâmetros de formato de dados do comando COPY do Amazon Redshift.

Para entrega de dados para OpenSearch Service e OpenSearch Serverless, o Amazon Data Firehose armazena registros de entrada com base na configuração de buffer do seu stream Firehose. Em seguida, ele gera uma solicitação em massa de OpenSearch serviço ou OpenSearch sem servidor para indexar vários registros em seu cluster de OpenSearch serviços ou coleção sem OpenSearch servidor. Certifique-se de que seu registro esteja codificado em UTF-8 e nivelado em um objeto JSON de linha única antes de enviá-lo para o Amazon Data Firehose. Além disso, a `rest.action.multi.allow_explicit_index` opção para seu cluster de OpenSearch serviços deve ser definida como verdadeira (padrão) para receber solicitações em massa com um índice explícito definido por registro. Para obter mais informações, consulte OpenSearch Service Configure Advanced Options no Amazon OpenSearch Service Developer Guide.

Para entrega de dados ao Splunk, o Amazon Data Firehose concatena os bytes que você envia. Se você quer delimitadores em seus dados, como um caractere de nova linha, deve inseri-los. Certifique-se de que o Splunk é configurado para analisar quaisquer delimitadores.

Ao entregar dados para um endpoint HTTP de propriedade de um provedor de serviços terceirizado compatível, você pode usar o serviço Amazon Lambda integrado para criar uma função para transformar os registros recebidos no formato que é esperado ela integração do provedor de serviços. Entre em contato com o provedor de serviços terceirizado cujo endpoint HTTP você escolheu como destino para saber mais sobre o formato de registro que ele aceita.

Para entrega de dados ao Snowflake, o Amazon Data Firehose armazena internamente os dados em buffer por um segundo e usa as operações da API de streaming do Snowflake para inserir dados no Snowflake. Por padrão, os registros que você insere são liberados e confirmados na tabela do Snowflake a cada segundo. Depois de fazer a chamada de inserção, o Firehose emite uma CloudWatch métrica que mede quanto tempo levou para que os dados fossem confirmados no Snowflake. Atualmente, o Firehose suporta apenas um único item JSON como carga útil de registro

e não oferece suporte a matrizes JSON. Certifique-se de que sua carga de entrada seja um objeto JSON válido e esteja bem formada sem aspas duplas, aspas ou caracteres de escape extras.

Entenda a frequência de entrega de dados

Cada destino do Firehose tem sua própria frequência de entrega de dados. Para ter mais informações, consulte Entenda as dicas de buffer.

Lidar com falhas na entrega de dados

Cada destino do Amazon Data Firehose tem seu próprio tratamento de falhas na entrega de dados.

Amazon S3

A entrega de dados para o bucket do S3 pode apresentar falha por vários motivos. Por exemplo, o bucket pode não existir mais, a função do IAM que o Amazon Data Firehose presume pode não ter acesso ao bucket, a rede falhou ou eventos similares. Sob essas condições, o Amazon Data Firehose continua tentando novamente por até 24 horas até que a entrega seja bem-sucedida. O tempo máximo de armazenamento de dados do Amazon Data Firehose é de 24 horas. Se a entrega de dados apresentar falha por mais de 24 horas, os dados serão perdidos.

Amazon Redshift

Para um destino do Amazon Redshift, você pode especificar uma duração de nova tentativa (0 a 7200 segundos) ao criar um stream do Firehose.

A entrega de dados ao cluster provisionado do Amazon Redshift ou ao grupo de trabalho do Amazon Redshift Sem Servidor pode falhar por vários motivos. Por exemplo, você pode ter uma configuração de cluster incorreta do seu stream do Firehose, um cluster ou grupo de trabalho em manutenção ou uma falha na rede. Sob essas condições, o Amazon Data Firehose tenta novamente pelo período de tempo especificado e ignora esse lote específico de objetos do Amazon S3. As informações dos objetos ignorados são entregues no bucket do S3 como um arquivo manifesto na pasta `errors/`, que você pode usar para alocação manual. Para obter informações sobre como COPIAR manualmente os dados com arquivos de manifesto, consulte Uso de um manifesto para especificar arquivos de dados.

Amazon OpenSearch Service e OpenSearch Serverless

Para o destino OpenSearch Service e OpenSearch Serverless, você pode especificar uma duração de nova tentativa (0 a 7200 segundos) durante a criação do stream do Firehose.

A entrega de dados para seu cluster OpenSearch de serviços ou coleção OpenSearch sem servidor pode falhar por vários motivos. Por exemplo, você pode ter uma configuração incorreta de cluster de OpenSearch serviços ou coleção OpenSearch Serverless do seu stream Firehose, um cluster de OpenSearch serviços ou coleção OpenSearch Serverless em manutenção, uma falha na rede ou eventos semelhantes. Sob essas condições, o Amazon Data Firehose tenta novamente pelo período de tempo especificado e, em seguida, ignora essa solicitação de índice específica. Os documentos ignorados são entregues no bucket do S3 na pasta `AmazonOpenSearchService_failed/`, que você pode usar para alocação manual.

Para OpenSearch Serviço, cada documento tem o seguinte formato JSON:

```
{
    "attemptsMade": "(number of index requests attempted)",
    "arrivalTimestamp": "(the time when the document was received by Firehose)",
    "errorCode": "(http error code returned by OpenSearch Service)",
    "errorMessage": "(error message returned by OpenSearch Service)",
    "attemptEndingTimestamp": "(the time when Firehose stopped attempting index
 request)",
    "esDocumentId": "(intended OpenSearch Service document ID)",
    "esIndexName": "(intended OpenSearch Service index name)",
    "esTypeName": "(intended OpenSearch Service type name)",
    "rawData": "(base64-encoded document data)"
}
```

Para OpenSearch Serverless, cada documento tem o seguinte formato JSON:

```
{
    "attemptsMade": "(number of index requests attempted)",
    "arrivalTimestamp": "(the time when the document was received by Firehose)",
    "errorCode": "(http error code returned by OpenSearch Serverless)",
    "errorMessage": "(error message returned by OpenSearch Serverless)",
    "attemptEndingTimestamp": "(the time when Firehose stopped attempting index
 request)",
    "osDocumentId": "(intended OpenSearch Serverless document ID)",
    "osIndexName": "(intended OpenSearch Serverless index name)",
    "rawData": "(base64-encoded document data)"
}
```

Splunk

Quando o Amazon Data Firehose envia dados para o Splunk, ele espera por uma confirmação do Splunk. Se ocorrer um erro ou a confirmação não chegar dentro do período de tempo limite da confirmação, o Amazon Data Firehose iniciará o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para o Splunk, seja na tentativa inicial ou em uma nova tentativa, ele reinicia o contador de tempo limite de confirmação. Em seguida, ele aguarda pela chegada de um reconhecimento do Splunk. Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela confirmação até recebê-la ou até que o tempo limite da confirmação seja atingido. Se a confirmação expirar, o Amazon Data Firehose verifica se ainda há tempo no contador de novas tentativas. Se houver tempo restante, ele tentará executar novamente e repetirá a lógica até receber um reconhecimento ou determinará que o tempo de tentar novamente expirou.

Uma falha em receber uma confirmação não é o único tipo de erro de entrega de dados que pode ocorrer. Para obter informações sobre outros tipos de erros de entrega de dados, consulte Erros de entrega de dados do Splunk. Qualquer erro de entrega de dados dispara a lógica de novas tentativas se a duração é maior que 0.

Veja a seguir um exemplo de registro de erro.


```
{
  "attemptsMade": 0,
  "arrivalTimestamp": 1506035354675,
  "errorCode": "Splunk.AckTimeout",
  "errorMessage": "Did not receive an acknowledgement from HEC before the HEC
 acknowledgement timeout expired. Despite the acknowledgement timeout, it's possible
 the data was indexed successfully in Splunk. Amazon Data Firehose backs up in
 Amazon S3 data for which the acknowledgement timeout expired.",
  "attemptEndingTimestamp": 13626284715507,
  "rawData":
 "MiAyNTE2MjAyNzIyMDkgZW5pLTA1ZjMyMmQ1IDIxOC45Mi4xODguMjE0IDE3Mi4xNi4xLjE2NyAyNTIzMyAxNDMzID
  "EventId": "49577193928114147339600778471082492393164139877200035842.0"
}
```

Destino do endpoint HTTP

Quando o Amazon Data Firehose envia dados para um destino de endpoint HTTP, ele espera por uma resposta desse destino. Se ocorrer um erro ou a resposta não chegar dentro do período de tempo limite de resposta, o Amazon Data Firehose inicia o contador de duração de novas tentativas. Ele continuará tentando novamente até a duração da nova tentativa expirar. Depois disso, o Amazon Data Firehose considera isso uma falha na entrega de dados e faz o backup dos dados em seu bucket do Amazon S3.

Toda vez que o Amazon Data Firehose envia dados para um destino de endpoint HTTP, seja a tentativa inicial ou uma nova tentativa, ele reinicia o contador de tempo limite de resposta. Depois, ele espera a chegada de uma resposta do destino de endpoint HTTP. Mesmo que a duração da nova tentativa expire, o Amazon Data Firehose ainda espera pela resposta até recebê-la ou até que o tempo limite de resposta seja atingido. Se o tempo de resposta expirar, o Amazon Data Firehose verifica se ainda há tempo no contador de novas tentativas. Se restar algum tempo, ele tentará novamente e repetirá a lógica até receber uma resposta ou determinar que o período de novas tentativas expirou.

Deixar de receber confirmação não é o único tipo de erro de entrega de dados que pode ocorrer. Para obter informações sobre outros tipos de erros de entrega de dados, consulte HTTP Endpoint Data Delivery Errors

Veja a seguir um exemplo de registro de erro.

```
{
 "attemptsMade":5,
 "arrivalTimestamp":1594265943615,
 "errorCode":"HttpEndpoint.DestinationException",
 "errorMessage":"Received the following response from the endpoint destination.
 {"requestId": "109777ac-8f9b-4082-8e8d-b4f12b5fc17b", "timestamp": 1594266081268,
 "errorMessage": "Unauthorized"}",
 "attemptEndingTimestamp":1594266081318,
 "rawData":"c2FtcGxlIHJhdyBkYXRh",
 "subsequenceNumber":0,
 "dataId":"49607357361271740811418664280693044274821622880012337186.0"
}
```

Destino Snowflake

Para o destino Snowflake, ao criar um stream do Firehose, você pode especificar uma duração opcional de nova tentativa (0 a 7200 segundos). O valor padrão para a duração da nova tentativa é 60 segundos.

A entrega de dados para sua tabela do Snowflake pode falhar por vários motivos, como configuração incorreta de destino do Snowflake, interrupção do Snowflake, falha na rede etc. A política de repetição não se aplica a erros não recuperáveis. Por exemplo, se o Snowflake rejeitar sua carga JSON porque ela tinha uma coluna extra que está faltando na tabela, o Firehose não tentará entregá-la novamente. Em vez disso, ele cria um backup para todas as falhas de inserção devido a problemas de carga JSON em seu bucket de erros do S3.

Da mesma forma, se a entrega falhar devido a uma função, tabela ou banco de dados incorretos, o Firehose não tentará novamente gravar os dados em seu bucket do S3. A duração da nova tentativa só se aplica a falhas devido a um problema no serviço Snowflake, falhas transitórias na rede etc. Sob essas condições, o Firehose tenta novamente pelo período de tempo especificado antes de entregá-los ao S3. Os registros com falha são entregues na pasta snowflake-failed/, que você pode usar para preenchimento manual.

Veja a seguir um exemplo de JSON para cada registro que você entrega ao S3.

```
{
    "attemptsMade": 3,
    "arrivalTimestamp": 1594265943615,
    "errorCode": "Snowflake.InvalidColumns",
    "errorMessage": "Snowpipe Streaming does not support columns of type
 AUTOINCREMENT, IDENTITY, GEO, or columns with a default value or collation",
    "attemptEndingTimestamp": 1712937865543,
    "rawData": "c2FtcGxlIHJhdyBkYXRh"
}
```

Configurar o formato de nome de objeto do Amazon S3

Quando o Firehose entrega dados para o Amazon S3, o nome da chave do objeto S3 segue o <evaluated prefix><suffix>formato, onde o sufixo tem o formato - - - - - - - - - - <Firehose stream name><Firehose stream version><year><month><day><hour><minute><second><uuid><file extension><Firehose stream version>começa com 1 e aumenta em 1 para cada alteração de configuração do stream do Firehose. Você pode alterar as configurações de stream do Firehose (por

exemplo, o nome do bucket do S3, dicas de buffer, compactação e criptografia). Você pode fazer isso usando o console Firehose ou a operação da UpdateDestinationAPI.

Para<evaluated prefix>, o Firehose adiciona um prefixo de hora padrão no formato. YYYY/MM/dd/HH Esse prefixo cria uma hierarquia lógica no bucket, em que cada barra (/) cria um nível na hierarquia. Você pode modificar essa estrutura especificando um prefixo personalizado que inclui expressões que são avaliadas em tempo de execução. Para obter informações sobre como especificar um prefixo personalizado, consulte Prefixos personalizados para objetos do Amazon Simple Storage Service.

Por padrão, o fuso horário usado para prefixo e sufixo de hora está em UTC, mas você pode alterá-lo para um fuso horário de sua preferência. Por exemplo, para usar o horário padrão do Japão em vez do UTC, você pode configurar o fuso horário para Ásia/Tóquio na configuração de parâmetros da AWS Management Console API (). CustomTimeZone A lista a seguir contém fusos horários compatíveis com o Firehose para a configuração do prefixo S3.

Fusos horários

A seguir está uma lista de fusos horários que o Firehose suporta para a configuração do prefixo S3.

Africa

```
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmera
Africa/Bangui
Africa/Banjul
Africa/Bissau
Africa/Blantyre
Africa/Bujumbura
Africa/Cairo
Africa/Casablanca
Africa/Conakry
Africa/Dakar
Africa/Dar_es_Salaam
Africa/Djibouti
Africa/Douala
Africa/Freetown
Africa/Gaborone
Africa/Harare
```

```
Africa/Johannesburg
Africa/Kampala
Africa/Khartoum
Africa/Kigali
Africa/Kinshasa
Africa/Lagos
Africa/Libreville
Africa/Lome
Africa/Luanda
Africa/Lubumbashi
Africa/Lusaka
Africa/Malabo
Africa/Maputo
Africa/Maseru
Africa/Mbabane
Africa/Mogadishu
Africa/Monrovia
Africa/Nairobi
Africa/Ndjamena
Africa/Niamey
Africa/Nouakchott
Africa/Ouagadougou
Africa/Porto-Novo
Africa/Sao_Tome
Africa/Timbuktu
Africa/Tripoli
Africa/Tunis
Africa/Windhoek
```

America

```
America/Adak
America/Anchorage
America/Anguilla
America/Antigua
America/Aruba
America/Asuncion
America/Barbados
America/Belize
America/Bogota
America/Buenos_Aires
America/Caracas
America/Cayenne
```

```
America/Cayman
America/Chicago
America/Costa_Rica
America/Cuiaba
America/Curacao
America/Dawson_Creek
America/Denver
America/Dominica
America/Edmonton
America/El_Salvador
America/Fortaleza
America/Godthab
America/Grand_Turk
America/Grenada
America/Guadeloupe
America/Guatemala
America/Guayaquil
America/Guyana
America/Halifax
America/Havana
America/Indianapolis
America/Jamaica
America/La_Paz
America/Lima
America/Los_Angeles
America/Managua
America/Manaus
America/Martinique
America/Mazatlan
America/Mexico_City
America/Miquelon
America/Montevideo
America/Montreal
America/Montserrat
America/Nassau
America/New_York
America/Noronha
America/Panama
America/Paramaribo
America/Phoenix
America/Port_of_Spain
America/Port-au-Prince
America/Porto_Acre
America/Puerto_Rico
```

```
America/Regina
America/Rio_Branco
America/Santiago
America/Santo_Domingo
America/Sao_Paulo
America/Scoresbysund
America/St_Johns
America/St_Kitts
America/St_Lucia
America/St_Thomas
America/St_Vincent
America/Tegucigalpa
America/Thule
America/Tijuana
America/Tortola
America/Vancouver
America/Winnipeg
```

Antarctica

```
Antarctica/Casey
Antarctica/DumontDUrville
Antarctica/Mawson
Antarctica/McMurdo
Antarctica/Palmer
```

Asia

```
Asia/Aden
Asia/Almaty
Asia/Amman
Asia/Anadyr
Asia/Aqtau
Asia/Aqtobe
Asia/Ashgabat
Asia/Ashkhabad
Asia/Baghdad
Asia/Bahrain
Asia/Baku
Asia/Bangkok
Asia/Beirut
Asia/Bishkek
Asia/Brunei
```

```
Asia/Calcutta
Asia/Colombo
Asia/Dacca
Asia/Damascus
Asia/Dhaka
Asia/Dubai
Asia/Dushanbe
Asia/Hong_Kong
Asia/Irkutsk
Asia/Jakarta
Asia/Jayapura
Asia/Jerusalem
Asia/Kabul
Asia/Kamchatka
Asia/Karachi
Asia/Katmandu
Asia/Krasnoyarsk
Asia/Kuala_Lumpur
Asia/Kuwait
Asia/Macao
Asia/Magadan
Asia/Manila
Asia/Muscat
Asia/Nicosia
Asia/Novosibirsk
Asia/Phnom_Penh
Asia/Pyongyang
Asia/Qatar
Asia/Rangoon
Asia/Riyadh
Asia/Saigon
Asia/Seoul
Asia/Shanghai
Asia/Singapore
Asia/Taipei
Asia/Tashkent
Asia/Tbilisi
Asia/Tehran
Asia/Thimbu
Asia/Thimphu
Asia/Tokyo
Asia/Ujung_Pandang
Asia/Ulaanbaatar
Asia/Ulan_Bator
```

```
Asia/Vientiane
Asia/Vladivostok
Asia/Yakutsk
Asia/Yekaterinburg
Asia/Yerevan
```

Atlantic

```
Atlantic/Azores
Atlantic/Bermuda
Atlantic/Canary
Atlantic/Cape_Verde
Atlantic/Faeroe
Atlantic/Jan_Mayen
Atlantic/Reykjavik
Atlantic/South_Georgia
Atlantic/St_Helena
Atlantic/Stanley
```

Australia

```
Australia/Adelaide
Australia/Brisbane
Australia/Broken_Hill
Australia/Darwin
Australia/Hobart
Australia/Lord_Howe
Australia/Perth
Australia/Sydney
```

Europe

```
Europe/Amsterdam
Europe/Andorra
Europe/Athens
Europe/Belgrade
Europe/Berlin
Europe/Brussels
Europe/Bucharest
Europe/Budapest
Europe/Chisinau
Europe/Copenhagen
Europe/Dublin
```

```
Europe/Gibraltar
Europe/Helsinki
Europe/Istanbul
Europe/Kaliningrad
Europe/Kiev
Europe/Lisbon
Europe/London
Europe/Luxembourg
Europe/Madrid
Europe/Malta
Europe/Minsk
Europe/Monaco
Europe/Moscow
Europe/Oslo
Europe/Paris
Europe/Prague
Europe/Riga
Europe/Rome
Europe/Samara
Europe/Simferopol
Europe/Sofia
Europe/Stockholm
Europe/Tallinn
Europe/Tirane
Europe/Vaduz
Europe/Vienna
Europe/Vilnius
Europe/Warsaw
Europe/Zurich
```

Indian

```
Indian/Antananarivo
Indian/Chagos
Indian/Christmas
Indian/Cocos
Indian/Comoro
Indian/Kerguelen
Indian/Mahe
Indian/Maldives
Indian/Mauritius
Indian/Mayotte
Indian/Reunion
```

Pacific

```
Pacific/Apia
Pacific/Auckland
Pacific/Chatham
Pacific/Easter
Pacific/Efate
Pacific/Enderbury
Pacific/Fakaofo
Pacific/Fiji
Pacific/Funafuti
Pacific/Galapagos
Pacific/Gambier
Pacific/Guadalcanal
Pacific/Guam
Pacific/Honolulu
Pacific/Kiritimati
Pacific/Kosrae
Pacific/Majuro
Pacific/Marquesas
Pacific/Nauru
Pacific/Niue
Pacific/Norfolk
Pacific/Noumea
Pacific/Pago_Pago
Pacific/Palau
Pacific/Pitcairn
Pacific/Ponape
Pacific/Port_Moresby
Pacific/Rarotonga
Pacific/Saipan
Pacific/Tahiti
Pacific/Tarawa
Pacific/Tongatapu
Pacific/Truk
Pacific/Wake
Pacific/Wallis
```

<file extension>Você não pode alterar o campo de sufixo, exceto. Quando você ativa a conversão ou a compactação do formato de dados, o Firehose anexa uma extensão de arquivo com base na configuração. A tabela a seguir explica a extensão de arquivo padrão anexada pelo Firehose:

Configuração	Extensão de arquivo
Conversão de formato de dados: Parquet	.parquete
Conversão de formato de dados: ORC	.orc
Compressão: Gzip	.gz
Compressão: Zip	.zip
Compressão: Snappy	.snappy
Compressão: Hadoop-Snappy	.hsnappy

Você também pode especificar uma extensão de arquivo de sua preferência no console ou na API do Firehose. A extensão do arquivo deve começar com um ponto (.) e pode conter caracteres permitidos: 0-9a-z! -_.*' (). A extensão do arquivo não pode exceder 128 caracteres.

> **Note**
>
> Quando você especifica uma extensão de arquivo, ela substitui a extensão de arquivo padrão que o Firehose adiciona quando a conversão ou compactação do formato de dados está ativada.

Configurar a rotação do índice para o OpenSearch Serviço

Para o destino do OpenSearch serviço, você pode especificar uma opção de rotação de índice com base no tempo a partir de uma das cinco opções a seguir: NoRotationOneHour,OneDay,OneWeek, ouOneMonth.

Dependendo da opção de rotação escolhida, o Amazon Data Firehose acrescenta uma parte da data e hora de chegada UTC ao nome do índice especificado. Ele alterna o time stamp anexado adequadamente. O exemplo a seguir mostra o nome do índice resultante em OpenSearch Serviço

para cada opção de rotação do índice, onde está o nome do índice especificado myindex e a data e hora de chegada. `2016-02-25T13:00:00Z`

RotationPeriod	IndexName
`NoRotation`	`myindex`
`OneHour`	`myindex-2016-02-25-13`
`OneDay`	`myindex-2016-02-25`
`OneWeek`	`myindex-2016-w08`
`OneMonth`	`myindex-2016-02`

> **Note**
>
> Com a opção `OneWeek`, o Data Firehose cria índices automaticamente usando o formato <ANO>-w <NÚMERO DASEMANA>(por exemplo,`2020-w33`), em que o número da semana é calculado usando o horário UTC e de acordo com as seguintes convenções dos EUA:
>
> - A semana começa no domingo
> - A primeira semana do ano é a primeira semana que contém um sábado naquele ano

Entenda a entrega em todas AWS as contas e regiões

O Amazon Data Firehose oferece suporte à entrega de dados para destinos de endpoints HTTP em todas as contas. AWS O stream do Firehose e o endpoint HTTP que você escolhe como destino podem pertencer a contas diferentes. AWS

O Amazon Data Firehose também oferece suporte à entrega de dados para destinos de endpoints HTTP em todas as regiões. AWS Você pode entregar dados de um stream do Firehose em uma AWS região para um endpoint HTTP em outra região. AWS Você também pode entregar dados de um stream do Firehose para um destino de endpoint HTTP fora das AWS regiões, por exemplo, para seu próprio servidor local, definindo a URL do endpoint HTTP como o destino desejado. Nesses cenários, taxas adicionais de transferência de dados são adicionadas aos seus custos de entrega.

Para obter mais informações, consulte a seção Transferência de dados na página "Preços sob demanda".

Registros duplicados

O Amazon Data Firehose usa at-least-once semântica para entrega de dados. Em algumas circunstâncias, como quando a entrega de dados expira, novas tentativas de entrega pelo Amazon Data Firehose podem introduzir duplicatas se a solicitação original de entrega de dados eventualmente for concluída. Isso se aplica a todos os tipos de destino compatíveis com o Amazon Data Firehose.

Pausar e retomar um stream do Firehose

Depois de configurar um stream do Firehose, os dados disponíveis na fonte do stream são continuamente entregues ao destino. Se você se deparar com situações em que o destino do fluxo esteja temporariamente indisponível (por exemplo, durante operações de manutenção planejadas), pode ser que queira pausar temporariamente a entrega de dados e continuar quando o destino estiver disponível novamente. As seções seguintes mostram como fazer isso:

> **Important**
>
> Ao usar a abordagem descrita abaixo para pausar e retomar um stream, depois de retomar o stream, você verá que poucos registros são entregues ao bucket de erros no Amazon S3, enquanto o resto do stream continua sendo entregue ao destino. Essa é uma limitação conhecida da abordagem e ocorre porque um pequeno número de registros que não puderam ser entregues anteriormente ao destino após várias tentativas é rastreado como falhado.

Entendendo como o Firehose lida com falhas de entrega

Ao configurar um stream do Firehose, para muitos destinos, como OpenSearch Splunk e endpoints HTTP, você também configura um bucket S3 em que os dados que não foram entregues podem ser copiados. Para obter mais informações sobre como o Firehose faz backup dos dados em caso de falhas nas entregas, consulte Tratamento de falhas na entrega de dados. Para obter mais informações sobre como conceder acesso aos buckets do S3 nos quais os dados que não foram entregues podem ser copiados, consulte Grant Firehose Access to an Amazon S3 Destination.

Quando o Firehose (a) falha em entregar dados ao destino do stream e (b) falha em gravar dados no bucket S3 de backup devido a entregas malsucedidas, ele efetivamente pausa a entrega do stream até que os dados possam ser entregues ao destino ou gravados no local de backup do S3.

Pausando um stream do Firehose

Para pausar a entrega do stream no Firehose, primeiro remova as permissões para que o Firehose grave no local de backup do S3 em caso de falhas nas entregas. Por exemplo, se quiser pausar o stream do Firehose com OpenSearch um destino, você pode fazer isso atualizando as permissões. Para obter mais informações, consulte Conceder acesso ao Firehose a um destino de OpenSearch serviço público.

Remova a permissão `"Effect": "Allow"` para a ação `s3:PutObject` e adicione explicitamente uma instrução que aplique a permissão `Effect": "Deny"` à ação `s3:PutObject` para o bucket do S3 usado para fazer backup de entregas com falha. Em seguida, desative o destino do stream (por exemplo, desative o OpenSearch domínio de destino) ou remova as permissões para que o Firehose grave no destino. Para atualizar as permissões para outros destinos, consulte a seção do seu destino em Controlando o acesso com o Amazon Data Firehose. Depois de concluir essas duas ações, o Firehose deixará de fornecer streams e você poderá monitorar isso usando CloudWatch métricas do Firehose.

