Logs do AmazonCloudWatch

Guia do usuário

Logs do AmazonCloudWatch: Guia do usuário

Table of Contents

O que é Amazon CloudWatch Logs?

Você pode usar o Amazon CloudWatch Logs para monitorar, armazenar e acessar seus arquivos de log de instâncias do Amazon Elastic Compute Cloud (Amazon EC2), do AWS CloudTrail, do Route 53 e de outras origens.

O CloudWatch Logs permite centralizar os logs de todos os sistemas, aplicativos e serviços da AWS que você usa em um único serviço altamente escalável. Em seguida, você pode facilmente visualizá-los, pesquisá-los por padrões ou códigos de erro específicos, filtrá-los com base em campos específicos ou arquivá-los de forma segura para análise futura. O CloudWatch Logs permite que você veja todos os seus logs, independentemente da origem, como um fluxo único e consistente de eventos ordenados por tempo, e você pode consultá-los e classificá-los com base em outras dimensões, agrupá-los por campos específicos, criar computações personalizadas com uma linguagem de consulta eficiente e visualizar dados de log em painéis.

Features

- Consultar seus dados de log – é possível usar o CloudWatch Logs Insights para pesquisar e analisar interativamente os dados de log. Pode realizar consultas para o ajudar a responder de forma mais eficiente e eficaz a problemas operacionais. CloudWatch Logs As perspetivas incluem uma linguagem de consulta concebida especificamente com alguns comandos simples, mas poderosos. Fornecemos exemplos de consultas, descrições de comandos, preenchimento automático de consultas e descoberta de campo de log para ajudar você a começar a usar. Os exemplos de consultas estão incluídos para diversos tipos de logs de serviço da AWS. Para começar, consulte o Analisar dados de log com o CloudWatch Logs Insights (p. 35).
- Monitorar logs de instâncias do Amazon EC2 – use o CloudWatch Logs para monitorar aplicativos e sistemas usando dados de log. Por exemplo, o CloudWatch Logs pode monitorar o número de erros que ocorrerem em seus logs de aplicativo e enviar para você uma notificação sempre que a taxa de erros exceder um limite especificado. O CloudWatch Logs usa seus dados de log para monitoramento. Assim, nenhuma alteração de código é necessária. Por exemplo, você pode monitorar os logs de aplicativo em relação a termos literais específicos (como "NullReferenceException") ou contar o número de ocorrências de um termo literal em uma determinada posição nos dados de log (como códigos de status "404" em um log de acesso do Apache). Quando o termo que você estiver procurando for encontrado, o CloudWatch Logs relatará os dados para uma métrica CloudWatch que você especificar. Os dados de log são criptografados em trânsito e em repouso. Para começar, consulte o Conceitos básicos com CloudWatch Logs (p. 5).
- Monitorar eventos registrados do AWS CloudTrail – você pode criar alarmes no CloudWatch e receber notificações de uma determinada atividade de API ao ser capturada pelo CloudTrail e usar a notificação para solucionar problemas. Para começar, consulte Envio de eventos do CloudTrail para o CloudWatch Logs no AWS CloudTrail User Guide.
- Retenção de logs – por padrão, os logs são mantidos indefinidamente e nunca expiram. Você pode ajustar a política de retenção para cada grupo de logs, mantendo a retenção indefinida ou escolhendo um período de retenção entre 10 anos e um dia.
- Arquivar dados de log – Você pode usar o CloudWatch Logs para armazenar seus dados de log em armazenamento altamente resiliente. O agente do CloudWatch Logs facilita o envio de dados de log alterados e não alterados de um host e para o serviço de log. Em seguida, você poderá acessar os dados de log brutos quando forem necessários.
- Registrar consultas de DNS do Route 53 – você pode usar o CloudWatch Logs para registrar informações sobre as consultas de DNS que o Route 53 recebe. Para obter mais informações, consulte Registro de consultas de DNS no Guia do desenvolvedor do Amazon Route 53.

Serviços da AWS relacionados

Os seguintes serviços são usados em conjunto com o CloudWatch Logs:

- O AWS CloudTrail é um serviço web que permite monitorar chamadas feitas para a API do CloudWatch Logs para a sua conta, incluindo as chamadas feitas pelo Console de gerenciamento da AWS, a AWS Command Line Interface (AWS CLI) e outros serviços. Quando o registro em log do CloudTrail é habilitado, o CloudTrail captura as chamadas de API em sua conta e fornece os arquivos de log para o bucket especificado do Amazon S3. Cada arquivo de log pode conter um ou mais registros, dependendo de quantas ações devem ser realizadas para atender a uma solicitação. Para obter mais informações sobre o AWS CloudTrail, consulte O que é o AWS CloudTrail? no AWS CloudTrail User Guide. Para obter um exemplo do tipo de dados que o CloudWatch grava em arquivos de log do CloudTrail, consulte Registrar chamadas de API do Amazon CloudWatch Logs no AWS CloudTrail (p. 149).
- AWS Identity and Access Management (IAM) é um serviço da web que ajuda você a controlar seguramente o acesso de seus usuários aos recursos da AWS. Use o IAM para controlar quem pode usar os recursos da AWS (autenticação) e quais recursos os usuários podem usar e de que maneira (autorização). Para obter mais informações, consulte O que é IAM? no Guia do usuário do IAM.
- Amazon Kinesis Data Streams é um serviço da Web que você pode usar para entrada e agregação de dados rápidas e contínuas. O tipo de dados usados inclui dados de log de infraestrutura de TI, logs de aplicativo, mídias sociais, feeds de dados de mercado e dados de sequência de cliques da Web. Como o tempo de resposta para a entrada e o processamento de dados é em tempo real, o processando geralmente é leve. Para obter mais informações, consulte O que é o Amazon Kinesis Data Streams? no Guia do desenvolvedor do Amazon Kinesis Data Streams.
- AWS Lambda é um serviço da Web que você pode usar para criar aplicativos que respondam rapidamente a novas informações. Faça upload do código do aplicativo como funções do Lambda e o Lambda executará seu código em uma infraestrutura de computação de alta disponibilidade e executará toda a administração de recursos de computação, incluindo manutenção do servidor e do sistema operacional, provisionamento da capacidade e escalabilidade automática, implantação de códigos e patches de segurança e monitoramento do código e registro em log. Tudo o que você precisa fazer é fornecer o código em uma das linguagens compatíveis com o Lambda. Para obter mais informações, consulte O que é o AWS Lambda? no AWS Lambda Developer Guide.

Pricing

Ao se cadastrar na AWS, você poderá começar a usar o CloudWatch Logs gratuitamente usando o Nível gratuito da AWS.

As taxas padrão aplicam-se a logs armazenados por outros serviços usando o CloudWatch Logs (por exemplo, os logs de fluxo do Amazon VPC e os logs do Lambda).

Para obter mais informações, consulte Definição de preço do Amazon CloudWatch.

Conceitos do Amazon CloudWatch Logs

A terminologia e os conceitos essenciais para compreender e usar o CloudWatch Logs são descritos abaixo.

Eventos de log

Evento de log é um registro de alguma atividade registrada pelo aplicativo ou recurso que está sendo monitorado. O registro de eventos de log que o CloudWatch Logs compreende contém duas propriedades: o time stamp de quando ocorreu o evento e a mensagem de eventos brutos. As mensagens de eventos devem estar codificadas por UTF-8.

Fluxos de log

Stream de log é uma sequência de eventos de log que compartilham a mesma origem. Mais especificamente, um stream de log geralmente representa a sequência de eventos que vem da instância do aplicativo ou do recurso que está sendo monitorado. Por exemplo, um stream de log pode estar associado a um log de acesso do Apache em um host específico. Quando você não precisar mais de um stream de log, poderá excluí-lo usando o comando aws logs delete-log-stream.

Grupos de logs

Os grupos de logs definem grupos de streams de log que compartilham as mesmas configurações de retenção, monitoramento e controle de acesso. Cada stream de log precisa pertencer a um grupo de logs. Por exemplo, se você tiver um stream de log separado para os logs de acesso do Apache a partir de cada host, poderá agrupar esses fluxos de log em um único grupo de log chamado `MyWebsite.com/Apache/access_log`.

Não há limite para o número de streams de log que podem pertencer a um grupo de logs.

Filtros de métrica

Você pode usar filtros de métrica para extrair as observações de métrica de eventos ingeridos e transformá-las em pontos de dados em uma métrica do CloudWatch. Filtros de métrica são atribuídos a grupos de logs, e todos os filtros atribuídas a um grupo de logs são aplicados a seus streams de log.

Configurações de retenção

As configurações de retenção podem ser usadas para especificar por quanto tempo os eventos de log serão mantidos no CloudWatch Logs. Os eventos de log expirados serão excluídos automaticamente. Assim como os filtros de métrica, as configurações de retenção também são atribuídas a grupos de logs, e a retenção atribuída a um grupo de logs é aplicada aos seus streams de log.

Configuração

Para usar o Amazon CloudWatch Logs, é necessária uma conta da AWS. Sua conta da AWS permite que você use serviços (por exemplo, o Amazon EC2) para gerar logs que você pode visualizar no console do CloudWatch, uma interface baseada na web. Além disso, você pode instalar e configurar a AWS Command Line Interface (AWS CLI).

Cadastrar-se na Amazon Web Services (AWS)

Quando você cria uma conta da AWS, cadastramos sua conta em todos os serviços da AWS automaticamente. Você será cobrado apenas pelos serviços que usar.

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

Para se cadastrar em uma conta AWS

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

   Parte do procedimento de cadastro envolve uma chamada telefônica e a digitação de um código de verificação usando o teclado do telefone.

Fazer login no console do Amazon CloudWatch

Para fazer login no console do Amazon CloudWatch

1. Faça login no Console de gerenciamento da AWS e abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Se necessário, altere a região. Na barra de navegação, selecione a região onde você tem seus recursos da AWS.
3. No painel de navegação, selecione Logs.

Configurar a interface de linha de comando

Você pode usar a AWS CLI para executar operações do CloudWatch Logs.

Para obter informações sobre como instalar e configurar a AWS CLI, consulte Configuração com a interface de linha de comando da AWS no Guia do usuário do AWS Command Line Interface.

Conceitos básicos com CloudWatch Logs

Para coletar logs de suas instâncias do Amazon EC2 e servidores no local para o CloudWatch Logs, a AWS oferece duas opções:

- Recomendado – o agente unificado do CloudWatch. Ele permite que você colete logs e métricas avançadas com um agente. Ele oferece suporte em sistemas operacionais, incluindo servidores que executam o Windows Server. Esse agente também proporciona melhor desempenho.

  Se você estiver usando o agente unificado para coletar métricas do CloudWatch, ele permitirá a coleta de métricas adicionais do sistema para proporcionar visibilidade de convidados. Ele também oferece suporte à coleta de métricas personalizadas usando `StatsD` ou `collectd`.

  Para obter mais informações, consulte Instalar o agente do CloudWatch no Guia do usuário do Amazon CloudWatch.
- Com suporte, mas no caminho para a depreciação – o agente mais antigo do CloudWatch Logs, que oferece suporte à coleta de logs de apenas servidores que executam o Linux. Se já estiver usando esse agente, você poderá continuar a fazê-lo. No entanto, o agente mais antigo exige o Python 2.7, 3.0 e 3.3. Como as instâncias atuais do EC2 não usam essas versões do Python e essas versões estão defasadas e não estão mais sendo corrigidas, é altamente recomendável que você migre para o agente unificado do CloudWatch.