> **Important**
>
> Ao pausar a entrega do stream no Firehose, você precisa garantir que a origem do stream (por exemplo, no Kinesis Data Streams ou no Managed Service for Kafka) esteja configurada para reter os dados até que a entrega do stream seja retomada e os dados sejam entregues ao destino. Se a fonte for DirectPut, o Firehose reterá os dados por 24 horas. Poderá ocorrer uma perda de dados se você não retomar o fluxo de entregar os dados antes da expiração do período de retenção de dados.

Retomando um stream do Firehose

Para retomar a entrega, primeiro reverta a alteração feita anteriormente para o destino do stream ativando o destino e garantindo que o Firehose tenha permissões para entregar o stream ao destino. Depois, reverta as alterações feitas anteriormente nas permissões aplicadas ao bucket do S3 de backup de entregas com falha. Remova a permissão `"Effect": "Allow"` para a ação `s3:PutObject` e remova a permissão `"Effect": "Deny"` para a ação `s3:PutObject` para o

bucket do S3 usado para backup das entregas com falha. Por fim, monitore usando CloudWatch métricas do Firehose para confirmar se o stream está sendo entregue ao destino. Para visualizar e solucionar erros, use o monitoramento do Amazon CloudWatch Logs para Firehose.

Monitoramento do Amazon Data Firehose

Você pode monitorar o Amazon Data Firehose usando os seguintes recursos:

Tópicos

Práticas recomendadas para alarmes do CloudWatch

Adicione CloudWatch alarmes para quando as seguintes métricas excederem o limite de buffer (máximo de 15 minutos):

- `DeliveryToS3.DataFreshness`
- `DeliveryToSplunk.DataFreshness`
- `DeliveryToAmazonOpenSearchService.DataFreshness`
- `DeliveryToAmazonOpenSearchServerless.DataFreshness`
- `DeliveryToHttpEndpoint.DataFreshness`

Além disso, crie alarmes com base nas expressões matemáticas de métricas a seguir.

- `IncomingBytes (Sum per 5 Minutes) / 300` se aproxima de uma porcentagem de `BytesPerSecondLimit`.
- `IncomingRecords (Sum per 5 Minutes) / 300` se aproxima de uma porcentagem de `RecordsPerSecondLimit`.
- `IncomingPutRequests (Sum per 5 Minutes) / 300` se aproxima de uma porcentagem de `PutRequestsPerSecondLimit`.

Outra métrica para a qual recomendamos um alarme é `ThrottledRecords`.

Para obter informações sobre solução de problemas quando os alarmes vá para o `ALARM` estado, consulte Solução de problemas.

Monitorando o Amazon Data Firehose usando métricas CloudWatch

> **Important**
>
> Certifique-se de ativar os alarmes em todas as CloudWatch métricas pertencentes ao seu destino para identificar erros em tempo hábil.

O Amazon Data Firehose se integra às CloudWatch métricas da Amazon para que você possa coletar, visualizar e analisar CloudWatch métricas para seus streams do Firehose. Por exemplo, você pode monitorar as `IncomingRecords` métricas `IncomingBytes` e para acompanhar os dados ingeridos no Amazon Data Firehose pelos produtores de dados.

O Amazon Data Firehose coleta e publica CloudWatch métricas a cada minuto. Porém, se ocorrerem surtos de dados recebidos apenas por alguns segundos, eles podem não ser totalmente capturados ou visíveis nas métricas de um minuto. Isso ocorre porque CloudWatch as métricas são agregadas do Amazon Data Firehose em intervalos de um minuto.

As métricas coletadas para os streams do Firehose são gratuitas. Para obter informações sobre as métricas do Kinesis Agent, consulte Monitorar a integridade do Kinesis Agent.

Tópicos

- Métricas de criptografia do lado do servidor (SSE) CloudWatch
- Dimensões do Amazon Data Firehose
- Métricas de uso do Amazon Data Firehose

Métricas de particionamento CloudWatch dinâmico

Se o particionamento dinâmico estiver ativado, o namespace AWS/Firehose incluirá as seguintes métricas.

Métrica	Descrição
`ActivePartitionsLimit`	O número máximo de partições ativas que um stream do Firehose processa antes de enviar dados para o bucket de erros. Unidades: contagem
`PartitionCount`	O número de partições que estão sendo processadas, em outras palavras, a contagem de partições ativas. Esse número varia entre 1 e o limite de contagem de partições que é 500 (padrão). Unidades: contagem
`PartitionCountExceeded`	Essa métrica indica se você está excedendo o limite de contagem de partições. Ela emite 1 ou 0 dependendo do limite ser violado ou não.
`JQProcessing.Duration`	Retorna a quantidade de tempo necessária para executar a expressão JQ na função JQ do Lambda. Unidade: milissegundos
`PerPartitionThroughput`	Indica o throughput que está sendo processado por partição. Essa métrica permite monitorar o throughput por partição. Unidades: StandardUnit. BytesSecond

Métrica	Descrição
`DeliveryToS3.ObjectCount`	Indica o número de objetos que estão sendo entregues ao bucket do S3. Unidades: contagem

CloudWatch Métricas de entrega de dados

O namespace `AWS/Firehose` inclui as métricas de nível do serviço a seguir. Se você observar pequenas quedas na média para `BackupToS3.Success`, `DeliveryToS3.Success`, `DeliveryToSplunk.Success`, `DeliveryToAmazonOpenSearchService.Success` ou`DeliveryToRedshift.Success`, isso não indica que há perda de dados. O Amazon Data Firehose repete erros de entrega e não avança até que os registros sejam entregues com sucesso no destino configurado ou no bucket S3 de backup.

Tópicos

- Entrega ao OpenSearch serviço
- Entrega sem OpenSearch servidor
- Entrega ao Amazon Redshift
- Entrega ao Amazon S3
- Entrega para Snowflake
- Entrega ao Splunk
- Entrega para endpoints HTTP

Entrega ao OpenSearch serviço

Métrica	Descrição
`DeliveryToAmazonOpenSearchService.Bytes`	O número de bytes indexados ao OpenSearch Serviço durante o período especificado. Unidade: bytes

Métrica	Descrição
DeliveryToAmazonOpenSearchService.DataFreshness	A era (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro com mais de essa idade foi entregue ao OpenSearch Serviço. Unidades: segundos
DeliveryToAmazonOpenSearchService.Records	O número de registros indexados ao OpenSearch Serviço durante o período especificado. Unidades: contagem
DeliveryToAmazonOpenSearchService.Success	A soma dos registros indexados com êxito sobre a soma de registros que foram tentados.
DeliveryToS3.Bytes	O número de bytes entregues ao Amazon S3 no período especificado. O Amazon Data Firehose emite essa métrica somente quando você ativa o backup de todos os documentos. Unidades: contagem
DeliveryToS3.DataFreshness	A era (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que esse foi enviado para o bucket do S3. O Amazon Data Firehose emite essa métrica somente quando você ativa o backup de todos os documentos. Unidades: segundos
DeliveryToS3.Records	O número de registros entregues ao Amazon S3 no período especificado. O Amazon Data Firehose emite essa métrica somente quando você ativa o backup de todos os documentos. Unidades: contagem

Métrica	Descrição
DeliveryToS3.Success	A soma de comandos put do Amazon S3 bem-sucedidos sobre a soma de todos os comandos put do Amazon S3. O Amazon Data Firehose sempre emite essa métrica, independentemente de o backup estar habilitado somente para documentos com falha ou para todos os documentos.
DeliveryToAmazonOpenSearchService.AuthFailure	Erro de autenticação/autorização. Verifique a política de cluster e as permissões de perfil de OS/ES. 0 indica que não há nenhum problema. 1 indica falha de autenticação.
DeliveryToAmazonOpenSearchService.DeliveryRejected	Erro de entrega rejeitada. Verifique a política de cluster e as permissões de perfil de OS/ES. 0 indica que não há nenhum problema. 1 indica uma falha de entrega.

Entrega sem OpenSearch servidor

Métrica	Descrição
DeliveryToAmazonOpenSearchServerless.Bytes	O número de bytes indexados ao OpenSearch Serverless durante o período especificado. Unidade: bytes
DeliveryToAmazonOpenSearchServerless.DataFreshness	A era (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro com mais de essa idade foi entregue ao OpenSearch Serverless. Unidades: segundos

Métrica	Descrição
DeliveryToAmazonOpenSearchServerless.Records	O número de registros indexados ao OpenSearch Serverless durante o período especificado. Unidades: contagem
DeliveryToAmazonOpenSearchServerless.Success	A soma dos registros indexados com êxito sobre a soma de registros que foram tentados.
DeliveryToS3.Bytes	O número de bytes entregues ao Amazon S3 no período especificado. O Amazon Data Firehose emite essa métrica somente quando você ativa o backup de todos os documentos. Unidades: contagem
DeliveryToS3.DataFreshness	A era (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que esse foi enviado para o bucket do S3. O Amazon Data Firehose emite essa métrica somente quando você ativa o backup de todos os documentos. Unidades: segundos
DeliveryToS3.Records	O número de registros entregues ao Amazon S3 no período especificado. O Amazon Data Firehose emite essa métrica somente quando você ativa o backup de todos os documentos. Unidades: contagem

Métrica	Descrição
DeliveryToS3.Success	A soma de comandos put do Amazon S3 bem-sucedidos sobre a soma de todos os comandos put do Amazon S3. O Amazon Data Firehose sempre emite essa métrica, independentemente de o backup estar habilitado ou somente para documentos com falha ou para todos os documentos.
DeliveryToAmazonOpenSearchServerless.AuthFailure	Erro de autenticação/autorização. Verifique a política de cluster e as permissões de perfil de OS/ES. 0 indica que não há nenhum problema. 1 indica falha de autenticação.
DeliveryToAmazonOpenSearchServerless.DeliveryRejected	Erro de entrega rejeitada. Verifique a política de cluster e as permissões de perfil de OS/ES. 0 indica que não há nenhum problema. 1 indica uma falha de entrega.

Entrega ao Amazon Redshift

Métrica	Descrição
DeliveryToRedshift.Bytes	O número de bytes copiados para o Amazon Redshift no período especificado. Unidades: contagem
DeliveryToRedshift.Records	O número de registros copiados para o Amazon Redshift no período especificado. Unidades: contagem
DeliveryToRedshift.Success	A soma de comandos COPY do Amazon Redshift bem-sucedidos sobre a soma de todos os comandos COPY do Amazon Redshift.

Métrica	Descrição
DeliveryToS3.Bytes	O número de bytes entregues ao Amazon S3 no período especificado. Unidade: bytes
DeliveryToS3.DataFreshness	A era (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que esse foi enviado para o bucket do S3. Unidades: segundos
DeliveryToS3.Records	O número de registros entregues ao Amazon S3 no período especificado. Unidades: contagem
DeliveryToS3.Success	A soma de comandos put do Amazon S3 bem-sucedidos sobre a soma de todos os comandos put do Amazon S3.
BackupToS3.Bytes	O número de bytes entregues ao Amazon S3 para backup no período especificado. O Amazon Data Firehose emite essa métrica quando o backup para o Amazon S3 está ativado. Unidades: contagem
BackupToS3.DataFreshness	Idade (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que isso foi entregue ao bucket do Amazon S3 para backup. O Amazon Data Firehose emite essa métrica quando o backup para o Amazon S3 está ativado. Unidades: segundos

Métrica	Descrição
`BackupToS3.Records`	O número de registros entregues ao Amazon S3 para backup no período especificado. O Amazon Data Firehose emite essa métrica quando o backup para o Amazon S3 está ativado. Unidades: contagem
`BackupToS3.Success`	Soma dos comandos put bem-sucedidos do Amazon S3 para backup sobre a soma de todos os comandos put de backup do Amazon S3. O Amazon Data Firehose emite essa métrica quando o backup para o Amazon S3 está ativado.

Entrega ao Amazon S3

As métricas na tabela a seguir estão relacionadas à entrega para o Amazon S3 quando ele é o principal destino do stream do Firehose.

Métrica	Descrição
`DeliveryToS3.Bytes`	O número de bytes entregues ao Amazon S3 no período especificado. Unidade: bytes
`DeliveryToS3.DataFreshness`	A era (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que esse foi enviado para o bucket do S3. Unidades: segundos
`DeliveryToS3.Records`	O número de registros entregues ao Amazon S3 no período especificado. Unidades: contagem

Métrica	Descrição
`DeliveryToS3.Success`	A soma de comandos put do Amazon S3 bem-sucedidos sobre a soma de todos os comandos put do Amazon S3.
`BackupToS3.Bytes`	O número de bytes entregues ao Amazon S3 para backup no período especificado. O Amazon Data Firehose emite essa métrica quando o backup está ativado (o que só é possível quando a transformação de dados também está ativada). Unidades: contagem
`BackupToS3.DataFreshness`	Idade (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que isso foi entregue ao bucket do Amazon S3 para backup. O Amazon Data Firehose emite essa métrica quando o backup está ativado (o que só é possível quando a transformação de dados também está ativada). Unidades: segundos
`BackupToS3.Records`	O número de registros entregues ao Amazon S3 para backup no período especificado. O Amazon Data Firehose emite essa métrica quando o backup está ativado (o que só é possível quando a transformação de dados também está ativada). Unidades: contagem
`BackupToS3.Success`	Soma dos comandos put bem-sucedidos do Amazon S3 para backup sobre a soma de todos os comandos put de backup do Amazon S3. O Amazon Data Firehose emite essa métrica quando o backup está ativado (o que só é possível quando a transformação de dados também está ativada).

Entrega para Snowflake

Métrica	Descrição
`DeliveryToSnowflake.Bytes`	O número de bytes entregues ao Snowflake durante o período especificado. Unidade: bytes
`DeliveryToSnowflake.DataFreshness`	Idade (de entrar no Firehose até agora) do disco mais antigo do Firehose. Qualquer registro com mais de essa idade foi entregue ao Snowflake. Observe que pode levar alguns segundos para confirmar os dados no Snowflake após a chamada de inserção do Firehose ser bem-sucedida. Para saber o tempo necessário para confirmar os dados no Snowflake, consulte a `DeliveryToSnowflake.DataCommitLatency` métrica. Unidades: segundos
`DeliveryToSnowflake.DataCommitLatency`	O tempo necessário para que os dados sejam confirmados no Snowflake depois que o Firehose inseriu os registros com sucesso. Unidades: segundos
`DeliveryToSnowflake.Records`	O número de registros entregues ao Snowflake durante o período especificado. Unidades: contagem
`DeliveryToSnowflake.Success`	A soma das chamadas de inserção bem-sucedidas feitas para o Snowflake sobre a soma das chamadas de inserção que foram tentadas.
`DeliveryToS3.Bytes`	O número de bytes entregues ao Amazon S3 no período especificado. Essa métrica só está disponível quando a entrega para o Snowflake falha e o Firehose tenta fazer backup dos dados com falha no S3.

Métrica	Descrição
	Unidade: bytes
DeliveryToS3.Records	O número de registros entregues ao Amazon S3 no período especificado. Essa métrica só está disponível quando a entrega para o Snowflake falha e o Firehose tenta fazer backup dos dados com falha no S3. Unidades: contagem
DeliveryToS3.Success	A soma de comandos put do Amazon S3 bem-sucedidos sobre a soma de todos os comandos put do Amazon S3. Essa métrica só está disponível quando a entrega para o Snowflake falha e o Firehose tenta fazer backup dos dados com falha no S3.
BackupToS3.DataFreshness	Idade (de Firehose até agora) do registro mais antigo do Firehose. Qualquer registro anterior a essa idade é copiado para o bucket do Amazon S3. Essa métrica está disponível quando o stream do Firehose está configurado para fazer backup de todos os dados. Unidades: segundos
BackupToS3.Records	O número de registros entregues ao Amazon S3 para backup no período especificado. Essa métrica está disponível quando o stream do Firehose está configurado para fazer backup de todos os dados. Unidades: contagem
BackupToS3.Bytes	O número de bytes entregues ao Amazon S3 para backup no período especificado. Essa métrica está disponível quando o stream do Firehose está configurado para fazer backup de todos os dados. Unidades: contagem

Métrica	Descrição
`BackupToS3.Success`	A soma dos comandos put bem-sucedidos do Amazon S3 para backup sobre a soma de todos os comandos de backup put do Amazon S3. O Firehose emite essa métrica quando o stream do Firehose é configurado para fazer backup de todos os dados.

Entrega ao Splunk

Métrica	Descrição
`DeliveryToSplunk.Bytes`	O número de bytes enviados para o Splunk ao longo do período de tempo especificado. Unidade: bytes
`DeliveryToSplunk.DataAckLatency`	A duração aproximada necessária para receber uma confirmação da Splunk após o Amazon Data Firehose enviar os dados. A tendência de aumento ou diminuição dessa métrica é mais útil que o valor aproximado absoluto. As tendências de aumento podem indicar taxas de confirmação e indexação mais lentas dos indexadores da Splunk. Unidades: segundos
`DeliveryToSplunk.DataFreshness`	Idade (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que esse foi enviado para o Splunk. Unidades: segundos
`DeliveryToSplunk.Records`	O número de registros enviados para o Splunk ao longo do período de tempo especificado. Unidades: contagem

Métrica	Descrição
DeliveryToSplunk.Success	A soma dos registros indexados com êxito sobre a soma de registros que foram tentados.
DeliveryToS3.Success	A soma de comandos put do Amazon S3 bem-sucedidos sobre a soma de todos os comandos put do Amazon S3. Essa métrica é emitida quando o backup no Amazon S3 está habilitado.
BackupToS3.Bytes	O número de bytes entregues ao Amazon S3 para backup no período especificado. O Amazon Data Firehose emite essa métrica quando o stream do Firehose é configurado para fazer backup de todos os documentos. Unidades: contagem
BackupToS3.DataFreshness	Idade (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que isso foi entregue ao bucket do Amazon S3 para backup. O Amazon Data Firehose emite essa métrica quando o stream do Firehose é configurado para fazer backup de todos os documentos. Unidades: segundos
BackupToS3.Records	O número de registros entregues ao Amazon S3 para backup no período especificado. O Amazon Data Firehose emite essa métrica quando o stream do Firehose é configurado para fazer backup de todos os documentos. Unidades: contagem

Métrica	Descrição
`BackupToS3.Success`	Soma dos comandos put bem-sucedidos do Amazon S3 para backup sobre a soma de todos os comandos put de backup do Amazon S3. O Amazon Data Firehose emite essa métrica quando o stream do Firehose é configurado para fazer backup de todos os documentos.

Entrega para endpoints HTTP

Métrica	Descrição
`DeliveryToHttpEndpoint.Bytes`	O número de bytes entregues com sucesso ao endpoint HTTP. Unidade: bytes
`DeliveryToHttpEndpoint.Records`	O número de registros entregues com sucesso ao endpoint HTTP. Unidade: contagens
`DeliveryToHttpEndpoint.DataFreshness`	Idade do registro mais antigo no Amazon Data Firehose. Unidades: segundos
`DeliveryToHttpEndpoint.Success`	A soma de todas as solicitações de entrega de dados bem-sucedidas para o endpoint HTTP Unidades: contagem
`DeliveryToHttpEndpoint.ProcessedBytes`	O número de tentativas de bytes processados, incluindo as novas tentativas.
`DeliveryToHttpEndpoint.ProcessedRecords`	O número de tentativas de registro, incluindo as novas tentativas.

Métricas de ingestão de dados

Tópicos

Ingestão de dados por meio do Kinesis Data Streams

Métrica	Descrição
`DataReadFromKinesisStream.Bytes`	Quando a fonte de dados é um fluxo de dados do Kinesis, essa métrica indica o número de bytes desse fluxo de dados que foram lidos. Esse número inclui releituras devido a failovers. Unidade: bytes
`DataReadFromKinesisStream.Records`	Quando a fonte de dados é um fluxo de dados do Kinesis, essa métrica indica o número de registros desse fluxo de dados que foram lidos. Esse número inclui releituras devido a failovers. Unidades: contagem
`ThrottledDescribeStream`	O número total de vezes que a operação `DescribeStream` será limitada quando a fonte de dados for um streaming de dados do Kinesis. Unidades: contagem
`ThrottledGetRecords`	O número total de vezes que a operação `GetRecords` será limitada quando a fonte de dados for um streaming de dados do Kinesis. Unidades: contagem

Métrica	Descrição
`ThrottledGetShardIterator`	O número total de vezes que a operação `GetShardIterator` será limitada quando a fonte de dados for um streaming de dados do Kinesis. Unidades: contagem

Ingestão de dados por meio de Direct PUT

Métrica	Descrição
`BackupToS3.Bytes`	O número de bytes entregues ao Amazon S3 para backup no período especificado. O Amazon Data Firehose emite essa métrica quando a transformação de dados está habilitada para destinos do Amazon S3 ou do Amazon Redshift. Unidade: bytes
`BackupToS3.DataFreshness`	Idade (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que isso foi entregue ao bucket do Amazon S3 para backup. O Amazon Data Firehose emite essa métrica quando a transformação de dados está habilitada para destinos do Amazon S3 ou do Amazon Redshift. Unidades: segundos
`BackupToS3.Records`	O número de registros entregues ao Amazon S3 para backup no período especificado. O Amazon Data Firehose emite essa métrica quando a transformação de dados está habilitada para destinos do Amazon S3 ou do Amazon Redshift. Unidades: contagem

Métrica	Descrição
`BackupToS3.Success`	Soma dos comandos put bem-sucedidos do Amazon S3 para backup sobre a soma de todos os comandos put de backup do Amazon S3. O Amazon Data Firehose emite essa métrica quando a transformação de dados está habilitada para destinos do Amazon S3 ou do Amazon Redshift.
`BytesPerSecondLimit`	O número máximo atual de bytes por segundo que um stream do Firehose pode ingerir antes da limitação. Para solicitar um aumento desse limite, acesse o AWS Support Center e escolha Criar caso e, depois, escolha Aumento de limite de serviço.
`DataReadFromKinesisStream.Bytes`	Quando a fonte de dados é um fluxo de dados do Kinesis, essa métrica indica o número de bytes desse fluxo de dados que foram lidos. Esse número inclui releituras devido a failovers. Unidade: bytes
`DataReadFromKinesisStream.Records`	Quando a fonte de dados é um fluxo de dados do Kinesis, essa métrica indica o número de registros desse fluxo de dados que foram lidos. Esse número inclui releituras devido a failovers. Unidades: contagem
`DeliveryToAmazonOpenSearchService.Bytes`	O número de bytes indexados ao OpenSearch Serviço durante o período especificado. Unidade: bytes

Métrica	Descrição
DeliveryToAmazonOpenSearchService.DataFreshness	A era (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro com mais de essa idade foi entregue ao OpenSearch Serviço. Unidades: segundos
DeliveryToAmazonOpenSearchService.Records	O número de registros indexados ao OpenSearch Serviço durante o período especificado. Unidades: contagem
DeliveryToAmazonOpenSearchService.Success	A soma dos registros indexados com êxito sobre a soma de registros que foram tentados.
DeliveryToRedshift.Bytes	O número de bytes copiados para o Amazon Redshift no período especificado. Unidade: bytes
DeliveryToRedshift.Records	O número de registros copiados para o Amazon Redshift no período especificado. Unidades: contagem
DeliveryToRedshift.Success	A soma de comandos COPY do Amazon Redshift bem-sucedidos sobre a soma de todos os comandos COPY do Amazon Redshift.
DeliveryToS3.Bytes	O número de bytes entregues ao Amazon S3 no período especificado. Unidade: bytes

Métrica	Descrição
DeliveryToS3.DataFreshness	A era (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que esse foi enviado para o bucket do S3. Unidades: segundos
DeliveryToS3.Records	O número de registros entregues ao Amazon S3 no período especificado. Unidades: contagem
DeliveryToS3.Success	A soma de comandos put do Amazon S3 bem-sucedidos sobre a soma de todos os comandos put do Amazon S3.
DeliveryToSplunk.Bytes	O número de bytes enviados para o Splunk ao longo do período de tempo especificado. Unidade: bytes
DeliveryToSplunk.DataAckLatency	A duração aproximada necessária para receber uma confirmação da Splunk após o Amazon Data Firehose enviar os dados. A tendência de aumento ou diminuição dessa métrica é mais útil que o valor aproximado absoluto. As tendências de aumento podem indicar taxas de confirmação e indexação mais lentas dos indexadores da Splunk. Unidades: segundos
DeliveryToSplunk.DataFreshness	Idade (de entrar no Amazon Data Firehose até agora) do registro mais antigo no Amazon Data Firehose. Qualquer registro mais antigo que esse foi enviado para o Splunk. Unidades: segundos

Métrica	Descrição
DeliveryToSplunk.Records	O número de registros enviados para o Splunk ao longo do período de tempo especificado. Unidades: contagem
DeliveryToSplunk.Success	A soma dos registros indexados com êxito sobre a soma de registros que foram tentados.
IncomingBytes	O número de bytes ingeridos com sucesso no stream do Firehose durante o período especificado. A ingestão de dados pode ser reduzida quando excede um dos limites de fluxo do Firehose. Os dados com controle de utilizaçã o não serão contabilizados em IncomingBytes .. Unidade: bytes
IncomingPutRequests	O número de PutRecordBatch solicitações bem-suced idas PutRecord e em um período de tempo especificado. Unidades: contagem
IncomingRecords	O número de registros ingeridos com sucesso no stream do Firehose durante o período especificado. A ingestão de dados pode ser reduzida quando excede um dos limites de fluxo do Firehose. Os dados com controle de utilização não serão contabilizados em IncomingR ecords .. Unidades: contagem
KinesisMillisBehindLatest	Quando a fonte de dados for um streaming de dados do Kinesis, esta métrica indica o número de milissegu ndos de diferença que o último registro de leitura está em relação ao registro mais recente nesse streaming. Unidade: milissegundo

Métrica	Descrição
`RecordsPerSecondLimit`	O número máximo atual de registros por segundo que um stream do Firehose pode ingerir antes da limitação. Unidades: contagem
`ThrottledRecords`	O número de registros que foram limitados porque a ingestão de dados excedeu um dos limites de fluxo do Firehose. Unidades: contagem

Ingestão de dados do MSK

Métrica	Descrição
`DataReadFromSource .Records`	O número de registros lidos do Kafka Topic de origem. Unidades: contagem
`DataReadFromSource.Bytes`	O número de bytes lidos do Kafka Topic de origem. Unidade: bytes
`SourceThrottled.Delay`	A quantidade de tempo que o cluster do Kafka de origem demora para retornar os registros do Kafka Topic de origem. Unidade: milissegundos
`BytesPerSecondLimit`	Limite atual do throughput com o qual o Firehose lerá em cada partição do Kafka Topic de origem. Unidades: bytes/segundo
`KafkaOffsetLag`	A diferença entre o maior deslocamento de registro que o Firehose leu no Kafka Topic de origem e o maior

Métrica	Descrição
	deslocamento de registro disponível do Kafka Topic de origem. Unidades: contagem
`FailedValidation.Records`	O número de registros que não foram aprovados na validação de registros. Unidades: contagem
`FailedValidation.Bytes`	O número de bytes que não foram aprovados na validação de registros. Unidade: bytes
`DataReadFromSource.Backpressured`	Indica que um stream do Firehose está atrasado na leitura de registros da partição de origem porque BytesPerSecondLimit cada partição excedeu ou porque o fluxo normal de entrega está lento ou parou. Unidade: booleano

Métricas em nível de API CloudWatch

O namespace `AWS/Firehose` inclui as métricas de nível da API a seguir.

Métrica	Descrição
`DescribeDeliveryStream.Latency`	O tempo gasto por operação `DescribeDeliveryStream` , medido ao longo do período de tempo especificado. Unidade: milissegundos
`DescribeDeliveryStream.Requests`	O número total de solicitações `DescribeDeliveryStream` .

Métrica	Descrição
	Unidades: contagem
`ListDeliveryStreams.Latency`	O tempo gasto por operação `ListDeliveryStream` , medido ao longo do período de tempo especificado. Unidade: milissegundos
`ListDeliveryStreams.Requests`	O número total de solicitações `ListFirehose` . Unidades: contagem
`PutRecord.Bytes`	O número de bytes colocados no stream do Firehose usando `PutRecord` durante o período de tempo especificado. Unidade: bytes
`PutRecord.Latency`	O tempo gasto por operação `PutRecord` , medido ao longo do período de tempo especificado. Unidade: milissegundos
`PutRecord.Requests`	O número total de solicitações `PutRecord` , igual ao número total de registros das operações `PutRecord` . Unidades: contagem
`PutRecordBatch.Bytes`	O número de bytes colocados no stream do Firehose usando `PutRecordBatch` durante o período de tempo especificado. Unidade: bytes
`PutRecordBatch.Latency`	O tempo gasto por operação `PutRecordBatch` , medido ao longo do período de tempo especificado. Unidade: milissegundos

Métrica	Descrição
PutRecordBatch.Records	O número total de registros das operações PutRecordBatch . Unidades: contagem
PutRecordBatch.Requests	O número total de solicitações PutRecordBatch . Unidades: contagem
PutRequestsPerSecondLimit	O número máximo de solicitações put por segundo que um stream do Firehose pode processar antes da limitação. Esse número inclui PutRecordBatch solicitações PutRecord e solicitações. Unidades: contagem
ThrottledDescribeStream	O número total de vezes que a operação DescribeStream será limitada quando a fonte de dados for um streaming de dados do Kinesis. Unidades: contagem
ThrottledGetRecords	O número total de vezes que a operação GetRecords será limitada quando a fonte de dados for um streaming de dados do Kinesis. Unidades: contagem
ThrottledGetShardIterator	O número total de vezes que a operação GetShardIterator será limitada quando a fonte de dados for um streaming de dados do Kinesis. Unidades: contagem
UpdateDeliveryStream.Latency	O tempo gasto por operação UpdateDeliveryStream , medido ao longo do período de tempo especificado. Unidade: milissegundos

Métrica	Descrição
`UpdateDeliveryStream.Requests`	O número total de solicitações `UpdateDeliveryStream` . Unidades: contagem

CloudWatch Métricas de transformação de dados

Se a transformação de dados com o Lambda estiver habilitada, o `AWS/Firehose` namespace incluirá as métricas a seguir.

Métrica	Descrição
`ExecuteProcessing.Duration`	O tempo necessário para cada invocação da função Lambda realizada pelo Firehose. Unidade: milissegundos
`ExecuteProcessing.Success`	A soma das invocações bem-sucedidas da função Lambda sobre a soma do total de invocações da função do Lambda.
`SucceedProcessing.Records`	O número de registros processados com sucesso no período especificado. Unidades: contagem
`SucceedProcessing.Bytes`	O número de bytes processados com sucesso no período especificado. Unidade: bytes

CloudWatch Métricas de descompressão de registros

Se a descompactação estiver ativada para entrega de CloudWatch registros, o `AWS/Firehose` namespace incluirá as seguintes métricas.

Métrica	Descrição
`OutputDecompressedBytes.Success`	Dados descompactados em bytes com sucesso Unidade: bytes
`OutputDecompressedBytes.Failed`	Falha nos dados descompactados em bytes Unidade: bytes
`OutputDecompressedRecords.Success`	Número de registros descompactados com sucesso Unidades: contagem
`OutputDecompressedRecords.Failed`	Número de registros descompactados com falha Unidades: contagem

CloudWatch Métricas de conversão de formato

Se a conversão de formato estiver desativada, o namespace `AWS/Firehose` incluirá as seguintes métricas.

Métrica	Descrição
`SucceedConversion.Records`	O número de registros convertidos com êxito. Unidades: contagem
`SucceedConversion.Bytes`	O tamanho dos registros convertidos com êxito. Unidade: bytes
`FailedConversion.Records`	O número de registros que não puderam ser convertidos. Unidades: contagem
`FailedConversion.Bytes`	O tamanho dos registros que não puderam ser convertidos.

Métrica	Descrição
	Unidade: bytes

Métricas de criptografia do lado do servidor (SSE) CloudWatch

O namespace do `AWS/Firehose` inclui as seguintes métricas relacionadas à SSE.

Métrica	Descrição
`KMSKeyAccessDenied`	O número de vezes que o serviço encontra um `KMSAccessDeniedException` para o stream Firehose. Unidades: contagem
`KMSKeyDisabled`	O número de vezes que o serviço encontra um `KMSDisabledException` para o stream Firehose. Unidades: contagem
`KMSKeyInvalidState`	O número de vezes que o serviço encontra um `KMSInvalidStateException` para o stream do Firehose. Unidades: contagem
`KMSKeyNotFound`	O número de vezes que o serviço encontra um `KMSNotFoundException` para o stream do Firehose. Unidades: contagem

Dimensões do Amazon Data Firehose

Para filtrar métricas por stream do Firehose, use a `DeliveryStreamName` dimensão.

Métricas de uso do Amazon Data Firehose

Você pode usar métricas de CloudWatch uso para dar visibilidade ao uso dos recursos da sua conta. Use essas métricas para visualizar seu uso atual do serviço em CloudWatch gráficos e painéis.

As métricas de uso da cota de serviço estão no namespace AWS/Usage e são coletadas a cada minuto.

Atualmente, o único nome de métrica CloudWatch publicado nesse namespace é. `ResourceCount` Essa métrica é publicada com as dimensões `Service`, `Class`, `Type` e `Resource`.

Métrica	Descrição
`ResourceCount`	O número dos recursos especificados em execução em sua conta. Os recursos são definidos pelas dimensões associadas à métrica. A estatística mais útil para essa métrica é MAXIMUM, que representa o número máximo de recursos usados durante o período de um minuto.

As dimensões a seguir são usadas para refinar as métricas de uso publicadas pelo Amazon Data Firehose.

Dimensão	Descrição
`Service`	O nome do AWS serviço que contém o recurso. Para as métricas de uso do Amazon Data Firehose, o valor dessa dimensão é. `Firehose`
`Class`	A classe do recurso sob acompanhamento. As métricas de uso da API Amazon Data Firehose usam essa dimensão com um valor de. None
`Type`	O tipo de recurso que está sendo acompanhado. Atualmente, quando a dimensão Service é `Firehose`, o único valor válido para Type é `Resource`.

Dimensão	Descrição
`Resource`	O nome do AWS recurso. Atualmente, quando a dimensão Service é `Firehose`, o único valor válido para Resource é `DeliveryStreams` .

Acessando CloudWatch métricas para o Amazon Data Firehose

Você pode monitorar as métricas do Amazon Data Firehose usando o CloudWatch console, a linha de comando ou CloudWatch a API. Os procedimentos a seguir mostram como acessar as métricas usando os seguintes métodos:

Para acessar métricas usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Na barra de navegação, escolha uma região.
3. No painel de navegação, selecione Métricas.
4. Escolha o namespace Firehose.
5. Escolha Firehose stream Metrics ou Firehose Metrics.
6. Selecione uma métrica a ser adicionada ao gráfico.

Para acessar métricas usando o AWS CLI

Use as métricas e get-metric-statisticsos comandos da lista.

```
aws cloudwatch list-metrics --namespace "AWS/Firehose"
```

```
aws cloudwatch get-metric-statistics --namespace "AWS/Firehose" \
--metric-name DescribeDeliveryStream.Latency --statistics Average --period 3600 \
--start-time 2017-06-01T00:00:00Z --end-time 2017-06-30T00:00:00Z
```

Monitorando o Amazon Data Firehose usando registros CloudWatch

O Amazon Data Firehose se integra ao Amazon CloudWatch Logs para que você possa visualizar os registros de erros específicos quando a invocação do Lambda para transformação ou entrega de dados falhar. Você pode ativar o registro de erros do Amazon Data Firehose ao criar seu stream do Firehose.

Se você ativar o registro de erros do Amazon Data Firehose no console do Amazon Data Firehose, um grupo de logs e os fluxos de log correspondentes serão criados para o stream do Firehose em seu nome. O formato do nome do grupo de registros é`/aws/kinesisfirehose/`*`delivery-stream-name`*, onde *`delivery-stream-name`* está o nome do stream correspondente do Firehose. `DestinationDelivery`é o fluxo de log criado e usado para registrar quaisquer erros relacionados à entrega ao destino principal. Outro fluxo de logs denominado `BackupDelivery` só é criado se o backup do S3 estiver habilitado para o destino. O fluxo de logs de `BackupDelivery` é usado para registrar em log quaisquer erros relacionados à entrega ao backup do S3.

Por exemplo, se você criar um stream do Firehose "MyStream" com o Amazon Redshift como destino e ativar o registro de erros do Amazon Data Firehose, o seguinte será criado em seu nome: um grupo de logs `aws/kinesisfirehose/MyStream` chamado e dois streams de log chamados e. `DestinationDelivery` BackupDelivery Neste exemplo, `DestinationDelivery` será usado para registrar em log quaisquer erros relacionados à entrega ao destino do Amazon Redshift e também ao destino intermediário do S3. `BackupDelivery`, caso o backup do S3 esteja habilitado, será usado para registrar em log quaisquer erros relacionados à entrega ao bucket de backup do S3.

Você pode ativar o registro de erros do Amazon Data Firehose por meio da AWS CLI API ou AWS CloudFormation usando a `CloudWatchLoggingOptions` configuração. Para fazer isso, crie um grupo de logs e um fluxo de log com antecedência. Recomendamos reservar esse grupo e fluxo de log exclusivamente para o registro de erros do Amazon Data Firehose. Além disso, garanta que a política do IAM; associada tenha a permissão `"logs:putLogEvents"`. Para ter mais informações, consulte Controle de acesso com o Amazon Data Firehose.

Observe que o Amazon Data Firehose não garante que todos os registros de erros de entrega sejam enviados para a CloudWatch Logs. Em circunstâncias em que a taxa de falha na entrega é alta, o Amazon Data Firehose coleta amostras dos registros de erros de entrega antes de enviá-los para a CloudWatch Logs.

Há uma cobrança nominal pelos registros de erros enviados para o CloudWatch Logs. Para obter mais informações, consulte Amazon CloudWatch Pricing.

Conteúdo

Erros de entrega de dados

A seguir está uma lista de mensagens e códigos de erro de entrega de dados para cada destino do Amazon Data Firehose. Cada mensagem de erro também descreve a ação apropriada a ser executada para resolver o problema.

Erros

Erros de entrega de dados do Amazon S3

O Amazon Data Firehose pode enviar os seguintes erros relacionados ao Amazon S3 para o Logs. CloudWatch

Código de erro	Mensagem de erros e informações
`S3.KMS.NotFoundException`	“A AWS KMS chave fornecida não foi encontrada. Se você estiver usando o que acredita ser uma AWS KMS chave válida com a função correta, verifique se há algum problema com a conta à qual a AWS KMS chave está anexada.”
`S3.KMS.RequestLimitExceeded`	"O limite de solicitações do KMS por segundo foi excedido durante a tentativa de criptografar objetos do S3. Aumente o limite de solicitação por segundo." Para obter mais informações, consulte Limites no Guia do desenvolvedor do AWS Key Management Service .
`S3.AccessDenied`	"Acesso negado. Certifique-se de que a política de confiança para a função do IAM fornecida permita que o Amazon Data Firehose assuma a função, e que a política de acesso permita o acesso ao bucket do S3.”
`S3.AccountProblem`	“Há um problema com sua AWS conta que impede que a operação seja concluída com êxito. Entre em contato com o AWS Support."
`S3.AllAccessDisabled`	"O acesso à conta fornecida foi desabilitado. Entre em contato com AWS o Support.”
`S3.InvalidPayer`	"O acesso à conta fornecida foi desabilitado. Entre em contato com AWS o Support.”
`S3.NotSignedUp`	"A conta não está cadastrada no Amazon S3. Cadastre a conta ou use outra conta."
`S3.NoSuchBucket`	"O bucket especificado não existe. Crie o bucket ou use um bucket existente."
`S3.MethodNotAllowed`	"O método especificado não é permitido neste recurso. Modifique a política do bucket para que sejam concedidas as permissões corretas de operação do Amazon S3."
`InternalError`	"Ocorreu um erro durante a entrega dos dados. A entrega será repetida; se o erro persistir, ele será reportado AWS para resolução.”

Código de erro	Mensagem de erros e informações
`S3.KMS.KeyDisabled`	"A chave do KMS fornecida está desabilitada. Habilite a chave ou use uma chave diferente."
`S3.KMS.InvalidStateException`	"A chave do KMS fornecida está em um estado inválido. Use uma chave diferente."
`KMS.InvalidStateException`	"A chave do KMS fornecida está em um estado inválido. Use uma chave diferente."
`KMS.DisabledException`	"A chave do KMS fornecida está desabilitada. Corrija a chave ou use uma chave diferente."
`S3.SlowDown`	"A taxa de solicitação de put para o bucket especificado era muito alta. Aumente o tamanho do buffer de stream do Firehose ou reduza as solicitações de venda de outros aplicativos.”
`S3.SubscriptionRequired`	"O acesso foi negado ao chamar o S3. Certifique-se de que o perfil do IAM e a chave do KMS (se fornecida) passadas tenham a assinatura do Amazon S3."
`S3.InvalidToken`	"O formato do token fornecido é malformado ou é inválido por algum outro motivo. Verifique as credenciais fornecidas."
`S3.KMS.KeyNotConfigured`	"Chave do KMS não configurada. Configure seu MasterKey ID KMS ou desative a criptografia para seu bucket do S3.”
`S3.KMS.AsymmetricCMKNotSupported`	"O Amazon S3 só é compatível com CMKs simétricas. Não é possível usar uma CMK assimétrica para criptografar dados no Amazon S3. Para obter o tipo da sua CMK, use a DescribeKey operação KMS.”
`S3.IllegalLocationConstraintException`	"Atualmente, o Firehose usa o endpoint global do s3 para entrega de dados ao bucket do s3 configurado. A região do bucket do s3 configurado não é compatível com o endpoint global do s3. Crie um stream do Firehose na mesma região do bucket s3 ou use o bucket s3 na região que oferece suporte ao endpoint global.”

Código de erro	Mensagem de erros e informações
`S3.InvalidPrefixConfigurationException`	"O prefixo do s3 personalizado usado para a avaliação do timestamp é inválido. Verifique se o prefixo s3 contém expressões válidas para a data e hora atuais do ano."
`DataFormatConversion.MalformedData`	"Caractere ilegal encontrado entre tokens."

Erros de entrega de dados do Amazon Redshift

O Amazon Data Firehose pode enviar os seguintes erros relacionados ao Amazon Redshift para o Logs. CloudWatch

Código de erro	Mensagem de erros e informações
`Redshift.TableNotFound`	"A tabela em que os dados foram carregados não foi encontrada. Verifique se a tabela especificada existe." Não foi possível encontrar no Amazon Redshift a tabela de destinos na qual os dados do S3 deveriam ser copiados. Observe que o Amazon Data Firehose não cria a tabela do Amazon Redshift se ela não existir.
`Redshift.SyntaxError`	"O comando COPY contém um erro de sintaxe. Repita o comando."
`Redshift.AuthenticationFailed`	"Falha na autenticação do nome de usuário e da senha fornecidos. Forneça um nome de usuário e uma senha válidos."
`Redshift.AccessDenied`	"Acesso negado. Certifique-se de que a política de confiança para a função do IAM fornecida permita que o Amazon Data Firehose assuma a função.”

Código de erro	Mensagem de erros e informações
`Redshift.S3BucketAccessDenied`	"O comando COPY não pôde acessar o bucket do S3. Verifique se a política de acesso padrão do perfil do IAM fornecido permite o acesso ao bucket do S3."
`Redshift.DataLoadFailed`	"Falha no carregamento de dados na tabela. Verifique se há detalhes na tabela de sistema STL_LOAD_ERRORS."
`Redshift.ColumnNotFound`	"Uma coluna do comando COPY não consta na tabela. Especifique um nome de coluna válido."
`Redshift.DatabaseNotFound`	"Não foi possível encontrar o banco de dados especificado na configuração do destino ou na URL de JDBC do Amazon Redshift. Especifique um nome de banco de dados válido."
`Redshift.IncorrectCopyOptions`	"Foram fornecidas opções de COPY redundantes ou conflitantes. Algumas opções não são compatíveis em determinadas combinações. Verifique a referência do comando COPY para obter mais informações." Para obter mais informações sobre as visualizações do Amazon Redshift, consulte Comando COPY do Amazon Redshift no Guia do desenvolvedor de bando de dados do Amazon.
`Redshift.MissingColumn`	"Há uma coluna no esquema de tabelas definida como NOT NULL sem o valor DEFAULT e não incluída na lista de colunas. Exclua essa coluna, certifique-se de que os dados carregados sempre forneçam um valor para essa coluna ou adicione um valor padrão ao esquema do Amazon Redshift para essa tabela."
`Redshift.ConnectionFailed`	"Falha na conexão com o cluster do Amazon Redshift especificado. Certifique-se de que as configurações de segurança permitam conexões do Amazon Data Firehose, que o cluster ou banco de dados especificado na configuração de destino do Amazon Redshift ou na URL do JDBC esteja correto e que o cluster esteja disponível.”
`Redshift.ColumnMismatch`	"O número de jsonpaths no comando COPY e o número de colunas na tabela de destinos devem ser compatíveis. Repita o comando."

Código de erro	Mensagem de erros e informações
`Redshift.IncorrectOrMissingRegion`	"O Amazon Redshift tentou usar o endpoint de região incorreto para acessar o bucket do S3. Especifique um valor de região correto nas opções do comando COPY ou certifique-se de que o bucket do S3 esteja na mesma região do banco de dados do Amazon Redshift."
`Redshift.IncorrectJsonPathsFile`	"O arquivo jsonpaths fornecido não está em um formato JSON compatível. Repita o comando."
`Redshift.MissingS3File`	"Um ou mais arquivos do S3 exigidos pelo Amazon Redshift foram removidos do bucket do S3. Verifique as politicas do bucket do S3 para remover qualquer exclusão automática dos arquivos do S3."
`Redshift.InsufficientPrivilege`	"O usuário não tem permissões para carregar dados na tabela. Verifique se as permissões de usuário do Amazon Redshift incluam o privilégio de INSERT."
`Redshift.ReadOnlyCluster`	"Não é possível executar a consulta porque o sistema está no modo de redimensionamento. Tente executar a consulta novamente mais tarde."
`Redshift.DiskFull`	"Não foi possível carregar os dados porque o disco está cheio. Aumente a capacidade do cluster do Amazon Redshift ou exclua os dados não utilizados para liberar espaço em disco."
`InternalError`	"Ocorreu um erro durante a entrega dos dados. A entrega será repetida; se o erro persistir, ele será reportado AWS para resolução.”
`Redshift.ArgumentNotSupported`	"O comando COPY contém opções não compatíveis."
`Redshift.AnalyzeTableAccessDenied`	"Acesso negado. A cópia do S3 para o Redshift está falhando porque a análise da tabela só pode ser feita pelo proprietário da tabela ou do banco de dados."

Código de erro	Mensagem de erros e informações
`Redshift.SchemaNotFound`	“O esquema especificado na configuração de destino DataTableName do Amazon Redshift não foi encontrado. Especifique um nome de esquema válido."
`Redshift.ColumnSpecifiedMoreThanOnce`	"A mesma coluna está especificada mais de uma vez na lista de colunas. Certifique-se de que as colunas duplicadas sejam removidas."
`Redshift.ColumnNotNullWithoutDefault`	"Uma coluna não nula sem DEFAULT não está incluída na lista de colunas. Certifique-se de que essas colunas estejam incluídas na lista de colunas."
`Redshift.IncorrectBucketRegion`	"O Redshift tentou usar um bucket em uma região diferente da região do cluster. Especifique um bucket na mesma região da região do cluster."
`Redshift.S3SlowDown`	"Alta taxa de solicitação ao S3. Reduza a taxa para evitar que o controle de utilização seja aplicado."
`Redshift.InvalidCopyOptionForJson`	"Use um caminho automático do S3 ou válido para copyOption do json."
`Redshift.InvalidCopyOptionJSONPathFormat`	"Falha de COPY com erro \"Formato JSONPath inválido. O índice da matriz está fora do intervalo\". Corrija a expressão JSONPath."
`Redshift.InvalidCopyOptionRBACAclNotAllowed`	"Falha de COPY com erro\" Não é possível usar a estrutura de acl RBAC enquanto a propagação de permissões não está habilitada. \"

Código de erro	Mensagem de erros e informações
`Redshift.DiskSpaceQuotaExceeded`	"Transação abortada porque a cota de espaço em disco foi excedida. Libere espaço em disco ou solicite uma cota maior para os esquemas."
`Redshift.ConnectionsLimitExceeded`	"Limite de conexão excedido para o usuário."
`Redshift.SslNotSupported`	"A conexão com o cluster especificado do Amazon Redshift falhou porque o servidor não é compatível com SSL. Verifique as configurações do cluster."
`Redshift.HoseNotFound`	"O hose foi excluído. Verifique o status do hose."
`Redshift.Delimiter`	"O delimitador copyOptions no copyCommand é inválido. Certifique-se de que ele seja um caractere único."
`Redshift.QueryCancelled`	"O usuário cancelou a operação COPY."
`Redshift.CompressionMismatch`	"O hose está configurado com UNCOMPRESSED, mas CopyOption inclui um formato de compactação."
`Redshift.EncryptionCredentials`	"A opção ENCRYPTED requer credenciais no formato: 'aws_iam_role=...;master_symmetric_key=...' or 'aws_access_key_id=...;aws_secret_access_key=...[;token=...];master_symmetric_key=...'"
`Redshift.InvalidCopyOptions`	"Opções de configuração de COPY inválidas."
`Redshift.InvalidMessageFormat`	"O comando Copy contém um caractere inválido."

Código de erro	Mensagem de erros e informações
`Redshift.TransactionIdLimitReached`	"Limite de ID de transação atingido."
`Redshift.DestinationRemoved`	"Verifique se o destino do redshift existe e está configurado corretamente na configuração do Firehose."
`Redshift.OutOfMemory`	"O cluster do Redshift está ficando sem memória. Certifique-se de que o cluster tenha capacidade suficiente."
`Redshift.CannotForkProcess`	"O cluster do Redshift está ficando sem memória. Certifique-se de que o cluster tenha capacidade suficiente."
`Redshift.SslFailure`	"A conexão SSL foi fechada durante o handshake."
`Redshift.Resize`	"O Amazon Redshift está redimensionando o cluster. O Firehose não poderá fornecer dados enquanto o cluster estiver sendo redimensionado."
`Redshift.ImproperQualifiedName`	"O nome qualificado é inadequado (muitos nomes pontilhados)."
`Redshift.InvalidJsonPathFormat`	"Formato de JSONPath inválido."
`Redshift.TooManyConnectionsException`	"Muitas conexões com o Redshift."

Código de erro	Mensagem de erros e informações
`Redshift.PSQLException`	“PS QlException observado a partir do Redshift.”
`Redshift.DuplicateSecondsSpecification`	"Especificação de segundos duplicados no formato de data/hora."
`Redshift.RelationCouldNotBeOpened`	"Foi encontrado um erro do Redshift, não foi possível abrir a relação. Verifique os logs do Redshift para o banco de dados especificado."
`Redshift.TooManyClients`	"Exceção do Redshift devido a muitos clientes encontrados. Revisite o máximo de conexões com o banco de dados se houver vários produtores gravando nele simultaneamente."

Erros de entrega de dados do Snowflake

O Firehose pode enviar os seguintes erros relacionados ao Snowflake para o Logs. CloudWatch

Código de erro	Mensagem de erros e informações
`Snowflake.InvalidUrl`	“O Firehose não consegue se conectar ao Snowflake. Certifique-se de que o URL da conta esteja especificado corretamente na configuração de destino do Snowflake.”
`Snowflake.InvalidUser`	“O Firehose não consegue se conectar ao Snowflake. Certifique-se de que o usuário esteja especificado corretamente na configuração de destino do Snowflake.”
`Snowflake.InvalidRole`	“A função especificada do floco de neve não existe ou não está autorizada. Certifique-se de que a função seja concedida ao usuário especificado”
`Snowflake.InvalidTable`	“A tabela fornecida não existe ou não está autorizada”

Código de erro	Mensagem de erros e informações
`Snowflake.InvalidSchema`	“O esquema fornecido não existe ou não está autorizado”
`Snowflake.InvalidDatabase`	“O banco de dados fornecido não existe ou não está autorizado”
`Snowflake.InvalidPrivateKeyOrPassphrase`	“A chave privada ou frase secreta especificada não é válida. Observe que a chave privada fornecida deve ser uma chave privada PEM RSA válida.”
`Snowflake.MissingColumns`	“A solicitação de inserção foi rejeitada devido à falta de colunas na carga de entrada. Certifique-se de que os valores sejam especificados para todas as colunas não anuláveis”
`Snowflake.ExtraColumns`	“A solicitação de inserção foi rejeitada devido a colunas extras. As colunas não presentes na tabela não devem ser especificadas”
`Snowflake.InvalidInput`	“A entrega falhou devido a um formato de entrada inválido. Certifique-se de que a carga de entrada fornecida esteja no formato JSON aceitável”
`Snowflake.IncorrectValue`	“A entrega falhou devido ao tipo de dados incorreto na carga de entrada. Certifique-se de que os valores JSON especificados na carga de entrada estejam de acordo com o tipo de dados declarado na definição da tabela do Snowflake.”