  Ao migrar o agente do CloudWatch Logs para o agente unificado do CloudWatch, o assistente de configuração do agente unificado pode ler o arquivo de configuração do agente atual do CloudWatch Logs e configurar o novo agente para coletar os mesmos logs. Para obter mais informações sobre o assistente, consulte Criar o arquivo de configuração do agente do CloudWatch com o assistente no Guia do usuário do Amazon CloudWatch.

Tópicos

Usar o agente unificado do CloudWatch para começar a usar o CloudWatch Logs

Para obter mais informações sobre como usar o agente unificado do CloudWatch com o CloudWatch Logs, consulte Coletar métricas e logs de instâncias do Amazon EC2 e de servidores no local com o agente do CloudWatch no Guia do usuário do Amazon CloudWatch. Você conclui as etapas listadas nesta seção para instalar, configurar e iniciar o agente. Se não estiver usando o agente também para coletar métricas do CloudWatch, você poderá ignorar as seções que se referem às métricas.

Se você estiver usando o agente mais antigo do CloudWatch Logs e quiser migrar para usar o novo agente unificado, recomendamos que você use o assistente incluído no novo pacote do agente. Esse assistente pode ler o arquivo de configuração do agente do CloudWatch Logs atual e configurar o agente do CloudWatch para coletar os mesmos logs. Para obter mais informações sobre o assistente, consulte Criar o arquivo de configuração do agente do CloudWatch com o assistente no Guia do usuário do Amazon CloudWatch.

Usar o agente anterior do CloudWatch Logs para começar a usar o CloudWatch Logs

Usando o agente do CloudWatch Logs, você pode publicar dados de log de instâncias do Amazon EC2 que executam o Linux ou o Windows Server e eventos registrados em log do AWS CloudTrail. Em vez disso, recomendamos usar o agente unificado do CloudWatch para publicar seus dados de log. Para obter mais informações sobre o novo agente, consulte Coletar métricas e logs de instâncias do Amazon EC2 e de servidores no local com o agente do CloudWatch no Guia do usuário do Amazon CloudWatch. Como alternativa, você pode continuar a usar o agente anterior do CloudWatch Logs.

Tópicos

Pré-requisitos do agente do CloudWatch Logs

O agente do CloudWatch Logs requer o Python versão 2.7, 3.0 ou 3.3 e qualquer uma das seguintes versões do Linux:

- Amazon Linux versão 2014.03.02 ou posterior O Amazon Linux 2 não é compatível
- Ubuntu Server versão 12.04, 14.04 ou 16.04
- CentOS versão 6, 6.3, 6.4, 6.5 ou 7.0
- Red Hat Enterprise Linux (RHEL) versão 6.5 ou 7.0
- Debian 8.0

Início rápido Instalar e configurar o CloudWatch Logs Agente numa instância EC2 Linux em execução

> Tip
>
> O CloudWatch inclui um novo agente unificado que pode coletar logs e métricas de instâncias do EC2 e de servidores locais. Se você não estiver usando o agente do CloudWatch Logs mais antigo, recomendamos que use a versão mais recente do agente unificado do CloudWatch. Para obter mais informações, consulte Conceitos básicos com CloudWatch Logs (p. 5).
> O restante desta seção explica o uso do agente do CloudWatch Logs mais antigo.

Configurar o agente do CloudWatch Logs mais antigo em uma instância do EC2 do Linux em execução

Você pode usar o instalador do agente do CloudWatch Logs em uma instância do EC2 para instalar e configurar o agente do CloudWatch Logs. Depois que a instalação é concluída, os logs vão automaticamente da instância para o fluxo de logs que você cria enquanto instala o agente. O agente confirma que ele foi iniciado e permanece em execução até que você o desative.

Além de usar o agente, você também pode publicar dados de log usando a AWS CLI, o SDK do CloudWatch Logs ou a API do CloudWatch Logs. A AWS CLI é mais adequada para a publicação de dados na linha de comando ou por meio de scripts. O SDK do CloudWatch Logs é mais adequado para a publicação de dados de log diretamente de aplicativos ou a criação de seu próprio aplicativo de publicação de logs.

Etapa 1 Configure o seu IAM Função ou Utilizador para CloudWatch Logs

O agente do CloudWatch Logs oferece suporte a funções e usuários do IAM. Se a sua instância já tiver uma função do IAM associada, certifique-se de incluir a política do IAM abaixo. Se você ainda não tiver uma função do IAM atribuída à sua instância, poderá usar suas credenciais do IAM para as próximas etapas ou atribuir uma função do IAM a essa instância. Para obter mais informações, consulte Como associar uma função do IAM a uma instância.

Para configurar sua função ou usuário do IAM para o CloudWatch Logs

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles (Funções).
3. Escolha a função selecionando o nome (não marque a caixa de seleção ao lado do nome).
4. Escolha Attach Policies (Anexar políticas), Create Policy (Criar política).

   Uma nova guia ou janela de navegação é aberta.
5. Escolha a guia JSON e digite o seguinte documento de política JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "*"
    ]
  }
 ]
}
```

6. Ao concluir, selecione Revisar política. O Validador de política indica se há qualquer erro de sintaxe.
7. Na página Review Policy (Revisar política), digite um Name (Nome) e uma Description (Descrição) (opcional) para a política que você está criando. Revise o Resumo da política para ver as permissões que são concedidas pela política. Em seguida, escolha Criar política para salvar seu trabalho.
8. Feche a guia ou janela de navegador e retorne à página Add permissions (Adicionar permissões) da sua função. Escolha Refresh (Atualizar) e, em seguida, escolha a nova política para anexá-la à sua função.

9. Escolha Attach Policy.

Etapa 2. Instalar e configurar CloudWatch Logs num existente Amazon EC2 Instância

O processo de instalação do agente do CloudWatch Logs dependerá do sistema que sua instância do Amazon EC2 estiver executando, ou seja, Amazon Linux, Ubuntu, CentOS ou Red Hat. Use as etapas apropriadas para a versão do Linux na sua instância.

Para instalar e configurar o CloudWatch Logs em uma instância do Amazon Linux

A partir do Amazon Linux AMI 2014.09, o agente do CloudWatch Logs está disponível como uma instalação RPM com o pacote awslogs. As versões anteriores do Amazon Linux podem acessar o pacote awslogs atualizando sua instância com o comando `sudo yum update -y`. Ao instalar o pacote awslogs como um RPM, em vez de usar o instalador do CloudWatch Logs, sua instância receberá atualizações de pacote e patches regulares da AWS sem a necessidade de reinstalar manualmente o agente do CloudWatch Logs.

> Warning
>
> Não atualize o agente do CloudWatch Logs usando o método de instalação RPM se você já usou o script Python para instalar o agente. Isso pode causar problemas de configuração que impedirão que o agente do CloudWatch Logs envie seus logs para o CloudWatch.

1. Conecte-se à sua instância Amazon Linux. Para obter mais informações, consulte o tópico Conecte-se à sua instância no Guia do usuário do Amazon EC2 para instâncias do Linux.

   Para obter mais informações sobre problemas de conexão, consulte Resolução de problemas para se conectar à sua instância no Guia do usuário do Amazon EC2 para instâncias do Linux.
2. Atualize sua instância do Amazon Linux para receber as alterações mais recentes nos repositórios de pacote.

   ```
   sudo yum update -y
   ```
3. Instale o pacote `awslogs`: Este é o método recomendado para instalar awslogs nas instâncias do Amazon Linux.

   ```
   sudo yum install -y awslogs
   ```
4. Edite o arquivo `/etc/awslogs/awslogs.conf` para configurar os logs a serem monitorados. Para obter mais informações sobre a edição desse arquivo, consulte Referência do agente do CloudWatch Logs (p. 152).
5. Por padrão, o `/etc/awslogs/awscli.conf` aponta para a região us-east-1. Para enviar seus logs para uma região diferente, edite o arquivo `awscli.conf` e especifique essa região.
6. Inicie o serviço `awslogs`.

   ```
   sudo service awslogs start
   ```

   Se você está executando o Amazon Linux 2, inicie o serviço `awslogs` com o comando a seguir.

   ```
   sudo systemctl start awslogsd
   ```
7. (Opcional) Verifique o arquivo `/var/log/awslogs.log` para ver se há erros registrados ao iniciar o serviço.

8. (Opcional) Execute o comando a seguir para iniciar o serviço `awslogs` em cada inicialização do sistema.

```
sudo chkconfig awslogs on
```

   Se você estiver executando o Amazon Linux 2, use o comando a seguir para iniciar o serviço a cada inicialização do sistema.

```
sudo systemctl enable awslogsd.service
```

9. Você deve ver o grupo de logs recém-criado e o fluxo de logs no console do CloudWatch depois de alguns minutos de execução do agente.

   Para obter mais informações, consulte Visualizar os dados de log enviados para o CloudWatch Logs (p. 59).

Para instalar e configurar o CloudWatch Logs em uma instância do Ubuntu Server, CentOS ou Red Hat

Se você estiver usando um AMI que esteja executando o Ubuntu Server, o CentOS ou o Red Hat, use o procedimento a seguir para instalar manualmente o agente do CloudWatch Logs em sua instância.

1. Conecte-se à sua instância do EC2. Para obter mais informações, consulte o tópico Conecte-se à sua instância no Guia do usuário do Amazon EC2 para instâncias do Linux.

   Para obter mais informações sobre problemas de conexão, consulte Resolução de problemas para se conectar à sua instância no Guia do usuário do Amazon EC2 para instâncias do Linux.

2. Execute o instalador do agente do CloudWatch Logs usando uma das duas opções. Você pode executá-lo diretamente na internet ou fazer download dos arquivos e executá-lo de forma autônoma.

   > **Note**
   >
   > Se você estiver executando CentOS 6.x, Red Hat 6.x ou Ubuntu 12.04, use as etapas para baixar e executar o instalador autônomo. A instalação do agente do CloudWatch Logs diretamente da Internet não é compatível com esses sistemas.

   > **Note**
   >
   > No Ubuntu, execute `apt-get update` antes de executar os comandos a seguir.

   Para executá-lo diretamente na internet, use os seguintes comandos e siga as instruções:

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1
```

   Se o comando anterior não funcionar, tente o seguinte:

```
sudo python3 ./awslogs-agent-setup.py --region us-east-1
```

   Para fazer download e executá-lo de forma independente, use os estes comandos e siga as instruções:

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
```

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/AgentDependencies.tar.gz
 -O
```

```
tar xvf AgentDependencies.tar.gz -C /tmp/
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1 --dependency-path /tmp/
AgentDependencies
```

Você pode instalar o agente do CloudWatch Logs especificando as regiões da us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, or sa-east-1.

> **Note**
>
> Para obter mais informações sobre a versão atual e o histórico das versões de `awslogs-agent-setup`, consulte CHANGELOG.txt.

O instalador do agente do CloudWatch Logs requer determinadas informações durante a configuração. Antes de começar, você precisa saber qual arquivo de log monitorar e seu formato do time stamp. Você também deve ter as seguintes informações à mão.

Item	Description (Descrição)
ID de chave de acesso da AWS	Pressione Enter se estiver usando uma função do IAM. Caso contrário, digite o ID de chave de acesso da AWS.
Chave de acesso secreta da AWS	Pressione Enter se estiver usando uma função do IAM. Caso contrário, digite a chave de acesso secreta da AWS.
Nome da região padrão	Pressione Enter. O padrão é us-east-2. Você pode definir isso como us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, or sa-east-1.
Formato de saída padrão	Deixe em branco e pressione Enter.
Caminho do arquivo de log para upload	A localização do arquivo que contém os dados do log a ser enviado. O instalador sugere um caminho para você.
Nome do grupo de logs de destino	O nome para o seu grupo de logs. O instalador sugere um nome de grupo de logs para você.
Nome do stream de log de destino	Por padrão, esse é o nome do host. O instalador sugere um nome de host para você.
Formato de time stamp	Especifique o formato do time stamp no arquivo de log especificado. Escolha Personalizado para especificar seu próprio formato.
Posição inicial	Como é feito upload dos dados. Defina isso como start_of_file para fazer upload de tudo no arquivo de dados. Defina como end_of_file para fazer upload somente dos dados recém-acrescentados.

Depois de concluir essas etapas, o instalador pergunta se você deseja configurar outro arquivo de log. Você pode executar o processo quantas vezes quiser para cada arquivo de log. Se você não tiver mais arquivos de log para monitorar, escolha N quando o instalador solicitar a configuração de

outro log. Para obter mais informações sobre as configurações no arquivo de configuração do agente, consulte Referência do agente do CloudWatch Logs (p. 152).

Note

A configuração de várias fontes de log para enviar dados a um único stream de logs não é suportada.

3. Você deve ver o grupo de logs recém-criado e o fluxo de logs no console do CloudWatch depois de alguns minutos de execução do agente.

   Para obter mais informações, consulte Visualizar os dados de log enviados para o CloudWatch Logs (p. 59).

Início rápido Instalar e configurar o CloudWatch Logs Agente numa instância de EC2 Linux no lançamento

Tip

O agente do CloudWatch Logs mais antigo discutido nesta seção está em vias de ser defasado. É altamente recomendável que você use o novo agente unificado do CloudWatch que pode coletar logs e métricas. Além disso, o agente mais antigo do CloudWatch Logs requer o Python 3.3 ou anterior e essas versões não são instaladas em novas instâncias do EC2 por padrão. Para obter mais informações sobre o agente unificado do CloudWatch, consulte Instalar o agente do CloudWatch.
O restante desta seção explica o uso do agente do CloudWatch Logs mais antigo.

Instalar o agente do CloudWatch Logs mais antigo em uma instância do EC2 do Linux na execução

Você pode usar os dados do usuário do Amazon EC2, um recurso do Amazon EC2, que permite que as informações paramétricas sejam transmitidas para a instância durante a execução para instalar e configurar o agente do CloudWatch Logs nessa instância. Para transmitir as informações de instalação e configuração do agente do CloudWatch Logs para o Amazon EC2, você pode fornecer o arquivo de configuração em um local de rede, como um bucket do Amazon S3.

A configuração de várias fontes de log para enviar dados a um único stream de logs não é suportada.

Prerequisite

Crie um arquivo de configuração do agente que descreva todos os seus grupos e streams de logs. Trata-se de um arquivo de texto que descreve os arquivos de log a serem monitorados, bem como os grupos e os streams de logs para os quais será feito upload deles. O agente consome esse arquivo de configuração e inicia o monitoramento e o upload de todos os arquivos de log descritos nele. Para obter mais informações sobre as configurações no arquivo de configuração do agente, consulte Referência do agente do CloudWatch Logs (p. 152).

Veja a seguir o exemplo de um arquivo de configuração do agente para o Amazon Linux

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/messages]
file = /var/log/messages
log_group_name = /var/log/messages
log_stream_name = {instance_id}
```

```
datetime_format = %b %d %H:%M:%S
```

Veja a seguir uma amostra de um arquivo de configuração do agente para o Ubuntu

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/syslog]
file = /var/log/syslog
log_group_name = /var/log/syslog
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

Para configurar sua função do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Políticas, Criar política.
3. Na página Criar política, em Criar sua própria política, escolha Selecionar. Para obter mais informações sobre a criação de políticas personalizadas, consulte Políticas do IAM para o Amazon EC2 no Guia do usuário do Amazon EC2 para instâncias do Linux.
4. Na página Revisar política, em Nome da política, digite um nome para a política.
5. Em Documento da política, cole a política a seguir:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::myawsbucket/*"
            ]
        }
    ]
}
```

6. Selecione Create Policy (Criar política).
7. No painel de navegação, escolha Funções, Criar nova função.
8. Na página Definir nome da função, digite um nome para a função e escolha Próxima etapa.
9. Na página Selecionar tipo de função, escolha Selecionar ao lado de Amazon EC2.
10. Na página Anexar política, no cabeçalho da tabela escolha Tipo de política, Gerenciado pelo cliente.
11. Selecione a política do IAM que você acabou de criar e, em seguida, escolha Next Step (Próxima etapa).
12. Selecione Create Role.

Para obter mais informações sobre políticas e usuários do IAM, consulte Usuários e grupos do IAM e Gerenciamento de políticas do IAM no Guia do usuário do IAM.

Para executar uma nova instância e habilitar o CloudWatch Logs

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. Escolha Executar instância.

   Para obter mais informações, consulte Execução de uma instância no Guia do usuário do Amazon EC2 para instâncias do Linux.
3. No Passo 1: Escolha uma imagem da máquina Amazon (AMI) , selecione o tipo de instância Linux para iniciar e, em seguida, no Passo 2: Escolher um tipo de instância página, escolher Seguinte: Configurar os detalhes da instância

   Certifique-se de que nuvem-inic está incluído na sua Imagem da Máquina Amazon (AMI). Amazon Linux amis e amis para Ubuntu e RHEL já incluem cloud-init, mas centos e outros amis no AWS Marketplace pode não ser.
4. No Passo 3: Configurar detalhes da instância página, para Função de IE, seleccione IAM função criada.
5. Em Detalhes avançados, em Dados do usuário, cole o script a seguir na caixa. Em seguida, atualize esse script alterando o valor da opção -c para o local do seu arquivo de configuração do agente:

```
#!/bin/bash
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
chmod +x ./awslogs-agent-setup.py
./awslogs-agent-setup.py -n -r us-east-1 -c s3://DOC-EXAMPLE-BUCKET1/my-config-file
```

6. Faça todas as outras alterações para a instância, revise suas configurações de execução e, em seguida, escolha Iniciar.
7. Você deve ver o grupo de logs recém-criado e o fluxo de logs no console do CloudWatch depois de alguns minutos de execução do agente.

   Para obter mais informações, consulte Visualizar os dados de log enviados para o CloudWatch Logs (p. 59).

Início rápido Ativar Amazon EC2 Instâncias a executar o Windows Server 2016 para enviar registos para CloudWatch Logs Utilizar o CloudWatch Logs Agente

Tip

O CloudWatch inclui um novo agente unificado que pode coletar logs e métricas de instâncias do EC2 e de servidores locais. Recomendamos que você o mais recente agente unificado do CloudWatch. Para obter mais informações, consulte Conceitos básicos com CloudWatch Logs (p. 5).
O restante desta seção explica o uso do agente do CloudWatch Logs mais antigo.

Habilitar suas instâncias do Amazon EC2 executando o Windows Server 2016 para enviar logs para o CloudWatch Logs usando o agente do CloudWatch Logs mais antigo

Há vários métodos que você pode usar para habilitar instâncias executando o Windows Server 2016 para enviar logs para o CloudWatch Logs. As etapas desta seção usam o Run Command do Systems Manager. Para obter informações sobre outros métodos possíveis, consulte Enviar contadores de logs, eventos e desempenho para o Amazon CloudWatch.

Etapas

Download do arquivo de configuração de exemplo

Transfira o seguinte ficheiro de amostra para o seu computador: AWS.EC2.Windows.cloudwatch.json.

Configurar o arquivo JSON para o CloudWatch

Você determina os logs a serem enviados ao CloudWatch especificando suas opções em um arquivo de configuração. O processo de criação desse arquivo e a especificação de suas opções pode levar 30 minutos ou mais para serem concluídos. Após concluir essa tarefa uma vez, você pode reutilizar o arquivo de configuração em todas as instâncias.

Etapas

Etapa 1 Activar registos de relógio cloudwatch

Na parte superior do arquivo JSON, altere "false" para "true" em `IsEnabled`:

```
"IsEnabled": true,
```

Etapa 2. Configurar definições para CloudWatch

Especifique as credenciais, a região, um nome de grupo de logs e um namespace de fluxo de log. Isso permite que a instância envie dados de log ao CloudWatch Logs. Para enviar os mesmos dados de log para locais diferentes, você pode incluir seções adicionais com IDs exclusivos (por exemplo, "CloudWatchLogs2" e CloudWatchLogs3”) e uma região diferente para cada ID.