Erros de entrega de dados do Splunk

O Amazon Data Firehose pode enviar os seguintes erros relacionados ao Splunk para o Logs. CloudWatch

Código de erro	Mensagem de erros e informações
`Splunk.ProxyWithoutStickySessions`	“Se você tiver um proxy (ELB ou outro) entre o Amazon Data Firehose e o nó HEC, você deve habilitar sessões fixas para oferecer suporte aos HEC ACKs.”
`Splunk.DisabledToken`	"O token do HEC está desativado. Ativar o token para permitir a entrega dados para o Splunk."
`Splunk.InvalidToken`	"O token do HEC é inválido. Atualize o Amazon Data Firehose com um token HEC válido.”
`Splunk.InvalidDataFormat`	"Os dados não estão formatados corretamente. Para ver como formatar os dados corretamente para endpoints de HEC de eventos ou brutos, consulte Dados de evento do Splunk."
`Splunk.InvalidIndex`	"O token do HEC ou a entrada está configurada com um índice inválido. Verifique a configuração do índice e tente novamente."
`Splunk.ServerError`	"A entrega de dados ao Splunk falhou devido a um erro de servidor do nó do HEC. O Amazon Data Firehose tentará enviar os dados novamente se a duração da nova tentativa em seu Amazon Data Firehose for maior que 0. Se todas as novas tentativas falharem, o Amazon Data Firehose fará o backup dos dados no Amazon S3.”
`Splunk.DisabledAck`	"Confirmação do indexador está desativada para o token do HEC. Ative a confirmação do indexador e tente novamente. Para obter mais informações, consulte Ativar confirmação do indexador."
`Splunk.AckTimeout`	"Não recebeu uma confirmação do HEC antes do tempo limite de confirmação do HEC expirar. Embora o tempo limite para confirmação tenha expirado, é possível que os dados tenham sido anexados com sucesso no Splunk. O Amazon Data Firehose faz backup nos dados do Amazon S3 para os quais o tempo limite de confirmação expirou.”
`Splunk.MaxRetriesFailed`	"Falha para entregar dados para o Splunk ou para receber confirmação. Verifique o status do HEC e tente novamente."

Código de erro	Mensagem de erros e informações
`Splunk.ConnectionTimeout`	"A conexão com o Splunk expirou. Isso pode ser um erro temporário e a será feita uma nova tentativa de solicitação. O Amazon Data Firehose faz backup dos dados no Amazon S3 se todas as novas tentativas falharem.”
`Splunk.InvalidEndpoint`	"Não foi possível se conectar ao endpoint do HEC. Certifique-se de que o URL do endpoint HEC seja válido e possa ser acessado pelo Amazon Data Firehose.”
`Splunk.ConnectionClosed`	"Não foi possível enviar dados para a Splunk devido a uma falha de conexão. Isso pode ser um erro temporário. Aumentar a duração da nova tentativa na configuração do Amazon Data Firehose pode evitar essas falhas transitórias.”
`Splunk.SSLUnverified`	"Não foi possível se conectar ao endpoint do HEC. O host não corresponde ao certificado fornecido pelo peer. Certifique-se de que o certificado e o host são válidos."
`Splunk.SSLHandshake`	"Não foi possível se conectar ao endpoint do HEC. Certifique-se de que o certificado e o host são válidos."
`Splunk.URLNotFound`	"A URL solicitada não foi encontrada no servidor do Splunk. Verifique o cluster do Splunk e certifique-se de que ele esteja configurado corretamente."
`Splunk.ServerError.ContentTooLarge`	"A entrega de dados para a Splunk falhou devido a um erro no servidor com uma mensagem statusCode: 413: a solicitação que seu cliente enviou era muito grande. Consulte a documentação do splunk para configurar max_content_length."
`Splunk.IndexerBusy`	"A entrega de dados ao Splunk falhou devido a um erro de servidor do nó do HEC. Certifique-se de que o endpoint do HEC ou do Elastic Load Balancer esteja acessível e íntegro."

Código de erro	Mensagem de erros e informações
`Splunk.ConnectionRecycled`	"A conexão do Firehose com o Splunk foi reciclada. A entrega será repetida."
`Splunk.AcknowledgementsDisabled`	"Não foi possível obter confirmações no POST. Certifique-se de que as confirmações estejam habilitadas no endpoint do HEC."
`Splunk.InvalidHecResponseCharacter`	"Caracteres inválidos encontrados na resposta do HEC, certifique-se de verificar o serviço e a configuração do HEC."

ElasticSearch Erros de entrega de dados

O Amazon Data Firehose pode enviar os seguintes ElasticSearch erros para CloudWatch o Logs.

Código de erro	Mensagem de erros e informações
`ES.AccessDenied`	"Acesso negado. Certifique-se de que o perfil do IAM associado fornecido ao firehose não seja excluído."
`ES.ResourceNotFound`	“O domínio AWS Elasticsearch especificado não existe.”

Erros de entrega de dados de endpoint HTTPS

O Amazon Data Firehose pode enviar os seguintes erros relacionados ao endpoint HTTP para o Logs. CloudWatch Se nenhum desses erros corresponder ao problema que você está tendo, o erro padrão é o seguinte: "Ocorreu um erro interno ao tentar entregar os dados. A entrega será repetida; se o erro persistir, ele será reportado AWS para resolução.”

Código de erro	Mensagem de erros e informações
`HttpEndpoint.RequestTimeout`	O tempo limite para entrega expirou antes que uma resposta fosse recebida e ela será repetida. Se esse erro persistir, entre em contato com a equipe de atendimento do AWS Firehose.
`HttpEndpoint.ResponseTooLarge`	"A resposta recebida do endpoint é muito grande. Entre em contato com o proprietário do endpoint para resolver esse problema."
`HttpEndpoint.InvalidResponseFromDestination`	"A resposta recebida do endpoint especificado é inválida. Entre em contato com o proprietário do endpoint para resolver o problema."
`HttpEndpoint.DestinationException`	"A resposta a seguir foi recebida do destino do endpoint."
`HttpEndpoint.ConnectionFailed`	"Não foi possível se conectar ao endpoint. Entre em contato com o proprietário do endpoint para resolver esse problema."
`HttpEndpoint.ConnectionReset`	"Não é possível manter a conexão com o endpoint. Entre em contato com o proprietário do endpoint para resolver esse problema."
`HttpEndpoint.ConnectionReset`	"Problemas em manter a conexão com o endpoint. Entre em contato com o proprietário do endpoint."
`HttpEndpoint.ResponseReasonPhraseExceededLimit`	"A frase do motivo da resposta recebida do endpoint excede o limite configurado de 64 caracteres."

Código de erro	Mensagem de erros e informações
`HttpEndpoint.InvalidResponseFromDestination`	"A resposta recebida do endpoint é inválida. Consulte Solução de problemas de endpoints HTTP na documentação do Firehose para obter mais informações. Motivo: "
`HttpEndpoint.DestinationException`	"A entrega para o endpoint não teve sucesso. Consulte Solução de problemas de endpoints HTTP na documentação do Firehose para obter mais informações. Resposta recebida com código de status "
`HttpEndpoint.InvalidStatusCode`	"Recebeu um código de status de resposta inválido."
`HttpEndpoint.SSLHandshakeFailure`	"Não foi possível concluir um handshake do SSL com o endpoint. Entre em contato com o proprietário do endpoint para resolver esse problema."
`HttpEndpoint.SSLHandshakeFailure`	"Não foi possível concluir um handshake do SSL com o endpoint. Entre em contato com o proprietário do endpoint para resolver esse problema."
`HttpEndpoint.SSLFailure`	"Não foi possível concluir o handshake do TLS com o endpoint. Entre em contato com o proprietário do endpoint para resolver esse problema."
`HttpEndpoint.SSLHandshakeCertificatePathFailure`	"Não foi possível concluir um handshake do SSL com o endpoint devido ao caminho de certificação inválido. Entre em contato com o proprietário do endpoint para resolver esse problema."

Código de erro	Mensagem de erros e informações
`HttpEndpoint.SSLHandshakeCertificatePathValidationFailure`	"Não foi possível concluir um handshake do SSL com o endpoint devido à falha na validação do caminho de certificação. Entre em contato com o proprietário do endpoint para resolver esse problema."
`HttpEndpoint.MakeRequestFailure.IllegalUriException`	“falha na HttpEndpoint solicitação devido a uma entrada inválida no URI. Certifique-se de que todos os caracteres no URI de entrada sejam válidos."
`HttpEndpoint.MakeRequestFailure.IllegalCharacterInHeaderValue`	“falha na HttpEndpoint solicitação devido a um erro de resposta ilegal. Caractere ilegal '\n' no valor do cabeçalho."
`HttpEndpoint.IllegalResponseFailure`	“falha na HttpEndpoint solicitação devido a um erro de resposta ilegal. A mensagem HTTP não deve conter mais de um cabeçalho Content-Type."
`HttpEndpoint.IllegalMessageStart`	“falha na HttpEndpoint solicitação devido a um erro de resposta ilegal. Início de mensagem HTTP ilegal. Consulte Solução de problemas de endpoints HTTP na documentação do Firehose para obter mais informações."

Erros de entrega de dados do Amazon OpenSearch Service

Para o destino do OpenSearch serviço, o Amazon Data Firehose envia erros para CloudWatch os registros à medida que eles são retornados pelo OpenSearch serviço.

Além dos erros que podem retornar dos OpenSearch clusters, você pode encontrar os dois erros a seguir:

- O erro de autenticação/autorização ocorre durante a tentativa de entregar dados ao cluster de OpenSearch serviços de destino. Isso pode acontecer devido a qualquer problema de permissão e/ou de forma intermitente quando a configuração do domínio de destino do Amazon Data Firehose OpenSearch Service é modificada. Verifique a política do cluster e as permissões do perfil.
- Os dados não puderam ser entregues ao cluster OpenSearch de serviços de destino devido a falhas de autenticação/autorização. Isso pode acontecer devido a qualquer problema de permissão e/ou de forma intermitente quando a configuração do domínio de destino do Amazon Data Firehose OpenSearch Service é modificada. Verifique a política do cluster e as permissões do perfil.

Código de erro	Mensagem de erros e informações
`OS.AccessDenied`	"Acesso negado. Certifique-se de que a política de confiança para a função do IAM fornecida permita que o Firehose assuma a função e que a política de acesso permita o acesso à API do Amazon OpenSearch Service.”
`OS.AccessDenied`	"Acesso negado. Certifique-se de que a política de confiança para a função do IAM fornecida permita que o Firehose assuma a função e que a política de acesso permita o acesso à API do Amazon OpenSearch Service.”
`OS.AccessDenied`	"Acesso negado. Certifique-se de que o perfil do IAM associado fornecido ao firehose não seja excluído."
`OS.AccessDenied`	"Acesso negado. Certifique-se de que o perfil do IAM associado fornecido ao firehose não seja excluído."

Código de erro	Mensagem de erros e informações
`OS.ResourceNotFound`	“O domínio especificado OpenSearch do Amazon Service não existe.”
`OS.ResourceNotFound`	“O domínio especificado OpenSearch do Amazon Service não existe.”
`OS.AccessDenied`	"Acesso negado. Certifique-se de que a política de confiança para a função do IAM fornecida permita que o Firehose assuma a função e que a política de acesso permita o acesso à API do Amazon OpenSearch Service.”
`OS.RequestTimeout`	“A solicitação para o cluster do Amazon OpenSearch Service ou a coleta OpenSearch sem servidor atingiu o tempo limite. Certifique-se de que o cluster ou a coleção tenha capacidade suficiente para a workload atual."
`OS.ClusterError`	“O cluster do Amazon OpenSearch Service retornou um erro não especificado.”
`OS.RequestTimeout`	“A solicitação para o cluster do Amazon OpenSearch Service atingiu o tempo limite. Certifique-se de que o cluster tenha capacidade suficiente para a workload atual."
`OS.ConnectionFailed`	“Problemas na conexão com o cluster do Amazon OpenSearch Service ou com a coleção OpenSearch Serverless. Certifique-se de que o cluster ou a coleção esteja íntegro e acessível."
`OS.ConnectionReset`	“Não é possível manter a conexão com o cluster do Amazon OpenSearch Service ou com a coleção OpenSearch Serverless. Entre em contato com o proprietário do cluster ou da coleção para resolver esse problema. "
`OS.ConnectionReset`	“Problemas em manter a conexão com o cluster do Amazon OpenSearch Service ou com a coleção OpenSearch Serverless. Certifique-se de que o cluster ou a coleção esteja íntegro e tenha capacidade suficiente para a workload atual."

Código de erro	Mensagem de erros e informações
`OS.ConnectionReset`	“Problemas em manter a conexão com o cluster do Amazon OpenSearch Service ou com a coleção OpenSearch Serverless. Certifique-se de que o cluster ou a coleção esteja íntegro e tenha capacidade suficiente para a workload atual."
`OS.AccessDenied`	"Acesso negado. Certifique-se de que a política de acesso no cluster do Amazon OpenSearch Service conceda acesso à função configurada do IAM.”
`OS.ValidationException`	“O OpenSearch cluster retornou um ESServiceException. Um dos motivos é que o cluster foi atualizado para o OS 2.x ou superior, mas a mangueira ainda tem o TypeName parâmetro configurado. Atualize a configuração da mangueira definindo TypeName a como uma string vazia ou altere o endpoint para o cluster, que suporta o parâmetro Type.”
`OS.ValidationException`	"O membro deve atender ao padrão de expressão regular: [a-z][a-z0-9\ \-]+
`OS.JsonParseException`	“O cluster Amazon OpenSearch Service retornou um JsonParse Exception. Certifique-se de que os dados inseridos sejam válidos."
`OS.AmazonOpenSearchServiceParseException`	“O cluster Amazon OpenSearch Service retornou um AmazonOpenSearchServiceParseException. Certifique-se de que os dados inseridos sejam válidos."
`OS.ExplicitIndexInBulkNotAllowed`	“Certifique-se de que rest.action.multi.allow_explicit_index esteja definido como verdadeiro no cluster do Amazon Service.” OpenSearch
`OS.ClusterError`	“O cluster do Amazon OpenSearch Service ou a coleção OpenSearch Serverless retornou um erro não especificado.”
`OS.ClusterBlockException`	“O cluster retornou um ClusterBlockException. Ele pode estar sobrecarregado."

Código de erro	Mensagem de erros e informações
`OS.InvalidARN`	“O ARN do Amazon OpenSearch Service fornecido é inválido. Verifique sua DeliveryStream configuração.”
`OS.MalformedData`	"Um ou mais registros estão malformado. Certifique-se de que cada registro seja um único objeto JSON válido e não contenha novas linhas."
`OS.InternalError`	"Ocorreu um erro durante a tentativa de entrega dos dados. A entrega será repetida; se o erro persistir, ele será reportado AWS para resolução .”
`OS.AliasWithMultipleIndicesNotAllowed`	"O alias tem mais de um índice associado a ele. Certifique-se de que o alias tenha apenas um índice associado a ele."
`OS.UnsupportedVersion`	“Atualmente, o Amazon OpenSearch Service 6.0 não é compatível com o Amazon Data Firehose. Entre em contato com o AWS Support para obter mais informações.”
`OS.CharConversionException`	"Um ou mais registros continham um caractere inválido."
`OS.InvalidDomainNameLength`	"O tamanho do nome de domínio não está dentro dos limites válidos do sistema operacional."
`OS.VPCDomainNotSupported`	“No momento, os domínios do Amazon OpenSearch Service em VPCs não são suportados.”
`OS.ConnectionError`	“O servidor http fechou a conexão inesperadamente. Verifique a integridade do cluster Amazon OpenSearch Service ou da coleção OpenSearch Serverless.”
`OS.LargeFieldData`	“O cluster do Amazon OpenSearch Service cancelou a solicitação, pois continha dados de campo maiores do que o permitido.”

Código de erro	Mensagem de erros e informações
`OS.BadGateway`	“O cluster do Amazon OpenSearch Service ou a coleção OpenSearch Serverless abortou a solicitação com uma resposta: 502 Bad Gateway.”
`OS.ServiceException`	“Erro recebido do cluster do Amazon OpenSearch Service ou da coleção OpenSearch Serverless. Se o cluster ou a coleção estiver por trás de uma VPC, garanta que a configuração da rede permita a conectividade."
`OS.GatewayTimeout`	“O Firehose encontrou erros de tempo limite ao se conectar ao cluster Amazon OpenSearch Service ou à coleção OpenSearch Serverless.”
`OS.MalformedData`	“O Amazon Data Firehose não oferece suporte aos comandos da API Amazon OpenSearch Service Bulk dentro do registro do Firehose.”
`OS.ResponseEntryCountMismatch`	"A resposta da API Bulk continha mais entradas do que o número de registros enviados. Certifique-se de que cada registro contenha somente um objeto JSON e de que não haja novas linhas."

Erros de invocação do Lambda

O Amazon Data Firehose pode enviar os seguintes erros de invocação do Lambda para o Logs. CloudWatch

Código de erro	Mensagem de erros e informações
`Lambda.AssumeRoleAccessDenied`	"Acesso negado. Certifique-se de que a política de confiança para a função do IAM fornecida permita que o Amazon Data Firehose assuma a função.”
`Lambda.InvokeAccessDenied`	"Acesso negado. Certifique-se de que a política de acesso permita o acesso à função do Lambda."
`Lambda.JsonProcessingException`	"Ocorreu um erro ao analisar os registros retornados da função do Lambda. Certifique-se de que os registros retornados sigam o modelo de status exigido pelo Amazon Data Firehose.”

Código de erro	Mensagem de erros e informações
	Para ter mais informações, consulte Transformação de dados e modelo de status.
`Lambda.InvokeLimitExceeded`	"O limite de execução simultânea do Lambda foi excedido. Aumente o limite de execução simultânea." Para obter mais informações, consulte Limites do AWS Lambda no Guia do desenvolvedor do AWS Lambda .
`Lambda.DuplicatedRecordId`	"Foram retornados vários registros com o mesmo ID. Certifique-se de que a função do Lambda retorne IDs de registro exclusivos para cada registro." Para ter mais informações, consulte Transformação de dados e modelo de status.
`Lambda.MissingRecordId`	"Um ou mais IDs de registro foram retornados. Certifique-se de que a função do Lambda retorne todos os IDs de registro recebidos." Para ter mais informações, consulte Transformação de dados e modelo de status.
`Lambda.ResourceNotFound`	"A função do Lambda especificada não existe. Use uma função existente."
`Lambda.InvalidSubnetIDException`	"O ID de sub-rede especificado na configuração da VPC da função do Lambda é inválido. Verifique se o ID de sub-rede é válido."
`Lambda.InvalidSecurityGroupIDException`	"O ID de grupo de segurança especificado na configuração da VPC da função do Lambda é inválido. Verifique se o ID de security group é válido."

Código de erro	Mensagem de erros e informações
`Lambda.SubnetIPAddressLimitReachedException`	“não AWS Lambda foi possível configurar o acesso à VPC para a função Lambda porque uma ou mais sub-redes configuradas não têm endereços IP disponíveis. Aumente o limite de endereços IP." Para obter mais informações consulte Limites da Amazon VPC: VPP e sub-redes no Manual do usuário da Amazon VPC.
`Lambda.ENILimitReachedException`	“não AWS Lambda foi possível criar uma interface de rede elástica (ENI) na VPC, especificada como parte da configuração da função Lambda, porque o limite para interfaces de rede foi atingido. Aumente o limite de interfaces de rede." Para obter mais informações, consulte Limites da VPC: interfaces de rede no Guia do usuário da Amazon VPC.
`Lambda.FunctionTimedOut`	A invocação da função do Lambda atingiu o tempo limite. Aumente a configuração de tempo limite na função do Lambda. Para obter mais informações, consulte Configurar tempo limite das funções.
`Lambda.FunctionError`	Isso pode acontecer devido a um dos seguintes erros: • Estrutura da saída inválida. Verifique a função e certifique-se de que a saída esteja no formato necessário. Além disso, certifique-se de que os registros processados contenham um status de resultado válido de `Dropped`, `Ok` ou `ProcessingFailed` . • A função do Lambda foi invocada com sucesso, mas retornou um resultado de erro. • O Lambda não pôde descriptografar as variáveis de ambiente porque o acesso ao KMS foi negado. Verifique as configurações de chave do KMS da função, bem como a política de chave. Para obter mais informações, consulte Solução de erros de acesso de chave.

Código de erro	Mensagem de erros e informações
`Lambda.FunctionRequestTimedOut`	O Amazon Data Firehose encontrou que a solicitação não foi concluída antes do erro de configuração do tempo limite da solicitação ao invocar o Lambda. Revise o código Lambda para verificar se o código Lambda deve ser executado além do tempo limite configurado. Nesse caso, considere ajustar as configurações do Lambda, incluindo memória e tempo limite. Para obter mais informações, consulte Configurar as opções da função do Lambda.
`Lambda.TargetServerFailedToRespond`	O Amazon Data Firehose encontrou um erro. Erro de falha no servidor de destino ao chamar o serviço AWS Lambda.
`Lambda.InvalidZipFileException`	O Amazon Data Firehose foi encontrado InvalidZipFileException ao invocar a função Lambda. Verifique as configurações da função do Lambda e o arquivo zip do código do Lambda.
`Lambda.InternalServerError`	“O Amazon Data Firehose foi encontrado InternalServerError ao chamar o serviço Lambda AWS . O Amazon Data Firehose tentará enviar dados novamente um número fixo de vezes. É possível especificar ou substituir as opções de nova tentativa usando as APIs `CreateDeliveryStream` ou `UpdateDestination` . Se o erro persistir, entre em contato com a equipe de suporte AWS da Lambda.
`Lambda.ServiceUnavailable`	O Amazon Data Firehose foi encontrado ServiceUnavailableException ao chamar o serviço Lambda AWS . O Amazon Data Firehose tentará enviar dados novamente um número fixo de vezes. É possível especificar ou substituir as opções de nova tentativa usando as APIs `CreateDeliveryStream` ou `UpdateDestination` . Se o erro persistir, entre em contato com o suporte AWS da Lambda.
`Lambda.InvalidSecurityToken`	Não é possível invocar a função do Lambda devido ao token de segurança inválido. A invocação do Lambda entre partições não é compatível.

Código de erro	Mensagem de erros e informações
`Lambda.InvocationFailure`	Isso pode acontecer devido a um dos seguintes erros: • O Amazon Data Firehose encontrou erros ao chamar o AWS Lambda. A operação será tentada novamente; se o erro persistir, ele será reportado à AWS para resolução." • O Amazon Data Firehose encontrou um KMS da LambdaInvalidState Exception . O Lambda não conseguiu descriptografar as variáveis de ambiente porque a chave do KMS usada está em um estado inválido para descriptografia. Verifique a chave do KMS da função do Lambda. • O Amazon Data Firehose encontrou um da AWS LambdaException Lambda. O Lambda não conseguiu inicializar a imagem do contêiner fornecida. Verifique a imagem. • O Amazon Data Firehose encontrou erros de tempo limite ao chamar o Lambda. AWS O tempo limite máximo da função é de 5 minutos. Para obter mais informações, consulte Duração da execução da transformação de dados.
`Lambda.JsonMappingException`	"Ocorreu um erro ao analisar os registros retornados pela função do Lambda. Certifique-se de que o campo de dados esteja codificado na base 64.

Erros de invocação do Kinesis

O Amazon Data Firehose pode enviar os seguintes erros de invocação do Kinesis para o Logs. CloudWatch

Código de erro	Mensagem de erros e informações
`Kinesis.AccessDenied`	"O acesso foi negado ao chamar o Kinesis. Certifique-se de que a política de acesso no perfil do IAM usado permita o acesso às APIs apropriadas do Kinesis."

Código de erro	Mensagem de erros e informações
`Kinesis.ResourceNotFound`	"O Firehose falhou ao ler o fluxo. Se o Firehose estiver conectado ao Kinesis Stream, o fluxo pode não existir ou o fragmento pode ter sido mesclado ou dividido. Se o Firehose for do DirectPut tipo, o Firehose pode não existir mais.”
`Kinesis.SubscriptionRequired`	"O acesso foi negado ao chamar o Kinesis. Certifique-se de que a função do IAM passada para o acesso ao stream do Kinesis tenha uma assinatura do AWS Kinesis.”
`Kinesis.Throttling`	"Erro de controle de utilização encontrado ao chamar o Kinesis. Isso pode ser devido ao fato de outros aplicativos chamarem as mesmas APIs do stream do Firehose ou porque você criou muitos streams do Firehose com o mesmo stream do Kinesis da origem.”
`Kinesis.Throttling`	"Erro de controle de utilização encontrado ao chamar o Kinesis. Isso pode ser devido ao fato de outros aplicativos chamarem as mesmas APIs do stream do Firehose ou porque você criou muitos streams do Firehose com o mesmo stream do Kinesis da origem.”
`Kinesis.AccessDenied`	"O acesso foi negado ao chamar o Kinesis. Certifique-se de que a política de acesso no perfil do IAM usado permita o acesso às APIs apropriadas do Kinesis."
`Kinesis.AccessDenied`	“O acesso foi negado ao tentar chamar operações de API no Kinesis Stream subjacente. Certifique-se de que a função do IAM seja propagada e válida.”
`Kinesis.KMS.AccessDeniedException`	"O Firehose não tem acesso à chave do KMS usada para criptografar/descriptografar o Kinesis Stream. Por favor, conceda ao perfil de entrega do Firehose acesso à chave."
`Kinesis.KMS.KeyDisabled`	"O Firehose não consegue ler o Kinesis Stream de origem porque a chave do KMS usada para criptografá-lo/descriptografá-lo está desabilitada. Habilite a chave para que as leituras possam continuar."

Código de erro	Mensagem de erros e informações
`Kinesis.KMS.InvalidStateException`	"O Firehose não consegue ler o Kinesis Stream de origem porque a chave do KMS usada para criptografá-lo está em um estado inválido."
`Kinesis.KMS.NotFoundException`	"O Firehose não consegue ler o Kinesis Stream de origem porque a chave do KMS usada para criptografá-lo não foi encontrada."

Erros de invocação do Kinesis DirectPut

O Amazon Data Firehose pode enviar os seguintes erros de DirectPut invocação do Kinesis para o Logs. CloudWatch

Código de erro	Mensagem de erros e informações
`Firehose.KMS.AccessDeniedException`	"O Firehose não tem acesso à chave do KMS. Por favor, verifique a política de chave."
`Firehose.KMS.InvalidStateException`	"O Firehose não consegue descriptografar os dados porque a chave do KMS usada para criptografá-los está em um estado inválido."
`Firehose.KMS.NotFoundException`	"O Firehose não consegue descriptografar os dados porque a chave do KMS usada para criptografá-los não foi encontrada."
`Firehose.KMS.KeyDisabled`	"O Firehose não consegue descriptografar os dados porque a chave do KMS usada para criptografar os dados está desabilitada. Habilite a chave para que a entrega de dados possa prosseguir."

AWS Glue Erros de invocação

O Amazon Data Firehose pode enviar os seguintes erros de AWS Glue invocação para o Logs. CloudWatch

Código de erro	Mensagem de erros e informações
`DataFormatConversion.InvalidSchema`	"O esquema é inválido."
`DataFormatConversion.EntityNotFound`	"Não foi possível encontrar a tabela/banco de dados especificada. Certifique-se de que a tabela/banco de dados exista e que os valores fornecidos na configuração do esquema estejam corretos, especialmente no que diz respeito ao uso de maiúsculas e minúsculas."
`DataFormatConversion.InvalidInput`	"Não foi possível encontrar um esquema correspondente do glue. Certifique-se de que o banco de dados especificado com o ID do catálogo fornecido exista."
`DataFormatConversion.InvalidInput`	"Não foi possível encontrar um esquema correspondente do glue. Certifique-se de que o ARN passado esteja no formato correto."
`DataFormatConversion.InvalidInput`	"Não foi possível encontrar um esquema correspondente do glue. Certifique-se de que o catalogId fornecido seja válido."
`DataFormatConversion.InvalidVersionId`	"Não foi possível encontrar um esquema correspondente do glue. Certifique-se de que a versão especificada da tabela exista."

Código de erro	Mensagem de erros e informações
`DataFormatConversion.NonExistentColumns`	"Não foi possível encontrar um esquema correspondente do glue. Certifique-se de que a tabela esteja configurada com um descritor de armazenamento não nulo contendo as colunas de destino."
`DataFormatConversion.AccessDenied`	"Acesso negado ao assumir o perfil. Certifique-se de que o perfil especificado na configuração de conversão de formato de dados tenha concedido permissão ao serviço Firehose para assumi-lo."
`DataFormatConversion.ThrottledByGlue`	"Erro de controle de utilização encontrado ao chamar o Glue. Aumente o limite da taxa de solicitações ou reduza a taxa atual de chamadas ao glue por outras aplicações."
`DataFormatConversion.AccessDenied`	"O acesso foi negado ao chamar o Glue. Certifique-se de que o perfil especificado na configuração de conversão de formato de dados tenha as permissões necessárias."
`DataFormatConversion.InvalidGlueRole`	"Perfil inválido. Certifique-se de que o perfil especificado na configuração de conversão de formato de dados exista."
`DataFormatConversion.InvalidGlueRole`	"O token de segurança incluído na solicitação é inválido. Certifique-se de que o perfil do IAM associado fornecido ao firehose não seja excluído."
`DataFormatConversion.GlueNotAvailableInRegion`	“O AWS Glue ainda não está disponível na região que você especificou; especifique uma região diferente.”

Código de erro	Mensagem de erros e informações
`DataFormatConversion.GlueEncryptionException`	"Houve um erro ao recuperar a chave-mestra. Certifique-se de que a chave exista e tenha as permissões de acesso corretas."
`DataFormatConversion.SchemaValidationTimeout`	"O tempo limite foi atingido ao recuperar a tabela do Glue. Se você tiver um grande número de versões da tabela Glue, adicione a permissão “glue:GetTableVersion” (recomendado) ou exclua as versões não utilizadas da tabela. Se você não tiver um grande número de tabelas no Glue, entre em contato com o AWS Support.”
`DataFirehose.InternalError`	"O tempo limite foi atingido ao recuperar a tabela do Glue. Se você tiver um grande número de versões da tabela Glue, adicione a permissão “glue:GetTableVersion” (recomendado) ou exclua as versões não utilizadas da tabela. Se você não tiver um grande número de tabelas no Glue, entre em contato com o AWS Support.”
`DataFormatConversion.GlueEncryptionException`	"Houve um erro ao recuperar a chave-mestra. Certifique-se de que a chave exista e que o estado esteja correto."

DataFormatConversion Erros de invocação

O Amazon Data Firehose pode enviar os seguintes erros de DataFormatConversion invocação para o Logs. CloudWatch

Código de erro	Mensagem de erros e informações
`DataFormatConversion.InvalidSchema`	"O esquema é inválido."

Código de erro	Mensagem de erros e informações
`DataFormatConversion.ValidationException`	"Os nomes e tipos das colunas não devem ser strings vazias."
`DataFormatConversion.ParseError`	"Foi encontrado um JSON malformados."
`DataFormatConversion.MalformedData`	"Os dados não correspondem ao esquema."
`DataFormatConversion.MalformedData`	"O comprimento da chave json não deve ser maior que 262.144"
`DataFormatConversion.MalformedData`	"Os dados não podem ser decodificados como UTF-8."
`DataFormatConversion.MalformedData`	"Caractere ilegal encontrado entre tokens."
`DataFormatConversion.InvalidTypeFormat`	"O formato do tipo é inválido. Verifique a sintaxe do tipo."

Código de erro	Mensagem de erros e informações
`DataFormatConversion.InvalidSchema`	"Esquema inválido. Certifique-se de que não haja caracteres especiais ou espaços em branco nos nomes das colunas.”
`DataFormatConversion.InvalidRecord`	"O registro não está de acordo com o esquema. Uma ou mais chaves de mapa eram inválidas para map<string,string>."
`DataFormatConversion.MalformedData`	"O JSON recebido continha uma primitiva no nível superior. O nível superior deve ser um objeto ou uma matriz."
`DataFormatConversion.MalformedData`	"O JSON recebido continha uma primitiva no nível superior. O nível superior deve ser um objeto ou uma matriz."
`DataFormatConversion.MalformedData`	"O registro estava vazio ou continha apenas espaços em branco."
`DataFormatConversion.MalformedData`	"Foram encontrados caracteres inválidos."
`DataFormatConversion.MalformedData`	"Foi encontrado um formato de timestamp inválido ou incompatível. Consulte o Guia do desenvolvedor do Firehose para ver os formatos de timestamp compatíveis."

Código de erro	Mensagem de erros e informações
`DataFormatConversion.MalformedData`	"Um tipo escalar foi encontrado nos dados, mas um tipo complexo estava especificado no esquema."
`DataFormatConversion.MalformedData`	"Os dados não correspondem ao esquema."
`DataFormatConversion.MalformedData`	"Um tipo escalar foi encontrado nos dados, mas um tipo complexo estava especificado no esquema."
`DataFormatConversion.ConversionFailureException`	"ConversionFailureException"
`DataFormatConversion.DataFormatConversionCustomerErrorException`	"DataFormatConversionCustomerErrorException"

Código de erro	Mensagem de erros e informações
`DataFormatConversion.DataFormatConversionCustomerErrorException`	"DataFormatConversionCustomerErrorException"
`DataFormatConversion.MalformedData`	"Os dados não correspondem ao esquema."
`DataFormatConversion.InvalidSchema`	"O esquema é inválido."
`DataFormatConversion.MalformedData`	"Os dados não correspondem ao esquema. Formato inválido para uma ou mais datas."
`DataFormatConversion.MalformedData`	"Os dados contêm uma estrutura JSON altamente aninhada que não é compatível."
`DataFormatConversion.EntityNotFound`	"Não foi possível encontrar a tabela/banco de dados especificada. Certifique-se de que a tabela/banco de dados exista e que os valores fornecidos na configuração do esquema estejam corretos, especialmente no que diz respeito ao uso de maiúsculas e minúsculas."

Código de erro	Mensagem de erros e informações
`DataFormatConversion.InvalidInput`	"Não foi possível encontrar um esquema correspondente do glue. Certifique-se de que o banco de dados especificado com o ID do catálogo fornecido exista."
`DataFormatConversion.InvalidInput`	"Não foi possível encontrar um esquema correspondente do glue. Certifique-se de que o ARN passado esteja no formato correto."
`DataFormatConversion.InvalidInput`	"Não foi possível encontrar um esquema correspondente do glue. Certifique-se de que o catalogId fornecido seja válido."
`DataFormatConversion.InvalidVersionId`	"Não foi possível encontrar um esquema correspondente do glue. Certifique-se de que a versão especificada da tabela exista."
`DataFormatConversion.NonExistentColumns`	"Não foi possível encontrar um esquema correspondente do glue. Certifique-se de que a tabela esteja configurada com um descritor de armazenamento não nulo contendo as colunas de destino."
`DataFormatConversion.AccessDenied`	"Acesso negado ao assumir o perfil. Certifique-se de que o perfil especificado na configuração de conversão de formato de dados tenha concedido permissão ao serviço Firehose para assumi-lo."
`DataFormatConversion.ThrottledByGlue`	"Erro de controle de utilização encontrado ao chamar o Glue. Aumente o limite da taxa de solicitações ou reduza a taxa atual de chamadas ao glue por outras aplicações."

Código de erro	Mensagem de erros e informações
`DataFormatConversion.AccessDenied`	"O acesso foi negado ao chamar o Glue. Certifique-se de que o perfil especificado na configuração de conversão de formato de dados tenha as permissões necessárias."
`DataFormatConversion.InvalidGlueRole`	"Perfil inválido. Certifique-se de que o perfil especificado na configuração de conversão de formato de dados exista."
`DataFormatConversion.GlueNotAvailableInRegion`	“O AWS Glue ainda não está disponível na região que você especificou; especifique uma região diferente.”
`DataFormatConversion.GlueEncryptionException`	"Houve um erro ao recuperar a chave-mestra. Certifique-se de que a chave exista e tenha as permissões de acesso corretas."
`DataFormatConversion.SchemaValidationTimeout`	"O tempo limite foi atingido ao recuperar a tabela do Glue. Se você tiver um grande número de versões da tabela Glue, adicione a permissão “glue:GetTableVersion” (recomendado) ou exclua as versões não utilizadas da tabela. Se você não tiver um grande número de tabelas no Glue, entre em contato com o AWS Support.”
`DataFirehose.InternalError`	"O tempo limite foi atingido ao recuperar a tabela do Glue. Se você tiver um grande número de versões da tabela Glue, adicione a permissão “glue:GetTableVersion” (recomendado) ou exclua as versões não utilizadas da tabela. Se você não tiver um grande número de tabelas no Glue, entre em contato com o AWS Support.”

Código de erro	Mensagem de erros e informações
`DataFormatConversion.MalformedData`	"Um ou mais campos têm formato incorreto."

Acessando CloudWatch registros do Amazon Data Firehose

Você pode visualizar os registros de erros relacionados à falha na entrega de dados do Amazon Data Firehose usando o console do Amazon Data Firehose ou o console. CloudWatch Os procedimentos a seguir mostram como acessar os logs de erros usando estes dois métodos:

Para acessar os registros de erros usando o console do Amazon Data Firehose

1. Faça login AWS Management Console e abra o console Firehose em https://console.aws.amazon.com/firehose
2. Na barra de navegação, escolha uma AWS região.
3. Escolha um nome de stream do Firehose para acessar a página de detalhes do stream do Firehose.
4. Escolha Error Log para exibir uma lista de logs de erros relacionados à falha de entrega de dados.

Para acessar os registros de erros usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Na barra de navegação, escolha uma Região.
3. No painel de navegação, selecione Logs.
4. Escolha um grupo de logs e um fluxo de logs para visualizar uma lista de logs de erros relacionados à falha de entrega de dados.

Monitorar a integridade do Kinesis Agent

O Kinesis Agent publica CloudWatch métricas personalizadas com um namespace de. AWS KinesisAgent Ele ajuda a avaliar se o agente está íntegro, enviando dados para o Amazon Data

Firehose conforme especificado e consumindo a quantidade adequada de recursos de CPU e memória no produtor de dados.

Métricas como número de registros e bytes enviados são úteis para entender a taxa na qual o agente está enviando dados para o stream do Firehose. Quando essas métricas ficarem abaixo dos limites esperados em alguns percentuais ou caírem para zero, isso poderá indicar problemas de configuração, erros de rede ou problemas de integridade do agente. As métricas como consumo de CPU e memória no host e contadores de erros do agente indicam uso de recurso por parte do produtor de dados e fornece informações sobre erros potenciais de configuração ou de host. Por fim, o agente também registra exceções de serviço para ajudar a investigar problemas do agente.

As métricas do agente são reportadas na região especificada na configuração de agente `cloudwatch.endpoint`. Para ter mais informações, consulte Configurações do agente.

As métricas do Cloudwatch publicadas de vários Kinesis Agents são agregadas ou combinadas.

Há um custo nominal para as métricas emitidas pelo Kinesis Agent, que são habilitadas por padrão. Para obter mais informações, consulte Amazon CloudWatch Pricing.

Monitoramento com CloudWatch

O Kinesis Agent envia as seguintes métricas para o. CloudWatch

Métrica	Descrição
`BytesSent`	O número de bytes enviados para o stream do Firehose durante o período especificado. Unidade: bytes
`RecordSendAttempts`	O número de tentativas de registro (primeira vez ou como nova tentativa) em uma chamada para `PutRecordBatch` no período especificado. Unidades: contagem
`RecordSendErrors`	O número de registros que retornaram status de falha em uma chamada para `PutRecordBatch` , incluindo novas tentativas, no período especificado. Unidades: contagem

Métrica	Descrição
`ServiceErrors`	O número de chamadas para `PutRecordBatch` que resultaram em erro de serviço (diferente de um erro de controle de utilização) no período especificado. Unidades: contagem

Registrando chamadas de API do Amazon Data Firehose com AWS CloudTrail

O Amazon Data Firehose é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Amazon Data Firehose. CloudTrail captura todas as chamadas de API para o Amazon Data Firehose como eventos. As chamadas capturadas incluem chamadas do console do Amazon Data Firehose e chamadas de código para as operações da API do Amazon Data Firehose. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para o Amazon Data Firehose. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Amazon Data Firehose, o endereço IP a partir do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais CloudTrail, inclusive como configurá-lo e ativá-lo, consulte o Guia AWS CloudTrail do usuário.

Informações sobre o Amazon Data Firehose em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando uma atividade de evento suportada ocorre no Amazon Data Firehose, essa atividade é registrada em um CloudTrail evento junto com outros eventos de AWS serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos do Amazon Data Firehose, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, a trilha se aplica a todas as AWS regiões. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de

log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte as informações a seguir.

- Visão Geral para Criar uma Trilha
- CloudTrail Serviços e integrações compatíveis
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e recebendo arquivos de CloudTrail log de várias contas

O Amazon Data Firehose suporta o registro das seguintes ações como eventos em arquivos de CloudTrail log:

- CreateDeliveryStream
- DeleteDeliveryStream
- DescribeDeliveryStream
- ListDeliveryStreams
- ListTagsForDeliveryStream
- TagDeliveryStream
- StartDeliveryStreamEncryption
- StopDeliveryStreamEncryption
- UntagDeliveryStream
- UpdateDestination

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 root ou AWS Identity and Access Management (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 serviço.

Para obter mais informações, consulte o elemento CloudTrailUserIdentity.

Exemplo: entradas do arquivo de log do Amazon Data Firehose

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, 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, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra as `DeleteDeliveryStream` ações `CreateDeliveryStream` `DescribeDeliveryStream``ListDeliveryStreams`,`UpdateDestination`,, e.

```
{
  "Records":[
        {
            "eventVersion":"1.02",
            "userIdentity":{
                "type":"IAMUser",
                "principalId":"AKIAIOSFODNN7EXAMPLE",
                "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
                "accountId":"111122223333",
                "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
                "userName":"CloudTrail_Test_User"
            },
            "eventTime":"2016-02-24T18:08:22Z",
            "eventSource":"firehose.amazonaws.com",
            "eventName":"CreateDeliveryStream",
            "awsRegion":"us-east-1",
            "sourceIPAddress":"127.0.0.1",
            "userAgent":"aws-internal/3",
            "requestParameters":{
                "deliveryStreamName":"TestRedshiftStream",
                "redshiftDestinationConfiguration":{
                "s3Configuration":{
                    "compressionFormat":"GZIP",
                    "prefix":"prefix",
                    "bucketARN":"arn:aws:s3:::firehose-cloudtrail-test-bucket",
                    "roleARN":"arn:aws:iam::111122223333:role/Firehose",
                    "bufferingHints":{
                        "sizeInMBs":3,
                        "intervalInSeconds":900
```

```
                    },
                    "encryptionConfiguration":{
                        "kMSEncryptionConfig":{
                            "aWSKMSKeyARN":"arn:aws:kms:us-east-1:key"
                        }
                    }
                },
                "clusterJDBCURL":"jdbc:redshift://example.abc123.us-
west-2.redshift.amazonaws.com:5439/dev",
                "copyCommand":{
                    "copyOptions":"copyOptions",
                    "dataTableName":"dataTable"
                },
                "password":"",
                "username":"",
                "roleARN":"arn:aws:iam::111122223333:role/Firehose"
            }
        },
        "responseElements":{
            "deliveryStreamARN":"arn:aws:firehose:us-
east-1:111122223333:deliverystream/TestRedshiftStream"
        },
        "requestID":"958abf6a-db21-11e5-bb88-91ae9617edf5",
        "eventID":"875d2d68-476c-4ad5-bbc6-d02872cfc884",
        "eventType":"AwsApiCall",
        "recipientAccountId":"111122223333"
    },
    {
        "eventVersion":"1.02",
        "userIdentity":{
            "type":"IAMUser",
            "principalId":"AKIAIOSFODNN7EXAMPLE",
            "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
            "accountId":"111122223333",
            "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
            "userName":"CloudTrail_Test_User"
        },
        "eventTime":"2016-02-24T18:08:54Z",
        "eventSource":"firehose.amazonaws.com",
        "eventName":"DescribeDeliveryStream",
        "awsRegion":"us-east-1",
        "sourceIPAddress":"127.0.0.1",
        "userAgent":"aws-internal/3",
        "requestParameters":{
```

```
            "deliveryStreamName":"TestRedshiftStream"
        },
        "responseElements":null,
        "requestID":"aa6ea5ed-db21-11e5-bb88-91ae9617edf5",
        "eventID":"d9b285d8-d690-4d5c-b9fe-d1ad5ab03f14",
        "eventType":"AwsApiCall",
        "recipientAccountId":"111122223333"
    },
    {
        "eventVersion":"1.02",
        "userIdentity":{
            "type":"IAMUser",
            "principalId":"AKIAIOSFODNN7EXAMPLE",
            "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
            "accountId":"111122223333",
            "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
            "userName":"CloudTrail_Test_User"
        },
        "eventTime":"2016-02-24T18:10:00Z",
        "eventSource":"firehose.amazonaws.com",
        "eventName":"ListDeliveryStreams",
        "awsRegion":"us-east-1",
        "sourceIPAddress":"127.0.0.1",
        "userAgent":"aws-internal/3",
        "requestParameters":{
            "limit":10
        },
        "responseElements":null,
        "requestID":"d1bf7f86-db21-11e5-bb88-91ae9617edf5",
        "eventID":"67f63c74-4335-48c0-9004-4ba35ce00128",
        "eventType":"AwsApiCall",
        "recipientAccountId":"111122223333"
    },
    {
        "eventVersion":"1.02",
        "userIdentity":{
            "type":"IAMUser",
            "principalId":"AKIAIOSFODNN7EXAMPLE",
            "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
            "accountId":"111122223333",
            "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
            "userName":"CloudTrail_Test_User"
        },
        "eventTime":"2016-02-24T18:10:09Z",
```

```
        "eventSource":"firehose.amazonaws.com",
        "eventName":"UpdateDestination",
        "awsRegion":"us-east-1",
        "sourceIPAddress":"127.0.0.1",
        "userAgent":"aws-internal/3",
        "requestParameters":{
            "destinationId":"destinationId-000000000001",
            "deliveryStreamName":"TestRedshiftStream",
            "currentDeliveryStreamVersionId":"1",
            "redshiftDestinationUpdate":{
                "roleARN":"arn:aws:iam::111122223333:role/Firehose",
                "clusterJDBCURL":"jdbc:redshift://example.abc123.us-
west-2.redshift.amazonaws.com:5439/dev",
                "password":"",
                "username":"",
                "copyCommand":{
                    "copyOptions":"copyOptions",
                    "dataTableName":"dataTable"
                },
                "s3Update":{
                    "bucketARN":"arn:aws:s3:::firehose-cloudtrail-test-bucket-update",
                    "roleARN":"arn:aws:iam::111122223333:role/Firehose",
                    "compressionFormat":"GZIP",
                    "bufferingHints":{
                        "sizeInMBs":3,
                        "intervalInSeconds":900
                    },
                    "encryptionConfiguration":{
                        "kMSEncryptionConfig":{
                            "aWSKMSKeyARN":"arn:aws:kms:us-east-1:key"
                        }
                    },
                    "prefix":"arn:aws:s3:::firehose-cloudtrail-test-bucket"
                }
            }
        },
        "responseElements":null,
        "requestID":"d549428d-db21-11e5-bb88-91ae9617edf5",
        "eventID":"1cb21e0b-416a-415d-bbf9-769b152a6585",
        "eventType":"AwsApiCall",
        "recipientAccountId":"111122223333"
    },
    {
        "eventVersion":"1.02",
```

```
        "userIdentity":{
            "type":"IAMUser",
            "principalId":"AKIAIOSFODNN7EXAMPLE",
            "arn":"arn:aws:iam::111122223333:user/CloudTrail_Test_User",
            "accountId":"111122223333",
            "accessKeyId":"AKIAI44QH8DHBEXAMPLE",
            "userName":"CloudTrail_Test_User"
        },
        "eventTime":"2016-02-24T18:10:12Z",
        "eventSource":"firehose.amazonaws.com",
        "eventName":"DeleteDeliveryStream",
        "awsRegion":"us-east-1",
        "sourceIPAddress":"127.0.0.1",
        "userAgent":"aws-internal/3",
        "requestParameters":{
            "deliveryStreamName":"TestRedshiftStream"
        },
        "responseElements":null,
        "requestID":"d85968c1-db21-11e5-bb88-91ae9617edf5",
        "eventID":"dd46bb98-b4e9-42ff-a6af-32d57e636ad1",
        "eventType":"AwsApiCall",
        "recipientAccountId":"111122223333"
    }
  ]
}
```

Prefixos personalizados para objetos do Amazon S3

<evaluated prefix><suffix>Os objetos entregues ao Amazon S3 seguem o formato do nome de. Você pode especificar seu prefixo personalizado que inclui expressões que são avaliadas em tempo de execução. O prefixo personalizado que você especificar substituirá o prefixo padrão de. `YYYY/MM/dd/HH`

É possível usar expressões nos seguintes formatos no seu prefixo personalizado: `!{namespace:value}`, em que `namespace` pode ser um dos que se seguem, como explicado nas próximas seções.

- `firehose`
- `timestamp`
- `partitionKeyFromQuery`
- `partitionKeyFromLambda`

Se um prefixo terminar com uma barra, ele aparecerá como uma pasta no bucket do Amazon S3. Para obter mais informações, consulte Formato de nome de objeto do Amazon S3 no Amazon Data FirehoseDeveloper Guide.

O namespace **`timestamp`**.

Os valores válidos para esse namespace são cadeias de caracteres Java válidas. DateTimeFormatter Como um exemplo, no ano 2018, a expressão `!{timestamp:yyyy}` é avaliada para `2018`.

Ao avaliar os timestamps, o Firehose usa o timestamp de chegada aproximado do registro mais antigo contido no objeto Amazon S3 que está sendo gravado.

Por padrão, o timestamp está em UTC. Mas você pode especificar o fuso horário de sua preferência. Por exemplo, você pode configurar o fuso horário para Ásia/Tóquio na configuração de parâmetros da API () AWS Management Console ou na configuração de parâmetros da API (CustomTimeZone) se quiser usar o horário padrão do Japão em vez do UTC. Para ver a lista de fusos horários compatíveis, consulte Formato de nome de objeto do Amazon S3.

Se você usar o namespace `timestamp` mais de uma vez na mesma expressão do prefixo, cada instância será avaliada no mesmo momento.

O namespace **`firehose`**.

Há dois valores que você pode usar com esse namespace: `error-output-type` e `random-string`. A tabela a seguir explica como usá-los.

Os valores do namespace **`firehose`**.

Conversão	Descrição	Exemplo de entrada	Exemplo de saída	Observações
`error-output-type`	É avaliado como uma das seguintes sequências de caracteres, dependendo da configuração do stream do Firehose e do motivo da falha: {processing-failed, AmazonOpenSearchService-failed, splunk-failed,,}. format-conversion-failed http-endpoint-failed Se você usá-lo mais de uma vez na mesma expressão, cada instância será avaliada para a mesma string de erro.	`myPrefix/result=!{firehose:error-output-type}/!{timestamp:yyyy/MM/dd}`	`myPrefix/result=processing-failed/2018/08/03`	O error-output-type valor só pode ser usado no ErrorOutputPrefix campo.

Conversão	Descrição	Exemplo de entrada	Exemplo de saída	Observações
`random-string`	Avalia para uma string aleatória de 11 caracteres. Se você usá-lo mais de uma vez na mesma expressão, cada instância será avaliada para uma nova string aleatória.	`myPrefix/!{firehose:random-string}/`	`myPrefix/046b6c7f-0b/`	É possível usá-lo com os dois tipos de prefixo. Pode ser colocado no início da string de formato para obter um prefixo aleatório, o que, às vezes, é necessário para atingir uma throughput extremamente alto com o Amazon S3.

Namespaces **`partitionKeyFromLambda`** e **`partitionKeyFromQuery`**

Para o particionamento dinâmico, você deve usar o seguinte formato de expressão no prefixo de bucket do S3: `!{namespace:value}`, em que o namespace pode ser `partitionKeyFromQuery`, `partitionKeyFromLambda` ou ambos. Se estiver usando análise em linha para criar as chaves de particionamento para os dados da fonte, você deverá especificar um valor de prefixo de bucket do S3 consistindo em expressões especificadas no seguinte formato: `"partitionKeyFromQuery:keyID"`. Se estiver usando função do AWS Lambda para criar as chaves de particionamento para os dados da fonte, você deverá especificar um valor de prefixo de bucket de S3 que consista em expressões especificadas no seguinte formato: `"partitionKeyFromLambda:keyID"`. Para obter mais informações, consulte “Escolha o Amazon S3 para seu destino” em Criar um stream do Amazon Firehose.

Regras semânticas

As regras a seguir se aplicam às expressões `Prefix` e `ErrorOutputPrefix`.