Para definir configurações para enviar dados de log ao CloudWatch Logs

1. No arquivo JSON, localize a seção `CloudWatchLogs`.

```
{
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. Deixe os campos `AccessKey` e `SecretKey` em branco. Configure as credenciais usando uma função do IAM.
3. Em `Region`, digite a região para a qual enviar os dados de log (por exemplo, `us-east-2`).
4. Em `LogGroup`, digite o nome do grupo de logs. Esse nome aparece na tela Log Groups (Grupos de logs) no console do CloudWatch.
5. Em `LogStream`, digite o fluxo de log de destino. Esse nome aparece na tela Log Groups > Streams (Grupos de logs > Fluxos) no console do CloudWatch.

   Se você usar `{instance_id}`, o padrão, o nome do fluxo de logs será o ID dessa instância.

   Se você especificar um nome de fluxo de logs que ainda não existe, o CloudWatch Logs o criará automaticamente. Você pode definir um nome de fluxo de log usando uma sequência literal, as variáveis predefinidas `{hostname}`, `{instance_id}` e `{ip_address}` ou uma combinação delas.

Etapa 3 Configurar os dados para enviar

Você pode enviar dados de log de eventos, dados do rastreamento de eventos para Windows (ETW – Event Tracing for Windows) e outros dados de log ao CloudWatch Logs.

Para enviar dados de log de eventos de aplicativo do Windows para o CloudWatch Logs

1. No arquivo JSON, localize a seção `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. Em `Levels`, especifique o tipo de mensagens das quais fazer upload. Você pode especificar um dos seguintes valores:

   - **1** – fazer upload apenas de mensagens de erro.
   - **2** – fazer upload apenas de mensagens de aviso.
   - **4** – fazer upload apenas de mensagens informativas.

   Você pode combinar valores para incluir mais de um tipo de mensagem. Por exemplo, o valor **3** faz upload de mensagens de erro (**1**) e de mensagens de aviso (**2**). Um valor **7** faz upload de mensagens de erro (**1**), de mensagens de aviso (**2**) e de mensagens informativas (**4**).

Para enviar dados de log de segurança para o CloudWatch Logs

1. No arquivo JSON, localize a seção `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. Em `Levels`, digite **`7`** para fazer upload de todas as mensagens.

Para enviar dados de log de eventos do sistema para o CloudWatch Logs

1. No arquivo JSON, localize a seção `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. Em `Levels`, especifique o tipo de mensagens das quais fazer upload. Você pode especificar um dos seguintes valores:

   - **`1`** – fazer upload apenas de mensagens de erro.
   - **`2`** – fazer upload apenas de mensagens de aviso.
   - **`4`** – fazer upload apenas de mensagens informativas.

   Você pode combinar valores para incluir mais de um tipo de mensagem. Por exemplo, o valor **`3`** faz upload de mensagens de erro (**`1`**) e de mensagens de aviso (**`2`**). Um valor **`7`** faz upload de mensagens de erro (**`1`**), de mensagens de aviso (**`2`**) e de mensagens informativas (**`4`**).

Para enviar outros tipos de dados de log de eventos para o CloudWatch Logs

1. No arquivo JSON, adicione uma nova seção. Cada seção deve ter um `Id` exclusivo.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. Em `Id`, digite o nome do log para fazer upload (por exemplo, **`WindowsBackup`**).
3. Em `LogName`, digite o nome do log do qual fazer upload. Você pode localizar o nome do log da seguinte forma.

a. Abra o Visualizador de eventos.
b. No painel de navegação, escolha Applications and Services Logs.
c. Navegue até o log e escolha Actions, Properties.

4. Em `Levels`, especifique o tipo de mensagens das quais fazer upload. Você pode especificar um dos seguintes valores:

   - **`1`** – fazer upload apenas de mensagens de erro.
   - **`2`** – fazer upload apenas de mensagens de aviso.
   - **`4`** – fazer upload apenas de mensagens informativas.

   Você pode combinar valores para incluir mais de um tipo de mensagem. Por exemplo, o valor **`3`** faz upload de mensagens de erro (**`1`**) e de mensagens de aviso (**`2`**). Um valor **`7`** faz upload de mensagens de erro (**`1`**), de mensagens de aviso (**`2`**) e de mensagens informativas (**`4`**).

Para enviar os dados de rastreamento de eventos para Windows ao CloudWatch Logs

O ETW (Rastreamento de Eventos para Windows) fornece um mecanismo de registro eficiente e detalhado no qual os aplicativos podem gravar logs. Cada ETW é controlado por um gerente de sessão que pode iniciar e parar a sessão de registro. Cada sessão tem um provedor e um ou mais consumidores.

1. No arquivo JSON, localize a seção `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. Em `LogName`, digite o nome do log do qual fazer upload.
3. Em `Levels`, especifique o tipo de mensagens das quais fazer upload. Você pode especificar um dos seguintes valores:

   - **`1`** – fazer upload apenas de mensagens de erro.
   - **`2`** – fazer upload apenas de mensagens de aviso.
   - **`4`** – fazer upload apenas de mensagens informativas.

   Você pode combinar valores para incluir mais de um tipo de mensagem. Por exemplo, o valor **`3`** faz upload de mensagens de erro (**`1`**) e de mensagens de aviso (**`2`**). Um valor **`7`** faz upload de mensagens de erro (**`1`**), de mensagens de aviso (**`2`**) e de mensagens informativas (**`4`**).

Para enviar logs personalizados (qualquer arquivo de log baseado em texto) ao CloudWatch Logs

1. No arquivo JSON, localize a seção `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
```

```
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. Em `LogDirectoryPath`, digite o caminho onde os logs estão armazenados na instância.
3. Em `TimestampFormat`, digite o formato do time stamp a ser usado. Para obter mais informações sobre os valores compatíveis, consulte o tópico Cadeias de caracteres de formato de data e hora personalizado no MSDN.

   Important

   Seu arquivo de log de origem deve ter o time stamp no início de cada linha do log e deve haver um espaço depois do time stamp.
4. Em `Encoding`, digite a codificação de arquivo a ser usada (por exemplo, UTF-8). Para obter uma lista de valores compatíveis, consulte o tópico Classe Encoding no MSDN.

   Note

   Use o nome da codificação, não o nome da exibição.
5. (Opcional) Em `Filter`, digite o prefixo de nomes de logs. Deixe esse parâmetro em branco para monitorar todos os arquivos. Para obter mais informações sobre os valores compatíveis, consulte o tópico Propriedade FileSystemWatcherFilter no MSDN.
6. (Opcional) Em `CultureName`, digite a localidade na qual o time stamp é registrado no log. Se `CultureName` ficar em branco, será adotada como padrão a mesma localidade usada atualmente pela instância do Windows. Para obter mais informações a respeito, consulte a coluna `Language tag` na tabela no tópico Product Behavior no MSDN.

   Note

   O `div`, `div-MV`, `hu`, e `hu-HU` os valores não são suportados.
7. (Opcional) Para `TimeZoneKind`, tipo `Local` ou `UTC`. Pode definir esta informação para fornecer informações sobre o fuso horário quando não estiver incluído informação de fuso horário no carimbo de hora do seu registo. Se esse parâmetro for deixado em branco e se o time stamp não incluir informações sobre fuso horário, o CloudWatch Logs adotará o fuso horário local como o padrão. Esse parâmetro será ignorado se o time stamp já incluir informações sobre o fuso horário.
8. (Opcional) Em `LineCount`, digite o número de linhas do cabeçalho para identificar o arquivo de log. Por exemplo, os arquivos de log do IIS têm cabeçalhos praticamente idênticos. Você pode especificar **5**, que determina a leitura das três primeiras linhas do cabeçalho do arquivo de log para identificá-lo. Nos arquivos de log do IIS, a terceira linha é o carimbo de data/hora, mas nem sempre é garantido que o carimbo será diferente entre os arquivos de log. Por esse motivo, recomendamos incluir pelo menos uma linha de dados de log reais para identificar o arquivo de log de forma exclusiva.

Para enviar dados de log do IIS ao CloudWatch Logs

1. No arquivo JSON, localize a seção `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
```

```
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. Em `LogDirectoryPath`, digite a pasta onde os logs do IIS são armazenados para um site individual (por exemplo, `C:\inetpub\logs\LogFiles\W3SVC`*n*).

   Note

   Há suporte apenas para o formato de log W3C. Não há suporte para os formatos IIS, NCSA e Personalizado.

3. Em `TimestampFormat`, digite o formato do time stamp a ser usado. Para obter mais informações sobre os valores compatíveis, consulte o tópico Cadeias de caracteres de formato de data e hora personalizado no MSDN.
4. Em `Encoding`, digite a codificação de arquivo a ser usada (por exemplo, UTF-8). Para obter mais informações sobre os valores compatíveis, consulte o tópico Encoding Class no MSDN.

   Note

   Use o nome da codificação, não o nome da exibição.

5. (Opcional) Em `Filter`, digite o prefixo de nomes de logs. Deixe esse parâmetro em branco para monitorar todos os arquivos. Para obter mais informações sobre os valores compatíveis, consulte o tópico Propriedade FileSystemWatcherFilter no MSDN.
6. (Opcional) Em `CultureName`, digite a localidade na qual o time stamp é registrado no log. Se `CultureName` ficar em branco, será adotada como padrão a mesma localidade usada atualmente pela instância do Windows. Para obter mais informações sobre os valores compatíveis, consulte a coluna `Language tag` na tabela no tópico Product Behavior no MSDN.

   Note

   O `div`, `div-MV`, `hu`, e `hu-HU` os valores não são suportados.

7. (Opcional) Para `TimeZoneKind`, enter `Local` ou `UTC`. Pode definir esta informação para fornecer informações sobre o fuso horário quando não estiver incluído informação de fuso horário no carimbo de hora do seu registo. Se esse parâmetro for deixado em branco e se o time stamp não incluir informações sobre fuso horário, o CloudWatch Logs adotará o fuso horário local como o padrão. Esse parâmetro será ignorado se o time stamp já incluir informações sobre o fuso horário.
8. (Opcional) Em `LineCount`, digite o número de linhas do cabeçalho para identificar o arquivo de log. Por exemplo, os arquivos de log do IIS têm cabeçalhos praticamente idênticos. Você pode especificar **`5`**, que leria as cinco primeiras linhas do cabeçalho do arquivo de log para identificá-lo. Nos arquivos de log do IIS, a terceira linha é o carimbo de data/hora, mas nem sempre é garantido que o carimbo será diferente entre os arquivos de log. Por esse motivo, recomendamos incluir pelo menos uma linha de dados de log reais para identificar o arquivo de log de forma exclusiva.

Etapa 4. Configurar controlo de fluxo

Cada tipo de dados deve ter um destino correspondente na seção `Flows`. Por exemplo, para enviar o log personalizado, o log do ETW e o log do sistema ao CloudWatch Logs, adicione `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` à seção do `Flows`.

Warning

A adição de uma etapa inválida bloqueia o fluxo. Por exemplo, se você adicionar uma etapa de métrica de disco, mas a instância não tiver um disco, todas as etapas do fluxo serão bloqueadas.

Você pode enviar o mesmo arquivo de log a mais de um destino. Por exemplo, para enviar o log do aplicativo a dois destinos diferentes que você definiu na seção `CloudWatchLogs`, adicione `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)` à seção `Flows`.

Para configurar o controle de fluxo

1. No arquivo `AWS.EC2.Windows.CloudWatch.json`, localize a seção `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. Em `Flows`, adicione cada tipo de dados os quais serão feito upload (por exemplo, `ApplicationEventLog`) e seu destino (por exemplo, `CloudWatchLogs`).

Etapa 5. Guardar conteúdo JSON

Agora você concluiu a edição do arquivo JSON. Salve-o e cole o conteúdo do arquivo em um editor de texto em outra janela. Você precisará do conteúdo do arquivo em uma etapa posterior deste procedimento.

Criar um usuário e uma função do IAM para o Systems Manager

Uma função do IAM para credenciais de instância é necessária quando você usa o Run Command do Systems Manager. Essa função permite que o Systems Manager execute ações na instância. Opcionalmente, você pode criar uma conta de usuário exclusiva do IAM para configurar e executar o Systems Manager. Para obter mais informações, consulte Configuração de funções de segurança para o Systems Manager no Guia do usuário do AWS Systems Manager. Para obter informações sobre como anexar uma função do IAM a uma instância, consulte Anexar uma função do IAM a uma instância no Guia do usuário do Amazon EC2 para instâncias do Windows.

Verifique os pré-requisitos do Systems Manager

Antes de usar o Run Command do Systems Manager para configurar a integração com o CloudWatch Logs, verifique se as instâncias atendem aos requisitos mínimos. Para obter mais informações, consulte Pré-requisitos do Systems Manager no Guia do usuário do AWS Systems Manager.

Verificar o acesso à Internet

As instâncias do Amazon EC2 do Windows Server e as instâncias gerenciadas devem ter acesso de saída à Internet para enviar dados de log e de eventos ao CloudWatch. Para obter mais informações sobre como configurar o acesso à Internet, consulte Gateways da Internet no Guia do usuário da Amazon VPC.

Habilitar o CloudWatch Logs usando o Run Command do Systems Manager

O Run Command permite gerenciar a configuração de suas instâncias sob demanda. Você especifica um documento do Systems Manager, especifica parâmetros e executa o comando em uma ou mais instâncias. O agente do SSM na instância processa o comando e configura a instância conforme especificado.

Para configurar a integração com o CloudWatch Logs usando o Run Command

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. Abra o console do SSM em https://console.aws.amazon.com/systems-manager.
3. No painel de navegação, escolha Run Command (Executar comando).

4. Escolha Run a command.
5. Em Command document, escolha AWS-ConfigureCloudWatch.
6. Em Target instances (Instâncias de destino), escolha as instâncias a serem integradas ao CloudWatch Logs. Se você não vir uma instância nessa lista, ela poderá não estar configurada para Run Command. Para obter mais informações, consulte Pré-requisitos do Systems Manager no Guia do usuário do Amazon EC2 para instâncias do Windows.
7. Em Status, escolha Enabled.
8. Em Propriedades, copie e cole o conteúdo JSON que você criou nas tarefas anteriores.
9. Preencha os campos opcionais restantes e escolha Run.

Use o seguinte procedimento para visualizar os resultados da execução do comando no console do Amazon EC2.

Para visualizar a saída do comando no console

1. Selecione um comando.
2. Escolha a guia Output.
3. Escolha View Output. A página de saída do comando mostra os resultados da execução do comando.

Início rápido Habilitar suas instâncias do Amazon EC2 que executam o Windows Server 2012 e o Windows Server 2008 para enviar logs ao CloudWatch Logs

> Tip
>
> O CloudWatch inclui um novo agente unificado que pode coletar logs e métricas de instâncias do EC2 e de servidores locais. Recomendamos que você o mais recente agente unificado do CloudWatch. Para obter mais informações, consulte Conceitos básicos com CloudWatch Logs (p. 5).
> O restante desta seção explica o uso do agente do CloudWatch Logs mais antigo.

Habilitar suas instâncias do Amazon EC2 que executam o Windows Server 2012 e o Windows Server 2008 para enviar logs ao CloudWatch Logs

Use as seguintes etapas para habilitar suas instâncias que executam o Windows Server 2012 e o Windows Server 2008 a enviar logs ao CloudWatch Logs.

Download do arquivo de configuração de exemplo

Transfira o seguinte ficheiro JSON de amostra para o seu computador: AWS.EC2.Windows.cloudwatch.json. Você o edita nas etapas a seguir.

Configurar o arquivo JSON para o CloudWatch

Você determina os logs a serem enviados ao CloudWatch especificando suas opções no arquivo de configuração JSON. O processo de criação desse arquivo e a especificação de suas opções pode levar 30 minutos ou mais para serem concluídos. Após concluir essa tarefa uma vez, você pode reutilizar o arquivo de configuração em todas as instâncias.

Etapas

Etapa 1 Activar registos de relógio cloudwatch

Na parte superior do arquivo JSON, altere "false" para "true" em `IsEnabled`:

```
"IsEnabled": true,
```

Etapa 2. Configurar definições para CloudWatch

Especifique as credenciais, a região, um nome de grupo de logs e um namespace de fluxo de log. Isso permite que a instância envie dados de log ao CloudWatch Logs. Para enviar os mesmos dados de log para locais diferentes, você pode incluir seções adicionais com IDs exclusivos (por exemplo, "CloudWatchLogs2" e CloudWatchLogs3") e uma região diferente para cada ID.

Para definir configurações para enviar dados de log ao CloudWatch Logs

1. No arquivo JSON, localize a seção `CloudWatchLogs`.

   ```
   {
       "Id": "CloudWatchLogs",
       "FullName":
    "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
       "Parameters": {
           "AccessKey": "",
           "SecretKey": "",
           "Region": "us-east-1",
           "LogGroup": "Default-Log-Group",
           "LogStream": "{instance_id}"
       }
   },
   ```

2. Deixe os campos `AccessKey` e `SecretKey` em branco. Configure as credenciais usando uma função do IAM.
3. Em `Region`, digite a região para a qual enviar os dados de log (por exemplo, `us-east-2`).
4. Em `LogGroup`, digite o nome do grupo de logs. Esse nome aparece na tela Log Groups (Grupos de logs) no console do CloudWatch.
5. Em `LogStream`, digite o fluxo de log de destino. Esse nome aparece na tela Log Groups > Streams (Grupos de logs > Fluxos) no console do CloudWatch.

   Se você usar `{instance_id}`, o padrão, o nome do fluxo de logs será o ID dessa instância.

   Se você especificar um nome de fluxo de logs que ainda não existe, o CloudWatch Logs o criará automaticamente. Você pode definir um nome de fluxo de log usando uma sequência literal, as variáveis predefinidas `{hostname}`, `{instance_id}` e `{ip_address}` ou uma combinação delas.

Etapa 3 Configurar os dados para enviar

Você pode enviar dados de log de eventos, dados do rastreamento de eventos para Windows (ETW – Event Tracing for Windows) e outros dados de log ao CloudWatch Logs.

Para enviar dados de log de eventos de aplicativo do Windows para o CloudWatch Logs

1. No arquivo JSON, localize a seção `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. Em `Levels`, especifique o tipo de mensagens das quais fazer upload. Você pode especificar um dos seguintes valores:

   - **1** – fazer upload apenas de mensagens de erro.
   - **2** – fazer upload apenas de mensagens de aviso.
   - **4** – fazer upload apenas de mensagens informativas.

   Você pode combinar valores para incluir mais de um tipo de mensagem. Por exemplo, o valor **3** faz upload de mensagens de erro (**1**) e de mensagens de aviso (**2**). Um valor **7** faz upload de mensagens de erro (**1**), de mensagens de aviso (**2**) e de mensagens informativas (**4**).

Para enviar dados de log de segurança para o CloudWatch Logs

1. No arquivo JSON, localize a seção `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. Em `Levels`, digite **7** para fazer upload de todas as mensagens.

Para enviar dados de log de eventos do sistema para o CloudWatch Logs

1. No arquivo JSON, localize a seção `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. Em `Levels`, especifique o tipo de mensagens das quais fazer upload. Você pode especificar um dos seguintes valores:

   - **1** – fazer upload apenas de mensagens de erro.
   - **2** – fazer upload apenas de mensagens de aviso.

- **4** – fazer upload apenas de mensagens informativas.

Você pode combinar valores para incluir mais de um tipo de mensagem. Por exemplo, o valor **3** faz upload de mensagens de erro (**1**) e de mensagens de aviso (**2**). Um valor **7** faz upload de mensagens de erro (**1**), de mensagens de aviso (**2**) e de mensagens informativas (**4**).

Para enviar outros tipos de dados de log de eventos para o CloudWatch Logs

1. No arquivo JSON, adicione uma nova seção. Cada seção deve ter um `Id` exclusivo.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. Em `Id`, digite o nome do log para fazer upload (por exemplo, **`WindowsBackup`**).
3. Em `LogName`, digite o nome do log do qual fazer upload. Você pode localizar o nome do log da seguinte forma.
   a. Abra o Visualizador de eventos.
   b. No painel de navegação, escolha Applications and Services Logs.
   c. Navegue até o log e escolha Actions, Properties.
4. Em `Levels`, especifique o tipo de mensagens das quais fazer upload. Você pode especificar um dos seguintes valores:

   - **1** – fazer upload apenas de mensagens de erro.
   - **2** – fazer upload apenas de mensagens de aviso.
   - **4** – fazer upload apenas de mensagens informativas.

   Você pode combinar valores para incluir mais de um tipo de mensagem. Por exemplo, o valor **3** faz upload de mensagens de erro (**1**) e de mensagens de aviso (**2**). Um valor **7** faz upload de mensagens de erro (**1**), de mensagens de aviso (**2**) e de mensagens informativas (**4**).

Para enviar os dados de rastreamento de eventos para Windows ao CloudWatch Logs

O ETW (Rastreamento de Eventos para Windows) fornece um mecanismo de registro eficiente e detalhado no qual os aplicativos podem gravar logs. Cada ETW é controlado por um gerente de sessão que pode iniciar e parar a sessão de registro. Cada sessão tem um provedor e um ou mais consumidores.

1. No arquivo JSON, localize a seção `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. Em `LogName`, digite o nome do log do qual fazer upload.
3. Em `Levels`, especifique o tipo de mensagens das quais fazer upload. Você pode especificar um dos seguintes valores:

   - **`1`** – fazer upload apenas de mensagens de erro.
   - **`2`** – fazer upload apenas de mensagens de aviso.
   - **`4`** – fazer upload apenas de mensagens informativas.

   Você pode combinar valores para incluir mais de um tipo de mensagem. Por exemplo, o valor **`3`** faz upload de mensagens de erro (**`1`**) e de mensagens de aviso (**`2`**). Um valor **`7`** faz upload de mensagens de erro (**`1`**), de mensagens de aviso (**`2`**) e de mensagens informativas (**`4`**).

Para enviar logs personalizados (qualquer arquivo de log baseado em texto) ao CloudWatch Logs

1. No arquivo JSON, localize a seção `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. Em `LogDirectoryPath`, digite o caminho onde os logs estão armazenados na instância.
3. Em `TimestampFormat`, digite o formato do time stamp a ser usado. Para obter mais informações sobre os valores compatíveis, consulte o tópico Cadeias de caracteres de formato de data e hora personalizado no MSDN.

   Important

   Seu arquivo de log de origem deve ter o time stamp no início de cada linha do log e deve haver um espaço depois do time stamp.
4. Em `Encoding`, digite a codificação de arquivo a ser usada (por exemplo, UTF-8). Para obter mais informações sobre os valores compatíveis, consulte o tópico Encoding Class no MSDN.

   Note

   Use o nome da codificação, não o nome da exibição.
5. (Opcional) Em `Filter`, digite o prefixo de nomes de logs. Deixe esse parâmetro em branco para monitorar todos os arquivos. Para obter mais informações sobre os valores compatíveis, consulte o tópico Propriedade FileSystemWatcherFilter no MSDN.
6. (Opcional) Em `CultureName`, digite a localidade na qual o time stamp é registrado no log. Se `CultureName` ficar em branco, será adotada como padrão a mesma localidade usada atualmente pela instância do Windows. Para obter mais informações sobre os valores compatíveis, consulte a coluna `Language tag` na tabela no tópico Product Behavior no MSDN.

   Note

   O `div`, `div-MV`, `hu`, e `hu-HU` os valores não são suportados.

7. (Opcional) Para `TimeZoneKind`, tipo `Local` ou `UTC`. Pode definir esta informação para fornecer informações sobre o fuso horário quando não estiver incluído informação de fuso horário no carimbo de hora do seu registo. Se esse parâmetro for deixado em branco e se o time stamp não incluir informações sobre fuso horário, o CloudWatch Logs adotará o fuso horário local como o padrão. Esse parâmetro será ignorado se o time stamp já incluir informações sobre o fuso horário.
8. (Opcional) Em `LineCount`, digite o número de linhas do cabeçalho para identificar o arquivo de log. Por exemplo, os arquivos de log do IIS têm cabeçalhos praticamente idênticos. Você pode especificar **`5`**, que determina a leitura das três primeiras linhas do cabeçalho do arquivo de log para identificá-lo. Nos arquivos de log do IIS, a terceira linha é o carimbo de data/hora, mas nem sempre é garantido que o carimbo será diferente entre os arquivos de log. Por esse motivo, recomendamos incluir pelo menos uma linha de dados de log reais para identificar o arquivo de log de forma exclusiva.

Para enviar dados de log do IIS ao CloudWatch Logs

1. No arquivo JSON, localize a seção `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. Em `LogDirectoryPath`, digite a pasta onde os logs do IIS são armazenados para um site individual (por exemplo, `C:\inetpub\logs\LogFiles\W3SVC`*`n`*).

   Note

   Há suporte apenas para o formato de log W3C. Não há suporte para os formatos IIS, NCSA e Personalizado.
3. Em `TimestampFormat`, digite o formato do time stamp a ser usado. Para obter mais informações sobre os valores compatíveis, consulte o tópico Cadeias de caracteres de formato de data e hora personalizado no MSDN.
4. Em `Encoding`, digite a codificação de arquivo a ser usada (por exemplo, UTF-8). Para obter mais informações sobre os valores compatíveis, consulte o tópico Encoding Class no MSDN.

   Note

   Use o nome da codificação, não o nome da exibição.
5. (Opcional) Em `Filter`, digite o prefixo de nomes de logs. Deixe esse parâmetro em branco para monitorar todos os arquivos. Para obter mais informações sobre os valores compatíveis, consulte o tópico Propriedade FileSystemWatcherFilter no MSDN.
6. (Opcional) Em `CultureName`, digite a localidade na qual o time stamp é registrado no log. Se `CultureName` ficar em branco, será adotada como padrão a mesma localidade usada atualmente pela instância do Windows. Para obter mais informações sobre os valores compatíveis, consulte a coluna `Language tag` na tabela no tópico Product Behavior no MSDN.

   Note

   O `div`, `div-MV`, `hu`, e `hu-HU` os valores não são suportados.

7. (Opcional) Para `TimeZoneKind`, enter `Local` ou `UTC`. Pode definir esta informação para fornecer informações sobre o fuso horário quando não estiver incluído informação de fuso horário no carimbo de hora do seu registo. Se esse parâmetro for deixado em branco e se o time stamp não incluir informações sobre fuso horário, o CloudWatch Logs adotará o fuso horário local como o padrão. Esse parâmetro será ignorado se o time stamp já incluir informações sobre o fuso horário.
8. (Opcional) Em `LineCount`, digite o número de linhas do cabeçalho para identificar o arquivo de log. Por exemplo, os arquivos de log do IIS têm cabeçalhos praticamente idênticos. Você pode especificar **`5`**, que leria as cinco primeiras linhas do cabeçalho do arquivo de log para identificá-lo. Nos arquivos de log do IIS, a terceira linha é o carimbo de data/hora, mas nem sempre é garantido que o carimbo será diferente entre os arquivos de log. Por esse motivo, recomendamos incluir pelo menos uma linha de dados de log reais para identificar o arquivo de log de forma exclusiva.

Etapa 4. Configurar controlo de fluxo

Cada tipo de dados deve ter um destino correspondente na seção `Flows`. Por exemplo, para enviar o log personalizado, o log do ETW e o log do sistema ao CloudWatch Logs, adicione `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs` à seção do `Flows`.

> Warning
>
> A adição de uma etapa inválida bloqueia o fluxo. Por exemplo, se você adicionar uma etapa de métrica de disco, mas a instância não tiver um disco, todas as etapas do fluxo serão bloqueadas.

Você pode enviar o mesmo arquivo de log a mais de um destino. Por exemplo, para enviar o log do aplicativo a dois destinos diferentes que você definiu na seção `CloudWatchLogs`, adicione `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)` à seção `Flows`.

Para configurar o controle de fluxo

1. No arquivo `AWS.EC2.Windows.CloudWatch.json`, localize a seção `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. Em `Flows`, adicione cada tipo de dados os quais serão feito upload (por exemplo, `ApplicationEventLog`) e seu destino (por exemplo, `CloudWatchLogs`).

Agora você concluiu a edição do arquivo JSON. Você o usa em uma etapa posterior.

Iniciar o agente

Para ativar Amazon EC2 instância a executar o Windows Server 2012 ou Windows Server 2008 para enviar registos para CloudWatch Logs, utilize o serviço EC2Config (`EC2Config.exe`). A sua instância deve ter EC2Config 4.0 ou posterior e pode utilizar este procedimento. Para obter mais informações sobre como usar uma versão anterior do EC2Config, consulte Usar o EC2Config 3.x ou anterior para configurar o CloudWatch no Guia do usuário do Amazon EC2 para instâncias do Windows

Para configurar o CloudWatch usando o EC2Config 4.x

1. Verifique a codificação do arquivo AWS.EC2.Windows.CloudWatch.json editado anteriormente neste procedimento. Só há suporte para a codificação UTF-8 sem BOM. Em seguida, salve o arquivo na

pasta a seguir na instância do Windows Server 2008 - 2012 R2: . `C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch\`.

2. Inicie ou reinicie o agente do SSM (`AmazonSSMAgent.exe`) usando o painel de controle de Serviços do Windows ou usando o seguinte comando do PowerShell:

```
PS C:\> Restart-Service AmazonSSMAgent
```

Após a reinicialização do agente do SSM, ele detecta o arquivo de configuração e configura a instância para integração com o CloudWatch. Se você alterar os parâmetros e as configurações no arquivo de configuração local, será necessário reiniciar o agente do SSM para que as alterações sejam efetivadas. Para desabilitar a integração com o `IsEnabled` na instância, altere `false` para CloudWatch e salve as alterações no arquivo de configuração.

Início rápido Instalar o CloudWatch Logs Agente utilizando AWS OpsWorks e Chef

Você pode instalar o agente do CloudWatch Logs e criar fluxos de logs usando o AWS OpsWorks e o Chef, que é uma ferramenta de automação de sistemas e infraestrutura em nuvem de terceiros. O Chef usa "receitas", que você grava para instalar e configurar o software em seu computador, e "livros de receitas", que são coleções de receitas, para executar suas tarefas de configuração e distribuição de políticas. Para obter mais informações, consulte Chef.

Os exemplos de receitas do Chef a seguir mostram como monitorar um arquivo de log em cada instância do EC2. As receitas usam o nome da pilha como o grupo de logs e o nome de host da instância como o nome do stream de logs. Para monitorar vários arquivos de log, você precisa estender as receitas para criar vários grupos e fluxos de logs.

Etapa 1 Criar receitas personalizadas

Crie um repositório para armazenar suas receitas. O AWS OpsWorks oferece suporte ao Git e ao Subversion, ou você pode armazenar um arquivo no Amazon S3. A estrutura do repositório do livro de receitas é descrita em Repositórios de livros de receitas no AWS OpsWorks User Guide. Os exemplos abaixo assumem que o livro de culinária é nomeado `logs`. A receita install.rb instala o CloudWatch Logs agente. Também é possível fazer download do exemplo de livro de receitas (CloudWatchLogs-Cookbooks.zip).

Crie um arquivo chamado metadata.rb que contém o código a seguir:

```
#metadata.rb

name             'logs'
version          '0.0.1'
```

Crie o arquivo de configuração CloudWatch Logs.

```
#config.rb

template "/tmp/cwlogs.cfg" do
  cookbook "logs"
  source "cwlogs.cfg.erb"
  owner "root"
  group "root"
  mode 0644
end
```

Faça download e instale o agente do CloudWatch Logs:

```
# install.rb

directory "/opt/aws/cloudwatch" do
  recursive true
end

remote_file "/opt/aws/cloudwatch/awslogs-agent-setup.py" do
  source "https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py"
  mode "0755"
end

  execute "Install CloudWatch Logs agent" do
  command "/opt/aws/cloudwatch/awslogs-agent-setup.py -n -r region -c /tmp/cwlogs.cfg"
  not_if { system "pgrep -f aws-logs-agent-setup" }
end
```

> **Note**
>
> No exemplo acima, substituir *region* com um dos seguintes: us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, or sa-east-1.

Se a instalação do agente falhar, verifique se o pacote `python-dev` está instalado. Se não estiver, use o comando a seguir e, em seguida, tente outra vez a instalação do agente:

```
sudo apt-get -y install python-dev
```

Essa receita usa um arquivo de modelo cwlogs.cfg.erb que você pode modificar para especificar vários atributos, como quais arquivos registrar. Para obter mais informações sobre esses atributos, consulte Referência do agente do CloudWatch Logs (p. 152).

```
[general]
# Path to the AWSLogs agent's state file. Agent uses this file to maintain
# client side state across its executions.
state_file = /var/awslogs/state/agent-state


## Each log file is defined in its own section. The section name doesn't
## matter as long as its unique within this file.
#
#[kern.log]
#
## Path of log file for the agent to monitor and upload.
#
#file = /var/log/kern.log
#
## Name of the destination log group.
#
#log_group_name = kern.log
#
## Name of the destination log stream.
#
#log_stream_name = {instance_id}
#
## Format specifier for timestamp parsing.
#
#datetime_format = %b %d %H:%M:%S
#
#
```

```
[<%= node[:opsworks][:stack][:name] %>]
datetime_format = [%Y-%m-%d %H:%M:%S]
log_group_name = <%= node[:opsworks][:stack][:name].gsub(' ','_') %>
file = <%= node[:cwlogs][:logfile] %>
log_stream_name = <%= node[:opsworks][:instance][:hostname] %>
```

O modelo obtém o nome da pilha e o nome de host consultando os atributos correspondentes na configuração de pilha e no JSON de implantação. O atributo que especifica o arquivo a ser registrado é definido no arquivo de atributos default.rb do livro de receitas (logs/atributos/default.rb).

```
default[:cwlogs][:logfile] = '/var/log/aws/opsworks/opsworks-agent.statistics.log'
```

Etapa 2. Criar um AWS OpsWorks Pilha

1. Abra o console do AWS OpsWorks em https://console.aws.amazon.com/opsworks/.
2. Em OpsWorks Dashboard (Painel do OpsWorks), escolha Add stack (Adicionar pilha) para criar uma pilha do AWS OpsWorks.
3. Na tela Adicionar pilha, escolha Pilha do Chef 11.
4. Em Nome da pilha, digite um nome.
5. Em Usar livros de receitas personalizadas do Chef, escolha Sim.
6. Em Tipo de repositório, selecione o tipo de repositório que você usa. Se você estiver usando o exemplo acima, escolha Arquivo Http.
7. Em URL de repositório, insira o repositório onde você armazenou o livro de receitas criado na etapa anterior. Se você estiver usando o exemplo acima, insira **`https://s3.amazonaws.com/aws-cloudwatch/downloads/CloudWatchLogs-Cookbooks.zip`**.
8. Selecione Criar para criar uma pilha.

Etapa 3 Prolongar o seu IAM Função

Para usar o CloudWatch Logs com suas instâncias do AWS OpsWorks, você precisa estender a função do IAM usada por suas instâncias.

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Políticas, Criar política.
3. Na página Criar política, em Criar sua própria política, escolha Selecionar. Para obter mais informações sobre a criação de políticas personalizadas, consulte Políticas do IAM para o Amazon EC2 no Guia do usuário do Amazon EC2 para instâncias do Linux.
4. Na página Revisar política, em Nome da política, digite um nome para a política.
5. Em Documento da política, cole a política a seguir:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
```

```
  }
 ]
}
```

6. Selecione Create Policy (Criar política).
7. No painel de navegação, escolha Roles (Funções) e, em seguida, no painel de conteúdo, em Role Name (Nome da função), selecione o nome da função da instância usada por sua pilha do AWS OpsWorks. Você pode encontrar a usada pela sua pilha nas configurações da pilha (o padrão é `aws-opsworks-ec2-role`).

   Note

   Escolha o nome da função, não a caixa de seleção.
8. Na guia Permissões, em Políticas gerenciadas, selecione Anexar política.
9. Na página Anexar política, no cabeçalho da tabela (ao lado de Filtro e Pesquisar), escolha Tipo de política, Políticas gerenciadas pelo cliente.
10. Em Customer Managed Policies (Políticas gerenciadas pelo cliente), selecione a política do IAM que você criou acima e escolha Attach Policy (Anexar política).

    Para obter mais informações sobre políticas e usuários do IAM, consulte Usuários e grupos do IAM e Gerenciamento de políticas do IAM no Guia do usuário do IAM.

Etapa 4. Adicionar uma camada

1. Abra o console do AWS OpsWorks em https://console.aws.amazon.com/opsworks/.
2. No painel de navegação, escolha Layers (Camadas).
3. No painel de conteúdo, selecione uma camada e escolha Adicionar camada.
4. Na guia OpsWorks, em Tipo de camada, escolha Personalizar.
5. Nos campos Nome e Nome curto, digite os nomes longo e curto da camada. Em seguida, escolha Adicionar camada.
6. Na guia Recipes (Receitas), em Custom Chef Recipes (Receitas personalizadas do Chef), há vários títulos – Setup (Instalar), Configure (Configurar), Deploy (Implantar), Undeploy (Cancelar implantação) e Shutdown (Encerrar) – que correspondem aos eventos de ciclo de vida do AWS OpsWorks. O AWS OpsWorks aciona esses eventos nos pontos chave no ciclo de vida da instância que executa as receitas associadas.

   Note

   Se os títulos acima não estiverem visíveis, em Receitas personalizadas do chef, escolha editar.
7. Digite logs::config, logs::install próximo de Configuração, escolha + para adicioná-lo à lista e escolha Salvar.

   O AWS OpsWorks executa essa receita em cada uma das novas instâncias nessa camada, logo depois que a instância é inicializada.