- Para o namespace `timestamp`, qualquer caractere que não estiver em aspas simples é avaliado. Em outras palavras, qualquer string recuada com aspas simples no campo do valor é considerada literalmente.
- Se você especificar um prefixo que não contenha uma expressão de namespace de carimbo de data/hora, o Firehose anexará a expressão ao valor no `!{timestamp:yyyy/MM/dd/HH/}` campo. `Prefix`
- A sequência `!{` pode aparecer somente em expressões `!{namespace:`*`value`*`}`.
- `ErrorOutputPrefix` poderá ser nulo somente se `Prefix` não tiver expressões; Neste caso, `Prefix` é avaliado como `<specified-prefix>yyyy/MM/DDD/HH/`, e `ErrorOutputPrefix` é avaliado como `<specified-prefix><error-output-type>YYYY/MM/DDD/HH/`. `DDD` representa o dia do ano.
- Se você especificar uma expressão para `ErrorOutputPrefix`, deverá incluir pelo menos uma instância de `!{firehose:error-output-type}`.
- `Prefix` não pode conter `!{firehose:error-output-type}`.
- `Prefix` e `ErrorOutputPrefix` não podem ter mais de 512 caracteres após serem avaliados.
- Se o destino for o Amazon Redshift, o `Prefix` não deverá conter expressões e o `ErrorOutputPrefix` deverá ser nulo.
- Quando o destino é Amazon OpenSearch Service ou Splunk e não `ErrorOutputPrefix` é especificado, o Firehose usa `Prefix` o campo para registros com falha.
- Quando o destino é o Amazon S3, o `Prefix` e o `ErrorOutputPrefix` na configuração de destino do Amazon S3 são usados para registros bem-sucedidos e registros com falha, respectivamente. Se você usar a AWS CLI ou a API, poderá usar a `ExtendedS3DestinationConfiguration` para especificar uma configuração de backup do Amazon S3 com seu próprio `Prefix` e `ErrorOutputPrefix`.
- Quando você usa AWS Management Console e define o destino como Amazon S3, o Firehose usa `Prefix` e `ErrorOutputPrefix` na configuração de destino para registros bem-sucedidos e registros com falha, respectivamente. Se você especificar um prefixo, mas nenhum prefixo de erro, o Firehose definirá automaticamente o prefixo de erro como. `!{firehose:error-output-type}/`

- Quando você usa `ExtendedS3DestinationConfiguration` com o AWS CLI, a API ou AWS CloudFormation, se você especificar um`S3BackupConfiguration`, o Firehose não fornece um padrão. `ErrorOutputPrefix`
- Você não pode usar `partitionKeyFromLambda` `partitionKeyFromQuery` namespaces ao criar ErrorOutputPrefix expressões.

Prefixos de exemplo

Exemplos de **`Prefix`** e **`ErrorOutputPrefix`**

Entrada	Prefixo avaliado (às 10:30 AM UTC em 27 de agosto de 2018)

`Prefix`: não especificado
 	
`ErrorOutputPrefix` : `myFirehoseFailures/!{firehose:error-output-type}/`
	`Prefix`: `2018/08/27/10`
	
	`ErrorOutputPrefix` : `myFirehoseFailures/processing-failed/`

`Prefix`: `!{timestamp:yyyy/MM/dd}` `ErrorOutputPrefix` : não especificado	Entrada inválida: `ErrorOutputPrefix` não poderá ser nulo quando o prefixo tiver expressões

`Prefix`: `myFirehose/DeliveredYear=!{timestamp:yyyy}/anyMonth/rand=!{firehose:random-string}`
 	
`ErrorOutputPrefix` : `myFirehoseFailures/!{firehose:error-output-type}/!{timestamp:yyyy}/anyMonth/!{timestamp:dd}`
	`Prefix`: `myFirehose/DeliveredYear=2018/anyMonth/rand=5abf82daaa5`
	
	`ErrorOutputPrefix` : `myFirehoseFailures/processing-failed/2018/anyMonth/10`

`Prefix`: `myPrefix/year=!{timestamp:yyyy}/month=!{timestamp:MM}/day=!{timestamp:dd}/hour=!{timestamp:HH}/`	`Prefix`: `myPrefix/year=2018/month=07/day=06/hour=23/`
	
	`ErrorOutputPrefix` : `myErrorPrefix/year=2018/month=07/day=06/hour=23/processing-failed`

Entrada	Prefixo avaliado (às 10:30 AM UTC em 27 de agosto de 2018)
`ErrorOutputPrefix` : `myErrorPrefix/year=!{timestamp:yyyy}/month=!{timestamp:MM}/day=!{timestamp:dd}/hour=!{timestamp:HH}/!{firehose:error-output-type}`	

`Prefix`: `myFirehosePrefix/`
 	
`ErrorOutputPrefix` : não especificado
	`Prefix`: `myFirehosePrefix/2018/08/27/`
	
	`ErrorOutputPrefix` : `myFirehosePrefix/processing-failed/2018/08/27/`

Usando o Amazon Data Firehose com AWS PrivateLink

Interface VPC endpoints ()AWS PrivateLink para Amazon Data Firehose

Você pode usar uma interface VPC endpoint para impedir que o tráfego entre sua Amazon VPC e o Amazon Data Firehose saia da rede Amazon. Os endpoints VPC de interface não exigem um gateway de internet, dispositivo NAT, conexão VPN ou conexão. AWS Direct Connect Os endpoints de VPC da Interface são alimentados por AWS PrivateLink uma AWS tecnologia que permite a comunicação privada entre AWS serviços usando uma interface de rede elástica com IPs privados em sua Amazon VPC. Para obter mais informações, consulte o Amazon Virtual Private Cloud.

Usando a interface VPC endpoints ()AWS PrivateLink para o Amazon Data Firehose

Para começar, crie uma interface VPC endpoint para que o tráfego do Amazon Data Firehose dos recursos do Amazon VPC comece a fluir pela interface VPC endpoint. Ao criar um endpoint, você pode anexar a ele uma política de endpoint que controla o acesso ao Amazon Data Firehose. Para saber mais sobre o uso de políticas para controlar o acesso de um VPC endpoint ao Amazon Data Firehose, consulte Controlando o acesso aos serviços com VPC Endpoints.

O exemplo a seguir mostra como você pode configurar uma AWS Lambda função em uma VPC e criar um VPC endpoint para permitir que a função se comunique com segurança com o serviço Amazon Data Firehose. Neste exemplo, você usa uma política que permite que a função Lambda liste os streams do Firehose na região atual, mas não descreva nenhum stream do Firehose.

Criar um VPC endpoint

1. Faça login AWS Management Console e abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel da VPC, selecione Endpoints.
3. Escolha Criar Endpoint.
4. Na lista de nomes de serviço, escolha `com.amazonaws.`*`your_region`*`.kinesis-firehose`.
5. Escolha a VPC e uma ou mais sub-redes nas quais criar o endpoint.

6. Escolha um ou mais grupos de segurança para associar ao endpoint.
7. Para Policy (Política), selecione Custom (Personalizar) e cole a seguinte política:

```
{
    "Statement": [
        {
            "Sid": "Allow-only-specific-PrivateAPIs",
            "Principal": "*",
            "Action": [
                "firehose:ListDeliveryStreams"
            ],
            "Effect": "Allow",
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "Allow-only-specific-PrivateAPIs",
            "Principal": "*",
            "Action": [
                "firehose:DescribeDeliveryStream"
            ],
            "Effect": "Deny",
            "Resource": [
                "*"
            ]
        }
    ]
}
```

8. Escolha Criar endpoint.

Criar um perfil do IAM para ser usado com a função do Lambda

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Perfis e Criar perfil.
3. Em Selecionar o tipo de entidade confiável, deixe a seleção padrão Serviço da AWS .
4. Em Choose the service that will use this role (Escolha o serviço que usará essa função), escolha Lambda.
5. Escolha Next: Permissions (Próximo: Permissões).

6. Na lista de políticas, procure e adicione as duas políticas chamadas `AWS LambdaVPCAccessExecutionRole` e `AmazonDataFirehoseReadOnlyAccess`.

> **Important**
>
> Este é um exemplo. Você pode precisar de políticas mais rigorosas para o ambiente de produção.

7. Escolha Próximo: etiquetas. Para a finalidade deste exercício, não é necessário adicionar tags. Selecione Next: Review (Próximo: revisar).
8. Insira um nome para o perfil e escolha Criar perfil.

Criar uma função do Lambda dentro da VPC

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. Insira um nome para a função e, em seguida, defina Tempo de Execução como `Python 3.9` ou superior.
5. Em Permissions (Permissões), expanda Choose or create an execution role (Escolher ou criar uma função de execução).
6. Na lista Execution role (Função de execução), selecione Use an existing role (Usar uma função existente).
7. Na lista Existing role (Função existente), selecione a função criada acima.
8. Escolha a opção Criar função.
9. Em Function code (Código da função), cole o código a seguir.

```
import json
import boto3
import os
from botocore.exceptions import ClientError

def lambda_handler(event, context):
    REGION = os.environ['AWS_REGION']
    client = boto3.client(
        'firehose',
        REGION
```

```
        )
        print("Calling list_delivery_streams with ListDeliveryStreams allowed
policy.")
        delivery_stream_request = client.list_delivery_streams()
        print("Successfully returned list_delivery_streams request %s." % (
            delivery_stream_request
        ))
        describe_access_denied = False
        try:
            print("Calling describe_delivery_stream with DescribeDeliveryStream
denied policy.")
            delivery_stream_info =
client.describe_delivery_stream(DeliveryStreamName='test-describe-denied')
        except ClientError as e:
            error_code = e.response['Error']['Code']
            print ("Caught %s." % (error_code))
            if error_code == 'AccessDeniedException':
                describe_access_denied = True

        if not describe_access_denied:
            raise
        else:
            print("Access denied test succeeded.")
```

10. Em Basic settings (Configurações básicas), defina o tempo limite como 1 minuto.
11. Em Network (Rede), selecione a VPC onde você criou o endpoint acima e selecione as sub-redes e o grupo de segurança que foram associados ao endpoint quando ele foi criado.
12. Próximo ao alto da página, selecione Salvar.
13. Escolha Testar.
14. Insira o nome de um evento e escolha Criar.
15. Escolha Test (Testar) novamente. Isso faz com que a função seja executada. Depois que o resultado da execução for exibido, expanda Details (Detalhes) e compare a saída do log com o código da função. Os resultados bem-sucedidos mostram uma lista dos fluxos do Firehose na região, bem como a seguinte saída:

```
Calling describe_delivery_stream.

AccessDeniedException

Access denied test succeeded.
```

Disponibilidade

No momento, VPC endpoints de interface são compatíveis com as seguintes regiões:

- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)
- Asia Pacific (Mumbai)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Ásia-Pacífico (Hong Kong)
- Canadá (Central)
- Oeste do Canadá (Calgary)
- China (Pequim)
- China (Ningxia)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- América do Sul (São Paulo)
- AWS GovCloud (Leste dos EUA)
- AWS GovCloud (Oeste dos EUA)
- Europa (Espanha)
- Oriente Médio (Emirados Árabes Unidos)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Osaka)
- Israel (Tel Aviv)

Marcando seus streams do Firehose no Amazon Data Firehose

Você pode atribuir seus próprios metadados aos streams do Firehose que você cria no Amazon Data Firehose na forma de tags. Tag é um par de chave-valor que você define para um stream. Usar tags é uma maneira simples, porém poderosa, de gerenciar AWS recursos e organizar dados, incluindo dados de faturamento.

Tópicos

Conceitos básicos de tags

Você pode usar a API Amazon Data Firehose para concluir as seguintes tarefas:

- Adicione tags a um stream do Firehose.
- Liste as tags dos seus streams do Firehose.
- Remova as tags de um stream do Firehose.

Você pode usar tags para categorizar seus streams do Firehose. Por exemplo, você pode categorizar os streams do Firehose por finalidade, proprietário ou ambiente. Como você define a chave e o valor para cada marca, você pode criar um conjunto de categorias personalizado para atender às suas necessidades específicas. Por exemplo, você pode definir um conjunto de tags que ajuda a rastrear streams do Firehose por proprietário e aplicativo associado.

Estes são diversos exemplos de tags:

- `Project: Project name`
- `Owner: Name`
- `Purpose: Load testing`

- `Application: Application name`
- `Environment: Production`

Se você especificar tags na `CreateDeliveryStream` ação, o Amazon Data Firehose executará uma autorização adicional na `firehose:TagDeliveryStream` ação para verificar se os usuários têm permissão para criar tags. Se você não fornecer essa permissão, as solicitações para criar novos streams do Firehose com tags de recursos do IAM falharão, conforme a seguir`AccessDeniedException`.

```
AccessDeniedException
User: arn:aws:sts::x:assumed-role/x/x is not authorized to perform:
 firehose:TagDeliveryStream on resource: arn:aws:firehose:us-east-1:x:deliverystream/x
 with an explicit deny in an identity-based policy.
```

O exemplo a seguir demonstra uma política que permite aos usuários criar um stream do Firehose e aplicar tags.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "firehose:CreateDeliveryStream",
            "Resource": "*",
            }
        },
        {
            "Effect": "Allow",
            "Action": "firehose:TagDeliveryStream",
            "Resource": "*",
            }
        }
    ]
}
```

Monitoramento de custos com marcação

Você pode usar tags para categorizar e monitorar seus AWS custos. Quando você aplica tags aos seus AWS recursos, incluindo streams do Firehose, seu relatório de alocação de AWS custos inclui

o uso e os custos agregados por tags. Você pode organizar seus custos de vários serviços aplicando tags que representam categorias de negócios (como centros de custos, nomes de aplicativos ou proprietários). Para obter mais informações, consulte Usar etiquetas de alocação de custos para relatórios de faturamento personalizados no Manual do usuário do AWS Billing .

Restrições de tag

As restrições a seguir se aplicam às tags no Amazon Data Firehose.

Restrições básicas

- O número máximo de tags por recurso (stream) é 50.
- As chaves e os valores de tags diferenciam maiúsculas de minúsculas.
- Você não pode alterar nem editar as tags de um stream excluído.

Restrições de chaves de marcas

- Cada chave de marca deve ser exclusiva. Se você adicionar uma marca com uma chave que já estiver em uso, sua nova marca existente substituirá o par de chave-valor.
- Não é possível iniciar uma chave de tag com `aws:`, pois esse prefixo é reservado para uso pela AWS. A AWS cria tags que começam com esse prefixo em seu nome, mas você não pode editá-las ou excluí-las.
- As chaves de marca devem ter entre 1 e 128 caracteres Unicode.
- As chaves de marca devem conter os seguintes caracteres: letras Unicode, dígitos, espaço em branco e os seguintes caracteres especiais: `_ . / = + - @`.

Restrições de valor de marcas

- Os valores de marca devem ter entre 0 e 255 caracteres Unicode.
- Os valores de marca podem estar em branco. Caso contrário, elas devem conter os seguintes caracteres: letras Unicode, dígitos, espaço em branco e qualquer um dos seguintes caracteres especiais: `_ . / = + - @`.

Marcação de streams do Firehose usando a API Amazon Data Firehose

Você pode especificar tags ao invocar CreateDeliveryStreampara criar um novo stream do Firehose. Para streams existentes do Firehose, você pode adicionar, listar e remover tags usando as três operações a seguir:

- TagDeliveryStream
- ListTagsForDeliveryStream
- UntagDeliveryStream

Tutorial: Ingira registros de fluxo de VPC no Splunk usando o Amazon Data Firehose

Para ver um tutorial, consulte Ingerir registros de fluxo de VPC no Splunk usando o Amazon Data Firehose.

Solução de problemas do Amazon Data Firehose

Se o Firehose encontrar erros ao entregar ou processar dados, ele tentará novamente até que a duração da nova tentativa configurada expire. Se a duração da nova tentativa terminar antes que os dados sejam entregues com sucesso, o Firehose fará backup dos dados no bucket de backup S3 configurado. Se o destino for o Amazon S3 e a entrega falhar ou se a entrega no bucket S3 de backup falhar, o Firehose continuará tentando novamente até que o período de retenção termine. Para streams do `DirectPut` Firehose, o Firehose retém os registros por 24 horas. Para um stream do Firehose cuja fonte de dados é um stream de dados do Kinesis, você pode alterar o período de retenção conforme descrito em Alterando o período de retenção de dados.

Se a fonte de dados for um stream de dados do Kinesis, o Firehose repetirá as seguintes operações indefinidamente:, e. `DescribeStream` `GetRecords` `GetShardIterator`

Se o stream do Firehose usar`DirectPut`, verifique as `IncomingRecords` métricas `IncomingBytes` e para ver se há tráfego de entrada. Se você estiver usando o `PutRecord` ou o `PutRecordBatch`, certifique-se de detectar as exceções e tentar novamente. Recomendamos uma política de repetição com recuo exponencial com tremulação e diversas tentativas. Além disso, se você usar a `PutRecordBatch` API, certifique-se de que seu código verifique o valor de FailedPutCountna resposta mesmo quando a chamada da API for bem-sucedida.

Se o stream do Firehose usa um stream de dados do Kinesis como fonte, verifique as `IncomingRecords` métricas `IncomingBytes` e do stream de dados de origem. Além disso, certifique-se de que as `DataReadFromKinesisStream.Records` métricas `DataReadFromKinesisStream.Bytes` e estejam sendo emitidas para o stream do Firehose.

Para obter informações sobre como rastrear erros de entrega usando CloudWatch, consultethe section called “Monitoramento com CloudWatch registros”.

Problemas comuns

Aqui estão alguns problemas comuns e como você pode resolvê-los.

- O stream do Firehose não está disponível como destino para CloudWatch registros, CloudWatch eventos ou ações de AWS IoT — alguns AWS serviços só podem enviar mensagens e eventos para um stream do Firehose que esteja no mesmo stream. Região da AWS Verifique se o stream do Firehose está localizado na mesma região dos outros serviços.

- Sem dados no destino, apesar das boas métricas — Se não houver problemas de ingestão de dados e as métricas emitidas para o stream do Firehose parecerem boas, mas você não ver os dados no destino, verifique a lógica do leitor. Certifique-se de que o leitor esteja analisando corretamente todos os dados.

Solução de problemas do Amazon S3

Verifique o seguinte se os dados não forem entregues ao bucket do Amazon Simple Storage Service (Amazon S3).

- Verifique o Firehose `IncomingBytes` e `IncomingRecords` as métricas para garantir que os dados sejam enviados para seu stream do Firehose com sucesso. Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando métricas CloudWatch .
- Se a transformação de dados com o Lambda estiver ativada, verifique a `ExecuteProcessingSuccess` métrica Firehose para garantir que o Firehose tenha tentado invocar sua função do Lambda. Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando métricas CloudWatch .
- Verifique a `DeliveryToS3.Success` métrica do Firehose para ter certeza de que o Firehose tentou colocar dados no seu bucket do Amazon S3. Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando métricas CloudWatch .
- Habilite do registro de erros caso ele ainda não esteja habilitado e verifique se os logs de erros acusa falha de entrega. Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando registros CloudWatch .
- Se você ver uma mensagem de erro no registro dizendo “Firehose found InternalServerError when call Amazon S3 service”. A operação será repetida; se o erro persistir, entre em contato com a S3 para obter uma solução.” , isso pode ser devido ao aumento significativo nas taxas de solicitação em uma única partição no S3. Você pode otimizar os padrões de design do prefixo S3 para mitigar o problema. Para obter mais informações, consulte Padrões de Design de Práticas Recomendadas: Otimizando a Performance do Amazon S3. Se isso não resolver o problema, entre em contato com o AWS Support para obter mais assistência.
- Certifique-se de que o bucket do Amazon S3 especificado em seu stream do Firehose ainda exista.
- Se a transformação de dados com o Lambda estiver ativada, certifique-se de que a função Lambda especificada em seu stream do Firehose ainda exista.

- Certifique-se de que a função do IAM especificada em seu stream do Firehose tenha acesso ao seu bucket do S3 e à sua função Lambda (se a transformação de dados estiver ativada). Além disso, certifique-se de que a função do IAM tenha acesso ao grupo de CloudWatch registros e aos fluxos de registros para verificar os registros de erros. Para ter mais informações, consulte Conceda ao Amazon Data Firehose acesso a um destino do Amazon S3.
- Se você estiver usando transformação de dados, certifique-se de que a função do Lambda nunca retorne respostas cuja carga útil exceda 6 MB. Para obter mais informações, consulte Amazon Data FirehoseData Transformation.

Solução de problemas do Amazon Redshift

Verifique o seguinte se os dados não forem entregues ao cluster provisionado do Amazon Redshift ou ao grupo de trabalho do Amazon Redshift sem servidor.

Os dados são entregues no bucket do S3 antes de serem carregados no Amazon Redshift. Se os dados não forem entregues ao bucket do S3, consulte Solução de problemas do Amazon S3.

- Verifique a `DeliveryToRedshift.Success` métrica do Firehose para garantir que o Firehose tenha tentado copiar dados do seu bucket do S3 para o cluster provisionado do Amazon Redshift ou para o grupo de trabalho Amazon Redshift Serverless. Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando métricas CloudWatch .
- Habilite do registro de erros caso ele ainda não esteja habilitado e verifique se os logs de erros acusa falha de entrega. Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando registros CloudWatch .
- Verifique a `STL_CONNECTION_LOG` tabela do Amazon Redshift para ver se o Firehose pode fazer conexões bem-sucedidas. Nessa tabela, você conseguirá ver as conexões e os respectivos status com base em um nome de usuário. Para obter mais informações, consulte `STL_CONNECTION_LOG` no Guia do desenvolvedor de banco de dados do Amazon Redshift.
- Se a verificação anterior mostrar que as conexões estão sendo estabelecidas, verifique a tabela `STL_LOAD_ERRORS` do Amazon Redshift para saber o motivo da falha do comando COPY. Para obter mais informações, consulte `STL_LOAD_ERRORS` no Guia do desenvolvedor de banco de dados do Amazon Redshift.
- Certifique-se de que a configuração do Amazon Redshift em seu stream do Firehose seja precisa e válida.
- Certifique-se de que a função do IAM especificada em seu stream do Firehose possa acessar o bucket do S3 do qual o Amazon Redshift copia os dados e também a função Lambda para

transformação de dados (se a transformação de dados estiver ativada). Além disso, certifique-se de que a função do IAM tenha acesso ao grupo de CloudWatch registros e aos fluxos de registros para verificar os registros de erros. Para ter mais informações, consulte Conceda ao Amazon Data Firehose acesso a um destino do Amazon Redshift .

- Se seu cluster provisionado pelo Amazon Redshift ou grupo de trabalho Amazon Redshift Serverless estiver em uma nuvem privada virtual (VPC), certifique-se de que o cluster permita acesso a partir de endereços IP do Firehose. Para ter mais informações, consulte Conceda ao Amazon Data Firehose acesso a um destino do Amazon Redshift .
- Certifique-se de que o cluster provisionado do Amazon Redshift ou o grupo de trabalho do Amazon Redshift sem servidor esteja disponível publicamente.
- Se você estiver usando transformação de dados, certifique-se de que a função do Lambda nunca retorne respostas cuja carga útil exceda 6 MB. Para obter mais informações, consulte Amazon Data FirehoseData Transformation.

Solução de problemas do Amazon OpenSearch Service

Verifique o seguinte se os dados não forem entregues ao seu domínio OpenSearch de serviço.

É possível fazer o backup simultâneo dos dados no bucket do Amazon S3. Se os dados não forem entregues ao bucket do S3, consulte Solução de problemas do Amazon S3.

- Verifique o Firehose `IncomingBytes` e `IncomingRecords` as métricas para garantir que os dados sejam enviados para seu stream do Firehose com sucesso. Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando métricas CloudWatch .
- Se a transformação de dados com o Lambda estiver ativada, verifique a `ExecuteProcessingSuccess` métrica Firehose para garantir que o Firehose tenha tentado invocar sua função do Lambda. Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando métricas CloudWatch .
- Verifique a `DeliveryToAmazonOpenSearchService.Success` métrica Firehose para garantir que o Firehose tenha tentado indexar dados no cluster de serviços. OpenSearch Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando métricas CloudWatch .
- Habilite do registro de erros caso ele ainda não esteja habilitado e verifique se os logs de erros acusa falha de entrega. Para ter mais informações, consulte Monitorando o Amazon Data Firehose usando registros CloudWatch .

- Certifique-se de que a configuração do OpenSearch serviço em seu stream do Firehose seja precisa e válida.
- Se a transformação de dados com o Lambda estiver ativada, certifique-se de que a função Lambda especificada em seu stream do Firehose ainda exista. Além disso, certifique-se de que a função do IAM tenha acesso ao grupo de CloudWatch registros e aos fluxos de registros para verificar os registros de erros. Para obter mais informações, consulte Concessão FirehoseAccess a um destino OpenSearch de serviço público.
- Certifique-se de que a função do IAM especificada em seu stream do Firehose possa acessar seu cluster de OpenSearch serviços, bucket de backup do S3 e função Lambda (se a transformação de dados estiver ativada). Além disso, certifique-se de que a função do IAM tenha acesso ao grupo de CloudWatch registros e aos fluxos de registros para verificar os registros de erros. Para obter mais informações, consulte Concessão FirehoseAccess a um destino OpenSearch de serviço público.
- Se você estiver usando transformação de dados, certifique-se de que a função do Lambda nunca retorne respostas cuja carga útil exceda 6 MB. Para obter mais informações, consulte Amazon Data FirehoseData Transformation.
- O Amazon Data Firehosecurently não suporta a entrega de registros CloudWatch para o OpenSearch destino do Amazon Service porque a Amazon CloudWatch combina vários eventos de log em um registro Firehose e o OpenSearch Amazon Service não pode aceitar vários eventos de log em um registro. Como alternativa, você pode considerar o uso do filtro de assinatura do Amazon OpenSearch Service em CloudWatch registros.

Solução de problemas do Splunk

Verifique o seguinte se os dados não forem entregues ao endpoint do Splunk.

- Se sua plataforma Splunk estiver em uma VPC, certifique-se de que o Firehose possa acessá-la. Para obter mais informações, consulte Acesso ao Splunk na VPC.
- Se você usa um balanceador de AWS carga, certifique-se de que seja um Classic Load Balancer ou um Application Load Balancer. Além disso, habilite sessões fixas com base na duração com a expiração de cookies desativada para o Classic Load Balancer e a expiração é definida como máxima (7 dias) para o Application Load Balancer. Para obter informações sobre como fazer isso, consulte Duration-Based Session Stickiness for Classic Load Balancer ou an Application Load Balancer.

- Revise os requisitos da plataforma Splunk. O complemento Splunk para Firehose requer a versão 6.6.X ou posterior da plataforma Splunk. Para obter mais informações, consulte Complemento do Splunk para o Amazon Kinesis Firehose.
- Se você tiver um proxy (Elastic Load Balancing ou outro) entre o Firehose e o nó HTTP Event Collector (HEC), habilite sessões fixas para oferecer suporte a confirmações HEC (ACKs).
- Verifique se o token do HEC que você está usando é válido.
- Verifique se o token do HEC está ativado. Consulte Ativar e desativar os tokens do Coletor de eventos.
- Verifique se os dados que você está enviando ao Splunk estão formatados corretamente. Para obter mais informações, consulte Formatar eventos para o Coletor de eventos HTTP.
- Verifique se o token do HEC e o evento de entrada estão configurados com um índice válido.
- Quando um upload para o Splunk falhar devido a um erro do servidor a partir do nó HEC, a solicitação será automaticamente tentada novamente Se todas as novas tentativas falharem, o backup dos dados será feito no Amazon S3. Verifique se os dados aparecem no Amazon S3, o que é uma indicação dessa falha.
- Verifique se você habilitou a confirmação do indexador no token do HEC. Para obter mais informações, consulte Habilitar confirmação do indexador.
- Aumente o valor de `HECAcknowledgmentTimeoutInSeconds` na configuração de destino do Splunk do seu stream Firehose.
- Aumente o valor de `DurationInSeconds` under `RetryOptions` na configuração de destino do Splunk do seu stream do Firehose.
- Verifique o status do HEC.
- Se você estiver usando transformação de dados, certifique-se de que a função do Lambda nunca retorne respostas cuja carga útil exceda 6 MB. Para obter mais informações, consulte Amazon Data FirehoseData Transformation.
- Verifique se o parâmetro do Splunk chamado `ackIdleCleanup` está definido como `true`. Ele é "false" por padrão. Para definir o parâmetro como `true`, faça o seguinte:
  - Para uma implantação de nuvem gerenciada do Splunk, envie um caso usando o portal de suporte do Splunk. Nesse caso, peça para que o suporte do Splunk habilite o coletor de eventos HTTP, defina o `ackIdleCleanup` como `true` em `inputs.conf` e crie ou modifique um load balancer para ser usado com esse complemento.
  - Para uma implantação empresarial de Splunk distribuída, defina o parâmetro `ackIdleCleanup` como verdadeiro no arquivo `inputs.conf`. Para usuários do *nix, este arquivo está localizado

em `$SPLUNK_HOME/etc/apps/splunk_httpinput/local/`. Para usuários do Windows, está em `%SPLUNK_HOME%\etc\apps\splunk_httpinput\local\`.

- Para uma implantação empresarial Splunk de única instância, defina o parâmetro `ackIdleCleanup` como `true` no arquivo `inputs.conf`. Para usuários do *nix, este arquivo está localizado em `$SPLUNK_HOME/etc/apps/splunk_httpinput/local/`. Para usuários do Windows, está em `%SPLUNK_HOME%\etc\apps\splunk_httpinput\local\`.

- Certifique-se de que a função do IAM especificada em seu stream do Firehose possa acessar o bucket de backup do S3 e a função Lambda para transformação de dados (se a transformação de dados estiver ativada). Além disso, certifique-se de que a função do IAM tenha acesso ao grupo de CloudWatch registros e aos fluxos de registros para verificar os registros de erros. Para obter mais informações, consulte Conceder FirehoseAccess a um destino do Splunk.
- Consulte Solução de problemas do Splunk para o Amazon Kinesis Firehose.

Solução de problemas do Snowflake

Esta seção descreve as etapas comuns de solução de problemas ao usar o Snowflake como destino

Falha na criação do stream Firehose

Se a criação do stream Firehose falhar em um stream que entrega dados para um cluster Snowflake PrivateLink habilitado, isso indica que o VPCE-ID não pode ser acessado pelo Firehose. Isso pode ser devido a um dos seguintes motivos:

- VPCE-ID incorreto. Confirme se não há erros tipográficos.
- O Firehose não é compatível com URLs do Snowflake sem região na versão prévia. Forneça o URL usando o Localizador de contas do Snowflake. Consulte a documentação do Snowflake para obter mais detalhes.
- Confirme se o stream do Firehose foi criado na mesma AWS região da região do Snowflake.
- Se o problema persistir, entre em contato com o AWS suporte.

Falhas na entrega

Verifique o seguinte se os dados não estiverem sendo entregues à sua tabela Snowflake. Os dados de falha na entrega do Snowflake serão entregues ao bucket de erros do S3 junto com um código de erro e uma mensagem de erro que correspondem à carga útil. A seguir estão alguns cenários de

erro comuns. Para ver a lista completa de códigos de erro, consulteErros de entrega de dados do Snowflake.

- Código de erro: Snowflake. DefaultRoleMissing: indica que a função snowflake não está configurada durante a criação do stream Firehose. Se a função do Snowflake não estiver configurada, certifique-se de definir uma função padrão para a função especificada pelo usuário do Snowflake.
- Código de erro: Snowflake. ExtraColumns: indica que a inserção no Snowflake foi rejeitada devido às colunas extras na carga de entrada. As colunas que não estão presentes na tabela não devem ser especificadas. Observe que os nomes das colunas do Snowflake diferenciam maiúsculas de minúsculas. Se a entrega falhar com esse erro, apesar da coluna estar presente na tabela, certifique-se de que a maiúscula e minúscula do nome da coluna na carga de entrada corresponda ao nome da coluna declarado na definição da tabela.
- Código de erro: Snowflake. MissingColumns: indica que a inserção no Snowflake foi rejeitada devido à falta de colunas na carga de entrada. Certifique-se de que os valores sejam especificados para todas as colunas não anuláveis.
- Código de erro: Snowflake. InvalidInput: isso pode acontecer quando o Firehose falhou em analisar a carga de entrada fornecida em um formato JSON válido. Certifique-se de que a carga json esteja bem formada, não tenha aspas duplas extras, aspas, caracteres de escape etc. Atualmente, o Firehose suporta apenas um único item JSON como carga útil de registro; matrizes JSON não são suportadas.
- Código de erro: Snowflake. InvalidValue: indica que a entrega falhou devido ao tipo de dados incorreto na carga de entrada. Certifique-se de que os valores JSON especificados na carga de entrada estejam de acordo com o tipo de dados declarado na definição da tabela do Snowflake.
- Código de erro: Snowflake. InvalidTableType: indica que o tipo de tabela configurado no stream do Firehose não é suportado. Consulte as limitações (em Limitações) do streaming de snowpipe para ver as tabelas, colunas e tipos de dados compatíveis.

> **Note**
>
> Por qualquer motivo, se a definição da tabela ou as permissões da função forem alteradas no seu destino do Snowflake após a criação do stream do Firehose, o Firehose poderá levar alguns minutos para detectar essas alterações. Se você estiver vendo erros de entrega devido a isso, tente excluir e recriar o stream do Firehose.

Solução de problemas de acessibilidade do endpoint Firehose

Se a API Firehose encontrar um tempo limite, execute as etapas a seguir para testar a acessibilidade do endpoint:

- Verifique se as solicitações de API são feitas de um host em uma VPC. Todo o tráfego de uma VPC exige a configuração de um endpoint de VPC Firehose. Para obter mais informações, consulte Usando o Firehose com. AWS PrivateLink
- Se o tráfego estiver vindo de uma rede pública ou VPC com o endpoint Firehose VPC configurado em uma sub-rede específica, execute os seguintes comandos no host para verificar a conectividade da rede. O endpoint Firehose pode ser encontrado em pontos de extremidade e cotas do Firehose.
  - Use ferramentas como traceroute ou tcping para verificar se a configuração da rede está correta. Se isso falhar, verifique sua configuração de rede:

    Por exemplo: .

    ```
    traceroute firehose.us-east-2.amazonaws.com
    ```

    ou

    ```
    tcping firehose.us-east-2.amazonaws.com 443
    ```

  - Se parecer que a configuração de rede está correta e o comando a seguir falhar, verifique se a Amazon CA (Autoridade Certificadora) está na cadeia de confiança.

    Por exemplo: .

    ```
    curl firehose.us-east-2.amazonaws.com
    ```

  Se os comandos acima forem bem-sucedidos, tente usar a API novamente para ver se há uma resposta retornada da API.

Solução de problemas de endpoints HTTP

Esta seção descreve etapas comuns de solução de problemas ao lidar com o Amazon Data Firehose entregando dados para destinos genéricos de endpoints HTTP e destinos de parceiros, incluindo

Datadog, Dynatrace, LogicMonitor MongoDB, New Relic, Splunk ou Sumo Logic. Para os fins desta seção, todos os destinos aplicáveis são chamados de endpoints HTTP. Certifique-se de que a função do IAM especificada em seu stream do Firehose possa acessar o bucket de backup do S3 e a função Lambda para transformação de dados (se a transformação de dados estiver ativada). Além disso, certifique-se de que a função do IAM tenha acesso ao grupo de CloudWatch registros e aos fluxos de registros para verificar os registros de erros. Para obter mais informações, consulte Conceder acesso ao Firehose a um destino de endpoint HTTP.

> **Note**
>
> As informações nesta seção não se aplicam aos seguintes destinos: Splunk, OpenSearch Service, S3 e Redshift.

CloudWatch Registros

É altamente recomendável que você ative o CloudWatch Logging for Firehose. Os registros só são publicados quando há erros na entrega ao seu destino.

Exceções de destino

ErrorCode: HttpEndpoint.DestinationException

```
{
    "deliveryStreamARN": "arn:aws:firehose:us-east-1:123456789012:deliverystream/
ronald-test",
    "destination": "custom.firehose.endpoint.com...",
    "deliveryStreamVersionId": 1,
    "message": "The following response was received from the endpoint destination.
 413: {\"requestId\": \"43b8e724-dbac-4510-adb7-ef211c6044b9\", \"timestamp\":
 1598556019164, \"errorMessage\": \"Payload too large\"}",
    "errorCode": "HttpEndpoint.DestinationException",
    "processor": "arn:aws:lambda:us-east-1:379522611494:function:httpLambdaProcessing"
}
```

As exceções de destino indicam que o Firehose é capaz de estabelecer uma conexão com o endpoint e fazer uma solicitação HTTP, mas não recebeu um código de resposta 200. As respostas 2xx que não sejam as respostas 200 também resultarão em uma exceção de destino. O Amazon

Data Firehose registra o código de resposta e uma carga de resposta truncada recebida do endpoint configurado no Logs. CloudWatch Como o Amazon Data Firehose registra o código de resposta e a carga sem modificação ou interpretação, cabe ao endpoint fornecer o motivo exato pelo qual rejeitou a solicitação de entrega HTTP do Amazon Data Firehose. Veja a seguir as recomendações de solução de problemas mais comuns para essas exceções:

- 400: Indica que você está enviando uma solicitação incorreta devido a uma configuração incorreta do Amazon Data Firehose. Certifique-se de ter o URL, os atributos comuns, a codificação do conteúdo, a chave de acesso e as sugestões de buffer corretos para o seu destino. Consulte a documentação específica do destino sobre a configuração necessária.
- 401: Indica que a chave de acesso que você configurou para o stream do Firehose está incorreta ou ausente.
- 403: indica que a chave de acesso que você configurou para seu stream do Firehose não tem permissões para entregar dados ao endpoint configurado.
- 413: Indica que a carga útil da solicitação que o Amazon Data Firehose envia para o endpoint é muito grande para ser processada pelo endpoint. Tente reduzir a sugestão de buffer para o tamanho recomendado para o destino.
- 429: Indica que o Amazon Data Firehose está enviando solicitações em uma taxa maior do que a capacidade do destino. Ajuste a solicitação de buffer aumentando o tempo de armazenamento em buffer e/ou aumentando o tamanho do buffer (mas ainda dentro do limite do destino).
- 5xx: indica que há um problema com o destino. O serviço Amazon Data Firehose ainda está funcionando corretamente.

> **Important**
>
> Importante: embora essas sejam as recomendações comuns de solução de problemas, endpoints específicos podem ter motivos diferentes para fornecer os códigos de resposta e as recomendações específicas do endpoint devem ser seguidas primeiro.

Resposta inválida

ErrorCode: HttpEndpoint.InvalidResponseFromDestination

```
{
```

```
    "deliveryStreamARN": "arn:aws:firehose:us-east-1:123456789012:deliverystream/
ronald-test",
    "destination": "custom.firehose.endpoint.com...",
    "deliveryStreamVersionId": 1,
    "message": "The response received from the specified endpoint is invalid.
 Contact the owner of the endpoint to resolve the issue. Response for request
 2de9e8e9-7296-47b0-bea6-9f17b133d847 is not recognized as valid JSON or has unexpected
 fields. Raw response received: 200 {\"requestId\": null}",
    "errorCode": "HttpEndpoint.InvalidResponseFromDestination",
    "processor": "arn:aws:lambda:us-east-1:379522611494:function:httpLambdaProcessing"
}
```

Exceções de resposta inválidas indicam que o Amazon Data Firehose recebeu uma resposta inválida do destino do endpoint. A resposta deve estar em conformidade com as especificações de resposta ou o Amazon Data Firehose considerará a tentativa de entrega uma falha e reentregará os mesmos dados até que a duração da nova tentativa configurada seja excedida. O Amazon Data Firehose trata respostas que não seguem as especificações de resposta como falhas, mesmo que a resposta tenha um status 200. Se você estiver desenvolvendo um endpoint compatível com o Amazon Data Firehose, siga as especificações de resposta para garantir que os dados sejam entregues com sucesso.

Veja abaixo alguns dos tipos comuns de respostas inválidas e como corrigi-las:

- Campos JSON inválidos ou inesperados: indica que a resposta não pode ser desserializada adequadamente como JSON ou tem campos inesperados. Certifique-se de que a resposta não seja codificada por conteúdo.
- Ausente RequestId: indica que a resposta não contém um RequestiD.
- RequestId não corresponde: indica que o RequestiD na resposta não corresponde ao RequestiD de saída.
- Timestamp ausente: indica que a resposta não contém um campo de timestamp. O campo de timestamp deve ser um número e não uma string.
- Cabeçalho de tipo de conteúdo ausente: indica que a resposta não contém um cabeçalho “content-type: application/json”. Nenhum outro tipo de conteúdo é aceito.

> **Important**
>
> Importante: o Amazon Data Firehose só pode entregar dados para endpoints que seguem as especificações de solicitação e resposta do Firehose. Se você estiver configurando seu

destino para um serviço de terceiros, certifique-se de usar o endpoint correto compatível com o Amazon Data Firehose, que provavelmente será diferente do endpoint público de ingestão. Por exemplo, o endpoint Amazon Data Firehose do Datadog é https://aws-kinesis-http-intake.logs.datadoghq.com/, enquanto seu endpoint público é https://api.datadoghq.com/.

Outros erros comuns

Os códigos de erro e as definições adicionais estão listados abaixo.

- Código de erro: HttpEndpoint. RequestTimeout- Indica que o endpoint demorou mais de 3 minutos para responder. Se você for o proprietário do destino, diminua o tempo de resposta do endpoint de destino. Se você não for o proprietário do destino, entre em contato com o proprietário e pergunte se algo pode ser feito para reduzir o tempo de resposta (ou seja, diminuir a sugestão de buffer para que haja menos dados sendo processados por solicitação).
- Código de erro: HttpEndpoint. ResponseTooLarge- Indica que a resposta é muito grande. A resposta deve ter menos do que 1 MiB, incluindo cabeçalhos.
- Código de erro: HttpEndpoint. ConnectionFailed- Indica que não foi possível estabelecer uma conexão com o endpoint configurado. Isso pode ser devido a um erro de digitação no URL configurado, ao fato de o endpoint não estar acessível ao Amazon Data Firehose ou ao endpoint demorar muito para responder à solicitação de conexão.
- Código de erro: HttpEndpoint. ConnectionReset- Indica que uma conexão foi estabelecida, mas foi reiniciada ou fechada prematuramente pelo endpoint.
- Código de erro: HttpEndpoint .SSL HandshakeFailure - Indica que um handshake SSL não pôde ser concluído com êxito com o endpoint configurado.

Solução de problemas do MSK como fonte

Esta seção descreve as etapas comuns de solução de problemas ao usar o MSK como fonte

> **Note**
>
> Para solucionar problemas de processamento, transformação ou entrega do S3, consulte as seções anteriores

Falha da criação do hose

Verifique o seguinte se a criação do hose com MSK como fonte estiver falhando

- Verifique se o cluster do MSK de origem está no estado Ativo.
- Se você estiver usando conectividade privada, certifique-se de que o link privado no cluster esteja ativado

  Se você estiver usando conectividade pública, certifique-se de que o acesso público no cluster esteja ativado
- Se você estiver usando conectividade privada, certifique-se de adicionar uma política baseada em recursos que permita que o Firehose crie um link privado. Consulte também: Permissões entre contas do MSK
- Certifique-se de que o perfil na configuração de origem tenha permissão para ingerir dados do tópico do cluster
- Garanta que seus grupos de segurança da VPC permitam tráfego de entrada nas portas usadas pelos servidores de bootstrap do cluster

Hose suspenso

Verifique os pontos a seguir se o hose estiver no estado SUSPENSO

- Verifique se o cluster do MSK de origem está no estado Ativo.
- Verifique se o tópico de origem existe. Caso o tópico tenha sido excluído e recriado, você também precisará excluir e recriar o stream do Firehose.

Hose com contrapressão

O valor de DataReadFromSource .Backpressured será 1 quando BytesPerSecondLimit cada partição for excedida ou se o fluxo normal de entrega for lento ou interrompido.

- Se você estiver pressionando BytesPerSecondLimit , verifique a métrica DataReadFromSource .Bytes e solicite um aumento de limite.
- Verifique os CloudWatch registros, as métricas de destino, as métricas de transformação de dados e as métricas de conversão de formato para identificar os gargalos.

Atualidade incorreta de dados

A atualidade dos dados parece incorreta

- O Firehose calcula a atualidade dos dados com base no timestamp do registro consumido. Para garantir que esse timestamp seja registrado corretamente quando o registro do produtor persiste nos registros do agente do Kafka, defina a configuração do tipo de timestamp do tópico Kafka como `message.timestamp.type=LogAppendTime`.

Problemas de conexão do cluster MSK

O procedimento a seguir explica como você pode validar a conectividade com clusters MSK. Para obter detalhes sobre a configuração do cliente Amazon MSK, consulte Introdução ao uso do Amazon MSK no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

Para validar a conectividade com clusters MSK

1. Crie uma instância do Amazon EC2 baseada em UNIX (preferencialmente AL2). Se você tiver somente a conectividade VPC habilitada em seu cluster, certifique-se de que sua instância do EC2 seja executada na mesma VPC. Insira SSH na instância quando estiver disponível. Para obter mais informações, consulte este tutorial no Guia do usuário do Amazon EC2.
2. Instale o Java usando o gerenciador de pacotes Yum executando o comando a seguir. Para obter mais informações, consulte as instruções de instalação no Guia do usuário do Amazon Corretto 8.

   ```
   sudo yum install java-1.8.0
   ```

3. Instale o AWS cliente executando o comando a seguir.

   ```
   curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
   unzip awscliv2.zip
   sudo ./aws/install
   ```

4. Faça o download da versão 2.6* do cliente Apache Kafka executando o seguinte comando.

   ```
   wget https://archive.apache.org/dist/kafka/2.6.2/kafka_2.12-2.6.2.tgz
   tar -xzf kafka_2.12-2.6.2.tgz
   ```

5. Acesse o diretório `kafka_2.12-2.6.2/libs` e execute o seguinte comando para baixar o arquivo JAR do IAM do Amazon MSK.

```
wget https://github.com/aws/aws-msk-iam-auth/releases/download/v1.1.3/aws-msk-iam-auth-1.1.3-all.jar
```

6. Crie um `client.properties` arquivo na pasta bin do Kafka.
7. `awsRoleArn`Substitua pelo ARN da função que você usou em seu Firehose `SourceConfiguration` e verifique a localização do certificado. Permita que seu usuário AWS cliente assuma a função`awsRoleArn`. AWS o usuário cliente tentará assumir a função que você especificou aqui.

```
[ec2-user@ip-xx-xx-xx-xx bin]$ cat client.properties
security.protocol=SASL_SSL
sasl.mechanism=AWS_MSK_IAM
sasl.jaas.config=software.amazon.msk.auth.iam.IAMLoginModule required
 awsRoleArn="<role arn>" awsStsRegion="<region name>";
sasl.client.callback.handler.class=software.amazon.msk.auth.iam.IAMClientCallbackHandler
awsDebugCreds=true
ssl.truststore.location=/usr/lib/jvm/java-1.8.0-
openjdk-1.8.0.342.b07-1.amzn2.0.1.x86_64/jre/lib/security/cacerts
ssl.truststore.password=changeit
```

8. Execute o seguinte comando do Kafka para listar tópicos. Se sua conexão for pública, use os servidores Bootstrap de endpoint público. Se sua conexão for privada, use os servidores Bootstrap de endpoint privados.

```
bin/kafka-topics.sh --list --bootstrap-server <bootstrap servers> --command-config
 bin/client.properties
```

Se a solicitação for bem-sucedida, você deverá ver uma saída semelhante ao exemplo a seguir.

```
[ec2-user@ip-xx-xx-xx-xx kafka_2.12-2.6.2]$ bin/kafka-topics.sh --list --bootstrap-
server <bootstrap servers> --command-config bin/client.properties

[xxxx-xx-xx 05:49:50,877] WARN The configuration 'awsDebugCreds' was supplied but
 isn't a known config. (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration 'ssl.truststore.location' was
 supplied but isn't a known config.
 (org.apache.kafka.clients.admin.AdminClientConfig)
```

```
[xxxx-xx-xx 05:49:50,878] WARN The configuration 'sasl.jaas.config' was supplied
 but isn't a known config. (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration
 'sasl.client.callback.handler.class' was supplied but isn't a known config.
 (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:49:50,878] WARN The configuration 'ssl.truststore.password' was
 supplied but isn't a known config.
 (org.apache.kafka.clients.admin.AdminClientConfig)
[xxxx-xx-xx 05:50:21,629] WARN [AdminClient clientId=adminclient-1] Connection to
 node...
__amazon_msk_canary
__consumer_offsets
```

9. Se você tiver algum problema ao executar o script anterior, verifique se os servidores de bootstrap fornecidos estão acessíveis na porta especificada. Para fazer isso, você pode baixar e usar o telnet ou um utilitário similar, conforme mostrado no comando a seguir.

```
sudo yum install telnet
telnet <bootstrap servers><port>
```

Se a solicitação for bem-sucedida, você obterá a seguinte saída. Isso significa que você pode se conectar ao seu cluster MSK em sua VPC local e que os servidores de bootstrap estão íntegros na porta especificada.

```
Connected to ..
```

10. Se a solicitação não for bem-sucedida, verifique as regras de entrada no seu grupo de segurança da VPC. Como exemplo, você pode usar as propriedades a seguir na regra de entrada.