Etapa 5. Adicionar uma instância

A camada só controla como configurar instâncias. Agora, é preciso adicionar algumas instâncias à camada e iniciá-las.

1. Abra o console do AWS OpsWorks em https://console.aws.amazon.com/opsworks/.
2. No painel de navegação, selecione Instâncias e, na sua camada, selecione + Instância.
3. Aceite as configurações padrão e escolha Adicionar instância para adicionar a instância à camada.

4. Na coluna Ações da linha, clique em iniciar para iniciar a instância.

   O AWS OpsWorks inicia uma nova instância do EC2 e configura o CloudWatch Logs. O status da instância mudará para online quando estiver pronta.

Etapa 6. Ver os seus registos

Você deve ver o grupo de logs recém-criado e o fluxo de logs no console do CloudWatch depois de alguns minutos de execução do agente.

Para obter mais informações, consulte Visualizar os dados de log enviados para o CloudWatch Logs (p. 59).

Relatar o status do agente do CloudWatch Logs

Use o procedimento a seguir para relatar o status do agente do CloudWatch Logs em sua instância do EC2.

Para relatar o status do agente

1. Conecte-se à sua instância do EC2. Para obter mais informações, consulte o tópico Conecte-se à sua instância no Guia do usuário do Amazon EC2 para instâncias do Linux.

   Para obter mais informações sobre problemas de conexão, consulte Resolução de problemas para se conectar à sua instância no Guia do usuário do Amazon EC2 para instâncias do Linux.
2. Em um prompt de comando, digite o seguinte comando:

   ```
   sudo service awslogs status
   ```

   Se você está executando o Amazon Linux 2, digite o seguinte comando:

   ```
   sudo service awslogsd status
   ```
3. Confira o arquivo /var/log/awslogs.log para verificar se há erros, avisos ou problemas com o agente do CloudWatch Logs.

Iniciar o agente do CloudWatch Logs

Se o agente do CloudWatch Logs em sua instância do EC2 não foi iniciado automaticamente após a instalação, ou se você interrompeu o agente, você poderá usar o procedimento a seguir para iniciar o agente.

Para iniciar o agente da

1. Conecte-se à sua instância do EC2. Para obter mais informações, consulte o tópico Conecte-se à sua instância no Guia do usuário do Amazon EC2 para instâncias do Linux.

   Para obter mais informações sobre problemas de conexão, consulte Resolução de problemas para se conectar à sua instância no Guia do usuário do Amazon EC2 para instâncias do Linux.
2. Em um prompt de comando, digite o seguinte comando:

   ```
   sudo service awslogs start
   ```

   Se você está executando o Amazon Linux 2, digite o seguinte comando:

```
sudo service awslogsd start
```

Interrompa o agente do CloudWatch Logs

Use o procedimento a seguir para interromper o agente do CloudWatch Logs em sua instância do EC2.

Para interromper o agente da

1. Conecte-se à sua instância do EC2. Para obter mais informações, consulte o tópico Conecte-se à sua instância no Guia do usuário do Amazon EC2 para instâncias do Linux.

   Para obter mais informações sobre problemas de conexão, consulte Resolução de problemas para se conectar à sua instância no Guia do usuário do Amazon EC2 para instâncias do Linux.
2. Em um prompt de comando, digite o seguinte comando:

   ```
   sudo service awslogs stop
   ```

   Se você está executando o Amazon Linux 2, digite o seguinte comando:

   ```
   sudo service awslogsd stop
   ```

Início rápido Utilização AWS CloudFormation para Começar com CloudWatch Logs

O AWS CloudFormation permite descrever e provisionar seus recursos da AWS no formato JSON. As vantagens desse método são a capacidade de gerenciar uma coleção de recursos da AWS como uma unidade e replicar com facilidade seus recursos da AWS em todas as regiões.

Ao provisionar a AWS com o AWS CloudFormation, você cria modelos que descrevem os recursos da AWS a serem usados. O exemplo a seguir é um trecho de modelo que cria um grupo de logs e um filtro de métrica que conta as ocorrências de 404 e envia essa contagem para o grupo de logs.

```
"WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {
        "RetentionInDays": 7
    }
},

"404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
        "LogGroupName": {
            "Ref": "WebServerLogGroup"
        },
        "FilterPattern": "[ip, identity, user_id, timestamp, request, status_code = 404,
 size, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
                "MetricName": "test404Count"
```

```
            }
        ]
    }
}
```

Este é um exemplo simples. Você pode configurar implantações mais avançadas do CloudWatch Logs usando o AWS CloudFormation. Para obter mais informações sobre exemplos de modelos, consulte Amazon CloudWatch LogsSnippets de modelo no Guia do usuário do AWS CloudFormation. Para obter mais informações sobre como começar, consulte Conceitos básicos do AWS CloudFormation no Guia do usuário do AWS CloudFormation.

Analisar dados de log com o CloudWatch Logs Insights

CloudWatch Logs InsightsO permite pesquisar e analisar interativamente os dados de log no Amazon CloudWatch Logs. Realize consultas para ajudar a responder de maneira mais rápida e eficiente a problemas operacionais. Se um problema ocorrer, use o CloudWatch Logs Insights para identificar causas em potencial e validar correções implantadas.

O CloudWatch Logs Insights inclui uma linguagem de consulta específica com alguns comandos simples, mas eficientes. O CloudWatch Logs Insights fornece exemplos de consultas, descrições de comando, preenchimento automático de consulta e descoberta do campo de log para ajudar você a dar os primeiros passos. Exemplos de consultas estão incluídos para diversos tipos de logs de serviço da AWS

O CloudWatch Logs Insights descobre automaticamente campos em logs em serviços da AWS, como Amazon Route 53, AWS Lambda, AWS CloudTrail e Amazon VPC, e qualquer aplicativo ou log personalizado que emita eventos de log como JSON.

É possível usar o CloudWatch Logs Insights para pesquisar dados de log que foram enviados para o CloudWatch Logs em 5 de novembro de 2018 ou posteriormente.

Uma única solicitação pode consultar até 20 grupos de logs. As consultas expiram após 15 minutos, caso não tenham sido concluídas. Os resultados da consulta ficam disponíveis por 7 dias.

Você pode salvar consultas que você criou. Isso pode ajudá-lo a realizar consultas complexas quando precisar, sem ter de recriá-las sempre que quiser executá-las.

> **Important**
>
> Se a equipe de segurança de rede não permitir o uso de soquetes web, não será possível acessar a parte CloudWatch Logs Insights do console CloudWatch Você pode usar os recursos de consulta do CloudWatch Logs Insights usando o APIs. Para obter mais informações, consulte StartQuery no Amazon CloudWatch Logs API Reference.

Tópicos

Logs compatíveis e campos descobertos

CloudWatch Logs InsightsO oferece suporte a todos os tipos de logs. Para cada log enviado ao CloudWatch Logs, cinco campos do sistema são gerados automaticamente:

- `@message` contém o evento de log não avaliado bruto. Isso é equivalente ao campo `message` em InputLogevent.
- `@timestamp` contém o time stamp do evento contido no campo `timestamp` do evento de log. Isso é equivalente ao campo `timestamp` em InputLogevent.
- `@ingestionTime` contém a hora em que o evento de log foi recebido pelo CloudWatch Logs.
- `@logStream` contém o nome do fluxo de logs ao qual o evento de log foi adicionado. Os fluxos de logs são usados para agrupar logs pelo mesmo processo que os gerou.
- `@log` é um identificador de grupo de logs na forma de *`account-id`*`:`*`log-group-name`*. Isso pode ser útil em consultas de vários grupos de logs, para identificar a qual grupo de logs um determinado evento pertence.

O CloudWatch Logs Insights insere o símbolo @ no início dos campos que ele gera.

Para muitos tipos de log, o CloudWatch Logs também detecta automaticamente os campos de log contidos nos logs. Esses campos de descoberta automática são mostrados na tabela a seguir.

Para outros tipos de logs com campos que o CloudWatch Logs Insights não descobre automaticamente, use o comando **`parse`** para extrair e criar campos efêmeros a serem usados nessa consulta. Para obter mais informações, consulte CloudWatch Logs InsightsSintaxe da consulta do (p. 41)

Se o nome de um campo de log descoberto começar com o caractere `@`, o CloudWatch Logs Insights o exibirá com um `@` adicional anexado ao início. Por exemplo, se um nome de campo de log for `@example.com`, o nome desse campo será exibido como `@@example.com`.

Tipo de log	Campos de log descobertos
Amazon VPCLogs de fluxo do	`@timestamp`, `@logStream`, `@message`, `accountId`, `endTime`, `interfaceId`, `logStatus`, `startTime`, `version`, `action`, `bytes`, `dstAddr`, `dstPort`, `packets`, `protocol`, `srcAddr`, `srcPort`
Route 53Logs do	`@timestamp`, `@logStream`, `@message`, `edgeLocation`, `hostZoneId`, `protocol`, `queryName`, `queryTimestamp`, `queryType`, `resolverIp`, `responseCode`, `version`
LambdaLogs do	`@timestamp`, `@logStream`, `@message`, `@requestId`, `@duration`, `@billedDuration`, `@type`, `@maxMemoryUsed`, `@memorySize` Se uma linha de log do Lambda contiver um ID de rastreamento X-Ray, ela também incluirá os seguintes campos: `@xrayTraceId` e `@xraySegmentId`. O CloudWatch Logs Insights descobre automaticamente campos de log em logs do Lambda, mas apenas para o primeiro fragmento JSON incorporado em cada evento de log. Se um evento de log do Lambda contiver vários fragmentos JSON, será possível analisar e extrair os campos de log usando o comando **`parse`** Para obter mais informações, consulte Campos em logs JSON (p. 37).
CloudTrailLogs do Logs em formato JSON	Para obter mais informações, consulte Campos em logs JSON (p. 37).

Tipo de log	Campos de log descobertos
Outros tipos de log	`@timestamp`, `@ingestionTime`, `@logStream`, `@message`, `@log`.

Campos em logs JSON

CloudWatch Logs InsightsO representa campos JSON aninhados que usam a notação de ponto final. No evento JSON de exemplo a seguir, o campo `type` no objeto JSON `userIdentity` é representado como `userIdentity.type`.

As matrizes JSON são mescladas em uma lista de nomes de campo e valores. Por exemplo, para especificar o valor de `instanceId` para o primeiro item em `requestParameters.instancesSet`, use `requestParameters.instancesSet.items.0.instanceId`.

CloudWatch Logs InsightsO pode extrair um máximo de 100 campos de eventos de log de um log JSON. Para campos extras que não são extraídos, você pode usar o comando parse para analisar esses campos no evento de log não analisado bruto no campo de mensagem.

```
{ "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21: 22: 54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "205.251.233.176",
    "userAgent": "ec2-api-tools1.6.12.2",
    "requestParameters": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123"
                }
            ]
        }
    },
    "responseElements": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123",
                    "currentState": {
                        "code": 0,
                        "name": "pending"
                    },
                    "previousState": {
                        "code": 80,
                        "name": "stopped"
                    }
                }
            ]
        }
    }
```

Tutorial: executar e modificar uma consulta de amostra

O tutorial a seguir ajuda você nos conceitos básicos do CloudWatch Logs Insights. Você executa uma consulta de amostra e vê como modificar e reexecutá-la.

Para executar uma consulta, você já deve ter logs armazenados no CloudWatch Logs. Se já estiver usando o CloudWatch Logs e tiver grupos de log e fluxos de logs configurados, você estará pronto para começar. Talvez você já tenha logs caso use serviços, como AWS CloudTrail, Amazon Route 53 ou Amazon VPC, e tenha configurado logs nesses serviços para acessar o CloudWatch Logs. Para obter mais informações sobre como enviar logs ao CloudWatch Logs, consulte Conceitos básicos com CloudWatch Logs (p. 5).