```
Type: All traffic
Port: Port used by the bootstrap server (e.g. 14001)
Source: 0.0.0.0/0
```

Tente novamente a conexão telnet, conforme mostrado na etapa anterior. Se você ainda não conseguir se conectar ou se sua conexão com o Firehose ainda estiver falhando, entre em contato com o suporte.AWS

Métrica de atualização de dados aumentando ou não emitida

A atualização dos dados é uma medida da atualidade dos dados no stream do Firehose. É a idade do registro de dados mais antigo no stream do Firehose, medida desde o momento em que o Firehose ingeriu os dados até o momento atual. O Firehose fornece métricas que você pode usar para monitorar a atualização dos dados. Para identificar a métrica de atualização de dados de um destino específico, consulte the section called “Monitoramento com CloudWatch métricas”.

Se você habilitar o backup de todos os eventos ou todos os documentos, monitore duas métricas de atualização de dados separadas: uma para o destino principal e outra para o backup.

Se a métrica de atualização de dados não estiver sendo emitida, isso significa que não há entrega ativa para o stream do Firehose. Isso acontece quando a entrega de dados está completamente bloqueada ou quando não há dados de entrada.

Se a métrica de atualização de dados estiver aumentando constantemente, isso significa que a entrega de dados está em atraso. Isso pode acontecer por um dos seguintes motivos.

- O destino não comporta a taxa de entrega. Se o Firehose encontrar erros transitórios devido ao alto tráfego, a entrega poderá ficar para trás. Isso pode acontecer para destinos diferentes do Amazon S3 (pode acontecer para OpenSearch Service, Amazon Redshift ou Splunk). Certifique-se de que o destino tenha capacidade suficiente para comportar o tráfego de entrada.
- O destino é lento. A entrega de dados pode ficar para trás se o Firehose encontrar alta latência. Monitore a métrica da latência do destino.
- A função do Lambda está lenta. Isso pode levar a uma taxa de entrega de dados menor que a taxa de ingestão de dados do stream Firehose. Se possível, melhore a eficiência da função do Lambda. Por exemplo, se a função executa a E/S de rede, use vários threads ou a E/S assíncrona para aumentar o paralelismo. Além disso, considere aumentar o tamanho da memória da função do Lambda para que a alocação de CPU possa aumentar de acordo. Isso pode levar a invocações do Lambda mais rápidas. Para obter informações sobre como configurar funções Lambda, consulte Configurando AWS funções Lambda.
- Há falhas durante a entrega de dados. Para obter informações sobre como monitorar erros usando o Amazon CloudWatch Logs, consultethe section called “Monitoramento com CloudWatch registros”.
- Se a fonte de dados do stream do Firehose for um stream de dados do Kinesis, a limitação pode estar ocorrendo. Verifique as métricas `ThrottledGetRecords`,

`ThrottledGetShardIterator` e `ThrottledDescribeStream`. Se houver vários consumidores conectados ao fluxo de dados do Kinesis, considere o seguinte:

  - Se as métricas `ThrottledGetRecords` e `ThrottledGetShardIterator` estiverem altas, recomendamos aumentar o número de estilhaços provisionados para o fluxo de dados.
  - Se `ThrottledDescribeStream` for alto, recomendamos que você adicione a `kinesis:listshards` permissão à função configurada em KinesisStreamSourceConfiguration.

- Dicas de baixa capacidade de buffer para o destino. Isso pode aumentar o número de viagens de ida e volta que o Firehose precisa fazer até o destino, o que pode atrasar a entrega. Considere aumentar o valor das dicas de buffer. Para obter mais informações, consulte BufferingHints.
- Uma longa duração para repetições pode gerar atrasos na entrega quando os erros são frequentes. Considere reduzir a duração das repetições. Além disso, monitore os erros e tente reduzi-los. Para obter informações sobre como monitorar erros usando o Amazon CloudWatch Logs, consultethe section called “Monitoramento com CloudWatch registros”.
- Se o destino for Splunk e `DeliveryToSplunk.DataFreshness` estiver alto, mas `DeliveryToSplunk.Success` parecer bom, o cluster do Splunk pode estar ocupado. Libere o cluster do Splunk se possível. Como alternativa, entre em contato com o AWS Support e solicite um aumento no número de canais que o Firehose está usando para se comunicar com o cluster Splunk.

Falha na conversão do formato de registro para o Apache Parquet

Isso acontece se você pegar dados do DynamoDB que incluem `Set` o tipo, transmiti-los por meio do Lambda para um stream do Firehose e usar AWS Glue Data Catalog an para converter o formato de registro em Apache Parquet.

Quando o AWS Glue rastreador indexa os tipos de dados do conjunto do DynamoDB (`StringSet`,, e`BinarySet`)`NumberSet`, ele os armazena no catálogo de dados como, e, respectivamente. `SET<STRING>` `SET<BIGINT>` `SET<BINARY>` No entanto, para que o Firehose converta os registros de dados para o formato Apache Parquet, ele requer os tipos de dados do Apache Hive. Como os tipos de conjunto não são tipos de dados válidos do Apache Hive, há falha na conversão. Para que a conversão funcione, atualize o catálogo de dados com os tipos de dados do Apache Hive. É possível fazer isso alterando `set` para `array` no catálogo de dados.

Para alterar um ou mais tipos de dados de **`set`** para **`array`** em um catálogo AWS Glue de dados

1. Faça login no AWS Management Console e abra o AWS Glue console em https://console.aws.amazon.com/glue/.
2. No painel esquerdo, no cabeçalho Data catalog (Catálogo de dados), escolha Tables (Tabelas).
3. Na lista de tabelas, escolha o nome da tabela na qual você precisa modificar um ou mais tipos de dados. Você será redirecionado para a página de detalhes da tabela
4. Escolha o botão Editar esquema no canto superior direito da página de detalhes.
5. Na coluna Data type (Tipo de dados), escolha o primeiro tipo de dados `set`.
6. Na lista suspensa Column type (Tipo de coluna), altere o tipo de `set` para `array`.
7. No ArraySchemacampo, insira`array<string>`,`array<int>`, ou`array<binary>`, dependendo do tipo de dados apropriado para seu cenário.
8. Selecione Atualizar.
9. Repita as etapas anteriores para converter outros tipos `set` em tipos `array`.
10. Escolha Salvar.

Cota do Amazon Data Firehose

O Amazon Data Firehose tem a seguinte cota.

- Com o Amazon MSK como fonte para o stream do Firehose, cada stream do Firehose tem uma cota padrão de 10 MB/seg de taxa de transferência de leitura por partição e tamanho máximo de registro de 10 MB. Você pode usar o aumento da cota de serviço para solicitar um aumento na cota padrão de 10 MB/seg de taxa de transferência de leitura por partição.
- Com o Amazon MSK como fonte para o stream do Firehose, há um tamanho máximo de registro de 6 MB se o AWS Lambda estiver ativado e um tamanho máximo de registro de 10 MB se o Lambda estiver desativado. AWS O Lambda limita seu registro de entrada para 6 MB, e o Amazon Data Firehose encaminha registros acima de 6 MB para um bucket S3 com erro. Se o Lambda estiver desativado, o Firehose limitará seu registro de entrada a 10 MB. Se o Amazon Data Firehose receber um tamanho de registro do Amazon MSK maior que 10 MB, o Amazon Data Firehose entregará esse registro ao bucket de erros do S3 e emitirá métricas do Cloudwatch para sua conta. Para obter mais informações sobre os limites do AWS Lambda, consulte: https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html.
- Quando o particionamento dinâmico em um stream do Firehose está ativado, há uma cota padrão de 500 partições ativas que podem ser criadas para esse stream do Firehose. A quantidade de partições ativas é o número total de partições ativas dentro do buffer de entrega. Por exemplo, se a consulta de particionamento dinâmico monta 3 partições por segundo e você tiver uma configuração de sugestão de buffer que aciona a entrega a cada 60 segundos, então, em média, você teria 180 partições ativas. Depois que os dados são entregues em uma partição, essa partição deixa de estar ativa. Você pode usar o formulário Amazon Data Firehose Limits para solicitar um aumento dessa cota para até 5.000 partições ativas por determinado stream do Firehose. Se precisar de mais partições, você pode criar mais streams do Firehose e distribuir as partições ativas entre elas.
- Quando o particionamento dinâmico em um stream do Firehose está ativado, uma taxa de transferência máxima de 1 GB por segundo é suportada para cada partição ativa.
- Cada conta terá a seguinte cota para o número de streams do Firehose por região:
  - Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Oregon), Europa (Irlanda), Ásia-Pacífico (Tóquio): 5.000 streams Firehose
  - Europa (Frankfurt), Europa (Londres), Ásia-Pacífico (Cingapura), Ásia-Pacífico (Sydney), Ásia-Pacífico (Seul), Ásia-Pacífico (Mumbai), AWS GovCloud (Oeste dos EUA), Canadá (Oeste), Canadá (Central): 2.000 streams Firehose

- Europa (Paris), Europa (Milão), Europa (Estocolmo), Ásia-Pacífico (Hong Kong), Ásia-Pacífico (Osaka), América do Sul (São Paulo), China (Ningxia), China (Pequim), Oriente Médio (Bahrein), (Leste dos EUA), África AWS GovCloud (Cidade do Cabo): 500 fluxos Firehose
- Europa (Zurique), Europa (Espanha), Ásia-Pacífico (Hyderabad), Ásia-Pacífico (Jacarta), Ásia-Pacífico (Melbourne), Oriente Médio (EAU), Israel (Tel Aviv), Oeste do Canadá (Calgary), Canadá (Central): 100 fluxos Firehose
- Se você exceder esse número, uma chamada para CreateDeliveryStreamresultará em uma `LimitExceededException` exceção. Para aumentar essa cota, é possível usar o Service Quotas, caso esteja disponíveis na sua região. Para obter mais informações sobre o uso de Service Quotas, consulte Solicitar um aumento de cota. Se as Cotas de Serviço não estiverem disponíveis na sua região, você pode usar o formulário Amazon Data Firehose Limits para solicitar um aumento.

- Quando o Direct PUT é configurado como fonte de dados, cada stream do Firehose fornece a seguinte cota e solicitações combinadas: PutRecordPutRecordBatch
  - Para Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon) e Europa (Irlanda): 500.000 registros/segundo, 2.000 solicitações/segundo e 5 MiB/segundo.
  - Para Leste dos EUA (Ohio), Oeste dos EUA (Norte da Califórnia), AWS GovCloud (Leste dos EUA), AWS GovCloud (Oeste dos EUA), Ásia-Pacífico (Hong Kong), Ásia-Pacífico (Mumbai), Ásia-Pacífico (Seul), Ásia-Pacífico (Cingapura), China (Pequim), China (Ningxia), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Canadá (Central), Oeste do Canadá (Calgary), Europa (Frankfurt), Europa (Londres), Europa (Paris), Europa (Estocolmo), Oriente Médio (Bahrein), América do Sul (São Paulo), África (Cidade do Cabo) e Europa (Milão): 100.000 registros/segundo, 1.000 solicitações/segundo e 1 MiB/segundo.

  Para solicitar um aumento na cota, use o formulário Amazon Data Firehose Limits. As três cotas são escaladas proporcionalmente. Por exemplo, se você aumentar a cota de throughput na região Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon) ou Europa (Irlanda) para 10 MiB/segundo, as outras duas cotas aumentarão para 4.000 solicitações/segundo e 1.000.000 de registros/segundo.

  > **Important**
  >
  > Se a cota aumentada for muito maior do que o tráfego em execução, isso ocasionará lotes de entrega pequenos para os destinos. Isso não é eficaz e pode ser dispendioso nos

serviços de destino. Certifique-se de aumentar a cota apenas para que corresponda ao tráfego atual e aumente-a ainda mais se o tráfego aumentar.

Important

Observe que registros de dados menores podem levar a custos mais altos. O preço de ingestão do Firehose é baseado no número de registros de dados que você envia para o serviço, multiplicado pelo tamanho de cada registro, arredondado para os 5 KB (5120 bytes) mais próximos. Portanto, para o mesmo volume de dados recebidos (bytes), se o número maior de registros recebidos for maior, o custo incorrido será maior. Por exemplo, se o volume total de dados recebidos for 5 MiB, enviar 5 MiB de dados em 5.000 registros custa mais do que enviar a mesma quantidade de dados usando 1.000 registros. Para obter mais informações, consulte Amazon Data Firehose na Calculadora.AWS

Note

Quando o Kinesis Data Streams é configurado como fonte de dados, essa cota não se aplica, e o Amazon Data Firehose aumenta e diminui a escala sem limite.

- Cada stream do Firehose armazena registros de dados por até 24 horas, caso o destino da entrega não esteja disponível e a origem esteja. DirectPut Se a fonte for o Kinesis Data Streams (KDS) e o destino não estiver disponível, os dados serão retidos de acordo com a configuração do KDS.
- O tamanho máximo de um registro enviado para o Amazon Data Firehose, antes da codificação base64, é de 1.000 KiB.
- A PutRecordBatchoperação pode levar até 500 registros por chamada ou 4 MiB por chamada, o que for menor. Essa cota não pode ser alterada.
- As operações a seguir podem fornecer até cinco invocações por segundo (esse é um limite fixo): `CreateDeliveryStream`, `DeleteDeliveryStream`, `DescribeDeliveryStream`, `ListDeliveryStreams`, `UpdateDestination`, `TagDeliveryStream`, `UntagDeliveryStream`, `ListTagsForDeliveryStream`, `StartDeliveryStreamEncryption`, `StopDeliveryStreamEncryption`.
- As dicas de intervalo do buffer variam de 60 a 900 segundos.

- Para entrega do Amazon Data Firehose para o Amazon Redshift, somente clusters do Amazon Redshift acessíveis ao público são suportados.
- O intervalo de duração da nova tentativa é de 0 segundos a 7.200 segundos para o Amazon Redshift OpenSearch e a entrega de serviços.
- O Firehose é compatível com as versões 1.5, 2.3, 5.1, 5.3, 5.5, 5.6 do Elasticsearch, bem como todas as versões 6.* e 7.* e Amazon Service 2.x até 2.11. OpenSearch
- Quando o destino é Amazon S3, Amazon Redshift ou OpenSearch Service, o Amazon Data Firehose permite até 5 invocações Lambda pendentes por fragmento. Para o Splunk, a cota é de 10 invocações pendentes do Lambda por fragmento.
- Você pode usar uma CMK do tipo CUSTOMER_MANAGED_CMK para criptografar até 500 streams do Firehose.

Apêndice - Especificações de solicitação e resposta de entrega de endpoint HTTP

Para que o Amazon Data Firehose entregue com sucesso dados para endpoints HTTP personalizados, esses endpoints devem aceitar solicitações e enviar respostas usando determinados formatos de solicitação e resposta do Amazon Data Firehose. Esta seção descreve as especificações de formato das solicitações HTTP que o serviço Amazon Data Firehose envia para endpoints HTTP personalizados, bem como as especificações de formato das respostas HTTP que o serviço Amazon Data Firehose espera. Os endpoints HTTP têm 3 minutos para responder a uma solicitação antes que o Amazon Data Firehose atinja o tempo limite dessa solicitação. O Amazon Data Firehose trata respostas que não seguem o formato adequado como falhas na entrega.

Tópicos

Formato de solicitação

Parâmetros de caminho e URL

Eles são configurados diretamente por você como parte de um único campo de URL. O Amazon Data Firehose os envia conforme configurados, sem modificação. Somente destinos https são compatíveis. As restrições de URL são aplicadas durante a configuração do fluxo de entrega.

> **Note**
>
> Atualmente, somente a porta 443 é compatível com a entrega de dados de endpoint HTTP.

Cabeçalhos HTTP: X-Amz-Firehose-Protocol-Version

Esse cabeçalho é usado para indicar a versão dos formatos de solicitação/resposta. Atualmente, a única versão é a 1.0.

Cabeçalhos HTTP: X-Amz-Firehose-Request-Id

O valor desse cabeçalho é um GUID opaco que pode ser usado para depuração e eliminação de duplicações. As implementações de endpoint devem registrar em log o valor desse cabeçalho, se possível, tanto para solicitações bem-sucedidas quanto malsucedidas. O ID da solicitação é mantido entre as várias tentativas da mesma solicitação.

Cabeçalhos HTTP: Content-Type

O valor do cabeçalho Content-Type é sempre `application/json`.

Cabeçalhos HTTP: Content-Encoding

Um stream do Firehose pode ser configurado para usar o GZIP para compactar o corpo ao enviar solicitações. Quando essa compactação está habilitada, o valor do cabeçalho Content-Encoding é definido como gzip, de acordo com a prática padrão. Se a compactação não estiver habilitada, o cabeçalho Content-Encoding estará totalmente ausente.

Cabeçalhos HTTP: Content-Length

Isso é usado da maneira padrão.

Cabeçalhos HTTP: X-Amz-Firehose-Source-Arn:

O ARN do stream Firehose representado no formato de string ASCII. O ARN codifica a região, o ID da AWS conta e o nome do stream. Por exemplo, `arn:aws:firehose:us-east-1:123456789:deliverystream/testStream`.

Cabeçalhos HTTP: X-Amz-Firehose-Access-Key

Esse cabeçalho carrega uma chave de API ou outras credenciais. Você pode criar ou atualizar a chave de API (também conhecida como token de autorização) ao criar ou atualizar o fluxo de entrega. O Amazon Data Firehose restringe o tamanho da chave de acesso a 4096 bytes. O Amazon Data Firehose não tenta interpretar essa chave de forma alguma. A chave configurada é copiada literalmente para o valor desse cabeçalho.

O conteúdo pode ser arbitrário e pode representar um token JWT ou uma ACCESS_KEY. Se um endpoint exigir credenciais com vários campos (por exemplo, nome de usuário e senha), os valores de todos os campos devem ser armazenados juntos em uma única chave de acesso em um formato que o endpoint entenda (JSON ou CSV). Esse campo pode ser codificado na base 64 se o conteúdo original for binário. O Amazon Data Firehose não modifica e/ou codifica o valor configurado e usa o conteúdo como está.

Cabeçalhos HTTP - X-Amz-Firehose-Common-Attributes

Esse cabeçalho transporta os atributos comuns (metadados) relativos à solicitação inteira e/ou a todos os registros dentro da solicitação. Eles são configurados diretamente por você ao criar um stream do Firehose. O valor desse atributo é codificado como um objeto JSON com o seguinte esquema:

```
"$schema": http://json-schema.org/draft-07/schema#

properties:
  commonAttributes:
    type: object
    minProperties: 0
    maxProperties: 50
    patternProperties:
      "^.{1,256}$":
        type: string
        minLength: 0
        maxLength: 1024
```

Veja um exemplo abaixo:

```
"commonAttributes": {
    "deployment -context": "pre-prod-gamma",
    "device-types": ""
  }
```

Corpo: tamanho máximo

O tamanho máximo do corpo é configurado por você e pode ter até 64 MiB, antes de compactado.

Corpo: esquema

O corpo leva um único documento JSON com o seguinte esquema JSON (escrito em YAML):

```
"$schema": http://json-schema.org/draft-07/schema#
```

```
title: FirehoseCustomHttpsEndpointRequest
description: >
  The request body that the Firehose service sends to
  custom HTTPS endpoints.
type: object
properties:
  requestId:
    description: >
      Same as the value in the X-Amz-Firehose-Request-Id header,
      duplicated here for convenience.
    type: string
  timestamp:
    description: >
      The timestamp (milliseconds since epoch) at which the Firehose
      server generated this request.
    type: integer
  records:
    description: >
      The actual records of the Firehose stream, carrying
      the customer data.
    type: array
    minItems: 1
    maxItems: 10000
    items:
      type: object
      properties:
        data:
          description: >
            The data of this record, in Base64. Note that empty
            records are permitted in Firehose. The maximum allowed
            size of the data, before Base64 encoding, is 1024000
            bytes; the maximum length of this field is therefore
            1365336 chars.
          type: string
          minLength: 0
          maxLength: 1365336

required:
  - requestId
  - records
```

Veja um exemplo abaixo:

```
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": 1578090901599
  "records": [
    {
      "data": "aGVsbG8="
    },
    {
      "data": "aGVsbG8gd29ybGQ="
    }
  ]
}
```

Formato de resposta

Comportamento padrão em caso de erro

Se uma resposta não estiver em conformidade com os requisitos abaixo, o servidor Firehose a tratará como se tivesse um código de status 500 sem corpo.

Código de status

O código de status do HTTP DEVE estar no intervalo 2XX, 4XX ou 5XX.

O servidor Amazon Data Firehose NÃO segue redirecionamentos (códigos de status 3XX). Somente o código de resposta 200 é considerado uma entrega bem-sucedida de registros para HTTP/EP. O código de resposta 413 (tamanho excedido) é considerado uma falha permanente, e o lote de registros não é enviado para o bucket de erros, se configurado. Todos os outros códigos de resposta são considerados erros passíveis de novas tentativas e estão sujeitos ao algoritmo de novas tentativas de recuo que será explicado posteriormente.

Cabeçalhos HTTP: tipo de conteúdo

O único tipo de conteúdo aceitável é aplicação/json.

Cabeçalhos HTTP: Content-Encoding

A codificação de conteúdo NÃO DEVE ser usada. O corpo DEVE estar descompactado.

Cabeçalhos HTTP: Content-Length

O cabeçalho Content-Length DEVE estar presente se a resposta tiver um corpo.

Corpo: tamanho máximo

O corpo da resposta deve ter no máximo 1 MiB.

```
"$schema": http://json-schema.org/draft-07/schema#

title: FirehoseCustomHttpsEndpointResponse

description: >
  The response body that the Firehose service sends to
  custom HTTPS endpoints.
type: object
properties:
  requestId:
    description: >
      Must match the requestId in the request.
    type: string

  timestamp:
    description: >
      The timestamp (milliseconds since epoch) at which the
      server processed this request.
    type: integer

  errorMessage:
    description: >
      For failed requests, a message explaining the failure.
      If a request fails after exhausting all retries, the last
      Instance of the error message is copied to error output
      S3 bucket if configured.
    type: string
    minLength: 0
    maxLength: 8192
required:
  - requestId
  - timestamp
```

Veja um exemplo abaixo:

```
Failure Case (HTTP Response Code 4xx or 5xx)
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": "1578090903599",
  "errorMessage": "Unable to deliver records due to unknown error."
}
Success case (HTTP Response Code 200)
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": 1578090903599
}
```

Lidar com respostas de erro

Em todos os casos de erro, o servidor Amazon Data Firehose tenta novamente a entrega do mesmo lote de registros usando um algoritmo de recuo exponencial. As novas tentativas são recuadas usando um tempo de recuo inicial (1 segundo) com um fator de instabilidade de (15%) e cada nova tentativa subsequente é recuada usando a fórmula (initial-backoff-time * (multiplicador (2) ^ retry_count)) com variação adicional. O tempo de recuo é limitado por um intervalo máximo de 2 minutos. Por exemplo, na 'n'-ésima repetição, o tempo de recuo é = MAX (120, 2^n) * aleatório (0,85, 1,15).

Os parâmetros especificados na equação anterior estão sujeitos a alterações. Consulte a documentação do AWS Firehose para ver o tempo exato de recuo inicial, o tempo máximo de recuo, o multiplicador e as porcentagens de instabilidade usadas no algoritmo de recuo exponencial.

Em cada nova tentativa subsequente, a chave de acesso e/ou o destino para o qual os registros são entregues podem mudar com base na configuração atualizada do stream do Firehose. O serviço Amazon Data Firehose usa o mesmo ID de solicitação em todas as novas tentativas da melhor maneira possível. Esse último atributo pode ser usado para eliminar duplicação pelo servidor de endpoint HTTP. Se a solicitação ainda não for entregue após o tempo máximo permitido (com base na configuração do stream do Firehose), o lote de registros poderá, opcionalmente, ser entregue a um bucket de erros com base na configuração do stream.

Exemplos

Exemplo de uma solicitação originada do CWLog:

```
{
  "requestId": "ed4acda5-034f-9f42-bba1-f29aea6d7d8f",
  "timestamp": 1578090901599,
  "records": [
   {
    "data": {
      "messageType": "DATA_MESSAGE",
      "owner": "123456789012",
      "logGroup": "log_group_name",
      "logStream": "log_stream_name",
      "subscriptionFilters": [
        "subscription_filter_name"
      ],
      "logEvents": [
        {
          "id": "01234567890123456789012345678901234567890123456",
          "timestamp": 1510109208016,
          "message": "log message 1"
        },
        {
          "id": "01234567890123456789012345678901234567890123456",
          "timestamp": 1510109208017,
          "message": "log message 2"
        }
      ]
    }
   }
  ]
}
```

Histórico do documento

A tabela a seguir descreve as mudanças importantes na documentação do Amazon Data Firehose.

Alteração	Descrição	Alterado em
Snowflake como destino em novas regiões	O Snowflake agora está disponível como destino na Ásia-Pacífico (Cingapura), Ásia-Pacífico (Seul) e Ásia-Pacífico (Sydney). Consulte the section called “Defina as configurações de destino para o Snowflake”.	19 de junho de 2024
O Amazon Data Firehose se integra com AWS Secrets Manager	Agora você pode acessar seus segredos e automatizar a rotação de credenciais com segurança com o Secrets Manager. Consulte the section called “Autenticar com AWS Secrets Manager”.	06 de junho de 2024
Foi adicionado suporte para ingestão de registros para o Dynatrace	Agora você pode enviar registros e eventos para a Dynatrace para análise posterior. Consulte the section called “Definir as configurações de destino para o Dynatrace”.	18 de abril de 2024
Versão de disponibilidade geral (GA) para Snowflake como destino	O Snowflake agora está geralmente disponível como destino. Consulte the section called “Defina as configurações de destino para o Snowflake”.	17 de abril de 2024
O Amazon Kinesis Data Firehose agora é conhecido como Amazon Data Firehose	O Amazon Kinesis Data Firehose foi renomeado para Amazon Data Firehose. Consulte O que é o Amazon Data Firehose?	9 de fevereiro de 2024
Adicionou Snowflake como	Você pode criar um stream do Firehose com o Snowflake como destino. Consulte the section called	19 de janeiro de 2024

Alteração	Descrição	Alterado em
destino (pré-visualização pública)	“Defina as configurações de destino para o Snowflake”.	
Adicionada descompressão automática de registros CloudWatch	Você pode ativar a descompactação em fluxos novos ou existentes para enviar CloudWatch dados de registros descompactados para destinos do Firehose. Consulte the section called “Escrevendo usando CloudWatch registros”.	15 de dezembro de 2023
Adicionada a Splunk Observability Cloud como destino	Você pode criar um stream do Firehose com o Splunk Observability Cloud como destino. Consulte the section called “Defina as configurações de destino para o Splunk Observability Cloud”.	3 de outubro de 2023
Adicionado o Amazon Managed Streaming for Apache Kafka como fonte de dados	Agora você pode configurar o Amazon MSK para enviar informações para um stream do Firehose. Consulte the section called “Gravar usando o Amazon MSK”.	26 de setembro de 2023
Foi adicionado suporte para o tipo DocumentID para o destino do serviço OpenSearch	Se OpenSearch Service for o destino do stream do Firehose, o tipo documentID indica o método para configurar o ID do documento. Os métodos suportados são ID do documento gerado pelo Firehose e ID do documento gerado pelo OpenSearch serviço. Consulte the section called “Definir configurações de destino”.	10 de maio de 2023
Adicionada compatibilidade com particionamento dinâmico	Foi adicionado suporte para particionamento dinâmico contínuo dos dados de streaming no Amazon Data Firehose. Consulte Particionamento dinâmico.	31 de agosto de 2021

Alteração	Descrição	Alterado em
Adicione um tópico sobre prefixos personalizados.	Adicionado um tópico sobre as expressões que você pode usar ao criar um prefixo personalizado para dados entregues ao Amazon S3. Consulte Prefixos personalizados do Amazon S3.	20 de dezembro de 2018
Novo tutorial do Amazon Data Firehose adicionado	Foi adicionado um tutorial que demonstra como enviar registros de fluxo da Amazon VPC para o Splunk por meio do Amazon Data Firehose. Consulte Tutorial: Ingira registros de fluxo de VPC no Splunk usando o Amazon Data Firehose.	30 de outubro de 2018
Foram adicionadas quatro novas regiões do Amazon Data Firehose	Adicionadas Paris, Mumbai, São Paulo e Londres. Para ter mais informações, consulte Cota do Amazon Data Firehose.	27 de junho de 2018
Foram adicionadas duas novas regiões do Amazon Data Firehose	Adicionadas Seul e Montreal. Para ter mais informações, consulte Cota do Amazon Data Firehose.	13 de junho de 2018
Novo recurso Kinesis Streams como fonte	Foi adicionado o Kinesis Streams como uma fonte potencial de registros para um stream do Firehose. Para ter mais informações, consulte Configurar origem e destino.	18 de agosto de 2017
Fazer a atualização para a documentação do console	O assistente de criação de stream do Firehose foi atualizado. Para ter mais informações, consulte Crie um stream do Firehose.	19 de julho de 2017
Nova transformação de dados	Você pode configurar o Amazon Data Firehose para transformar seus dados antes da entrega dos dados. Para ter mais informações, consulte Transformação de dados do Amazon Data Firehose.	19 de dezembro de 2016

Alteração	Descrição	Alterado em
Nova tentativa de COPY do Amazon Redshift	Você pode configurar o Amazon Data Firehose para repetir um comando COPY em seu cluster do Amazon Redshift se ele falhar. Para obter mais informações, consulte Crie um stream do Firehose, Entenda a entrega de dados do Amazon Data Firehose e Cota do Amazon Data Firehose.	18 de maio de 2016
Novo destino do Amazon Data Firehose, Amazon Service OpenSearch	Você pode criar um stream do Firehose com o Amazon OpenSearch Service como destino. Para obter mais informações, consulte Crie um stream do Firehose, Entenda a entrega de dados do Amazon Data Firehose e Conceda ao Amazon Data Firehose acesso a um destino de serviço público OpenSearch .	19 de abril de 2016
Novas CloudWatch métricas aprimoradas e recursos de solução de problemas	Atualização do Monitoramento do Amazon Data Firehose e do Solução de problemas do Amazon Data Firehose.	19 de abril de 2016
Novo agente do Kinesis aprimorado	Atualizado Escrevendo para o Amazon Data Firehose usando o Kinesis Agent.	11 de abril de 2016
Novos agentes do Kinesis	Adição do Escrevendo para o Amazon Data Firehose usando o Kinesis Agent.	2 de outubro de 2015
Lançamento inicial	Versão inicial do Amazon Data Firehose Developer Guide.	4 de outubro de 2015

Glossário da AWS

Para obter a terminologia mais recente da AWS, consulte o glossário da AWS na Referência do Glossário da AWS.