As consultas no CloudWatch Logs Insights retornam um conjunto de campos de eventos de log, o resultado de uma agregação matemática ou outra operação realizada em eventos de log. Este tutorial demonstra uma consulta que retorna uma lista de eventos de log.

Executar uma consulta de amostra

Comece executando uma consulta de amostra.

Para executar uma consulta de amostra do CloudWatch Logs Insights

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.

   O editor de consultas está próximo à parte superior da página. Quando você abre o CloudWatch Logs Insights pela primeira vez, essa caixa contém uma consulta padrão que retorna os 20 eventos de log mais recentes.
3. Selecione um ou mais grupos de logs a serem consultados, acima do editor de consultas. Para ajudar a encontrar os grupos de logs, você pode inserir texto na barra de pesquisa e o CloudWatch Logs exibirá grupos de log correspondentes na barra de pesquisa.

   Quando você seleciona um grupo de logs, o CloudWatch Logs Insights automaticamente detecta os campos nos dados no grupo de logs. Para ver esses campos descobertos, selecione Campos à direita da página.
4. (Opcional) Use o seletor de tempo no canto superior direito para selecionar o período que você deseja consultar.
5. Escolha Run (Executar).

   Os resultados da consulta são exibidos. Neste exemplo, os resultados são 20 eventos de log mais recentes de qualquer tipo.

   CloudWatch LogsO também exibe um gráfico de barras de eventos de log neste grupo de logs com o passar do tempo. Esse gráfico de barras mostra a distribuição de eventos no grupo de logs correspondente à consulta e ao intervalo de tempo, e não apenas os eventos exibidos na tabela.
6. Para ver todos os campos de um dos eventos de log retornados, escolha o ícone à esquerda desse evento de log.

Modificar a consulta de amostra

Neste tutorial, você modifica a consulta de amostra para mostrar os 50 eventos de log mais recentes.

Se você ainda não tiver executado o tutorial anterior, faça isso agora. Este tutorial começa onde esse tutorial anterior termina.

> Note
>
> Algumas consultas de exemplo fornecidas com o CloudWatch Logs Insights usam os comandos `head` ou `tail` em vez de `limit`. Esses comandos estão defasados e foram substituídos por `limit`. Use `limit` em vez de `head` ou `tail` em todas as consultas que você gravar.

Para modificar a consulta de amostra do CloudWatch Logs Insights

1. No editor de consultas, altere 20 para 50, e escolha Executar.

   Os resultados da nova consulta são exibidos. Pressupondo-se que haja dados suficientes no grupo de logs no intervalo de tempo padrão, agora há 50 eventos de log listados.
2. (Opcional) Você pode salvar consultas que você criou. Para salvar essa consulta, escolha Salvar. Para obter mais informações, consulte Salvar e executar novamente as consultas do CloudWatch Logs Insights (p. 53).

Adicionar um comando de filtro à consulta de amostra

Este tutorial mostra como fazer uma alteração mais eficiente na consulta no editor. Neste tutorial, filtre os resultados da consulta anterior com base em um campo nos eventos de log recuperados.

Se você ainda não tiver executado os tutoriais anteriores, faça isso agora. Este tutorial começa onde esse tutorial anterior termina.

Para adicionar um comando de filtro à consulta anterior

1. Decida um campo a ser filtrado. Para ver os campos mais comuns que o CloudWatch Logs detectou nos eventos de log contidos nos grupos de log selecionados nos últimos 15 minutos e a porcentagem desses eventos de log em que cada campo aparece, selecione Campos no lado direito da página.

   Para ver os campos contidos em um determinado evento de log, escolha o ícone à esquerda dessa linha.

   O campo awsRegion pode ser exibido no evento de log, dependendo de quais eventos estão nos logs. No restante deste tutorial, usaremos awsRegion como o campo de filtro, mas você poderá usar um campo diferente se esse campo não estiver disponível.
2. Na caixa do editor de consultas, coloque o cursor após 50 e pressione Enter.
3. Na nova linha, digite primeiro | (o caractere de barra vertical) e um espaço. Os comandos em uma consulta do CloudWatch Logs Insights devem ser separados pelo caractere de barra vertical.
4. Insira **`filter awsRegion="us-east-1"`**.
5. Escolha Run (Executar).

   A consulta é reexecutada e agora exibe os 50 resultados mais recentes correspondentes ao novo filtro.

   Se tiver filtrado em um campo diferente e obtido um resultado de erro, você poderá precisar inserir caracteres de escape no nome do campo. Se o nome do campo incluir caracteres não alfanuméricos, você deverá inserir caracteres de apóstrofo (`) antes e depois do nome do campo (por exemplo, **`` `error-code`="102" ``**).

   Você deve usar os caracteres de apóstrofo para nomes de campo que contenham caracteres não alfanuméricos, mas não para valores. Os valores estão sempre entre aspas (").

CloudWatch Logs InsightsO inclui recursos de consulta eficientes, inclusive vários comandos e suporte para expressões regulares, além de operações matemáticas e estatísticas. Para obter mais informações, consulte CloudWatch Logs InsightsSintaxe da consulta do (p. 41).

Tutorial: executar uma consulta com uma função de agregação

Neste tutorial, execute uma consulta que retorna os resultados de executar funções de agregação em registros de log.

Para executar uma consulta de agregação

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.
3. Selecione um ou mais grupos de logs acima do editor de consultas. Para ajudar a encontrar seus grupos de logs, insira texto na barra de pesquisa e o CloudWatch Logs exibirá grupos de logs correspondentes na barra de pesquisa.
4. No editor de consultas, exclua a consulta mostrada no momento, insira a seguinte e escolha Executar. Substitua fieldname pelo nome de um campo exibido na área Campos no lado direito da página.

```
stats count(*) by fieldname
```

   Os resultados mostram o número de eventos de log no grupo de logs que foram recebidos pelo CloudWatch Logs que contêm cada valor diferente do nome do campo escolhido.

Tutorial: Executar uma consulta que produz uma visualização agrupada por campos de log

Ao executar uma consulta que usa a função `stats` para agrupar os resultados retornados pelos valores de um ou mais campos nas entradas de log, você pode visualizar os resultados como um gráfico de barras, gráfico de pizza, gráfico de linhas ou gráfico de áreas empilhadas. Isso ajuda a visualizar as tendências em seus logs de forma mais eficiente.

Para executar uma consulta de visualização

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.
3. Selecione um ou mais grupos de logs a serem consultados.
4. No editor de consultas, exclua o conteúdo atual, insira a função `stats` a seguir e escolha Run query (Executar consulta).

```
stats count(*) by @logStream
    | limit 100
```

   Os resultados mostram o número de eventos de log no grupo de logs para cada fluxo de log. Os resultados são limitados a somente 100 linhas.
5. Escolha a guia Visualization (Visualização).
6. Selecione a seta ao lado de Linha, e escolha Barra.

O gráfico de barras será exibido, mostrando uma barra para cada fluxo de log no grupo de logs.

Tutorial: Executar uma consulta que produz uma visualização de séries temporais

Ao executar uma consulta que usa a função `bin()` para agrupar os resultados retornados por um período, é possível visualizar os resultados como um gráfico de linhas, gráfico de áreas empilhadas, gráfico de pizza ou gráfico de barras. Isso ajuda a visualizar as tendências em eventos de log com mais eficiência ao longo do tempo.

Para executar uma consulta de visualização

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.
3. Selecione um ou mais grupos de logs a serem consultados.
4. No editor de consultas, exclua o conteúdo atual, insira a função `stats` a seguir e escolha Run query (Executar consulta).

   ```
   stats count(*) by bin(30s)
   ```

   Os resultados mostram o número de eventos de log no grupo de logs que foram recebidos pelo CloudWatch Logs para cada período de 30 segundos.
5. Escolha a guia Visualization (Visualização).

   Os resultados são mostrados como um gráfico de linhas. Para alternar para um gráfico de barras, gráfico de pizza ou gráfico de áreas empilhadas, escolha a seta ao lado de Line (Linha) no canto superior esquerdo do gráfico.

CloudWatch Logs InsightsSintaxe da consulta do

CloudWatch Logs InsightsO oferece suporte a uma linguagem de consulta que você é possível realizar consultas nos grupos de logs. Cada consulta pode incluir um ou mais comandos de consulta separados por caracteres de barra vertical em estilo Unix (`|`).

Há suporte para seis comandos de consulta, além de muitas funções e operações de suporte, inclusive expressões regulares, operações aritméticas, operações de comparação, funções numéricas, funções de datetime, funções de string e funções genéricas.

Também há suporte para os comentários. As linhas de consulta que começam com o caractere `#` são ignoradas.

Campos que começam com o símbolo `@` são gerados pelo CloudWatch Logs Insights. Para obter mais informações sobre os campos que o CloudWatch Logs descobre automaticamente e gera, consulte Logs compatíveis e campos descobertos (p. 36).

CloudWatch Logs InsightsComandos de consulta do

A tabela a seguir lista seis comandos de consulta compatíveis com exemplos básicos. Para obter consultas de exemplo mais eficientes, consulte Consultas de exemplo (p. 54).

Comando	Descrição	Exemplos
`display`	Especifica quais campos serão exibidos nos resultados da consulta. Se você especificar esse comando mais de uma vez em sua consulta, somente os campos especificados na última ocorrência serão usados.	O exemplo a seguir usa o campo `@message` e cria os campos efêmeros `loggingType` e `loggingMessage` para uso na consulta. Ela filtra os eventos apenas para aqueles que têm `ERROR` como o valor de `loggingType`, mas exibe apenas o campo `loggingMessage` desses eventos nos resultados. `fields @message	parse @message "[*] *" as loggingType, loggingMessage	filter loggingType = "ERROR"	display loggingMessage`

`fields`	Recupera os campos especificados de eventos de log para exibição.
	
	Você pode usar funções e operações em um comando fields para modificar valores de campo para exibição e criar novos campos para uso no restante da consulta.
		O exemplo a seguir exibe os campos `foo-bar`, `action` e o valor absoluto da diferença entre `f3` e `f4` para todos os eventos de log no grupo de logs.
		`fields `foo-bar`, action, abs(f3-f4)`
		
		O exemplo a seguir cria e exibe um campo efêmero `opStatus`. O valor de `opStatus` para cada entrada de log é a concatenação dos valores dos campos `Operation` e `StatusCode`, com um hífen entre esses valores.
		`fields concat(Operation, '-', StatusCode) as opStatus`

`filter`	Filtra os resultados de uma consulta com base em uma ou mais condições. É possível usar uma variedade de operadores e expressões no comando **`filter`**. Para obter mais informações, consulte the section called “Correspondências e expressões regulares no comando de filtro” (p. 45).	O exemplo a seguir recupera os campos `f1`, `f2` e `f3` para todos os eventos de log com um valor acima de 2000 no campo `duration`. `fields f1, f2, f3	filter (duration>2000)`  O exemplo a seguir também é uma consulta válida, mas os resultados não exibem campos separados. Em vez disso, os resultados exibem o `@timestamp` e todos os dados de log no campo `@message` para todos os eventos de log em que a duração for maior que 2000. `filter (duration>2000)`

Comando	Descrição	Exemplos

		O exemplo a seguir recupera os campos `f1` e `f2` para todos os eventos de log em que `f1` for 10 ou `f3` for maior que 25.
		`fields f1, f2 | filter (f1=10 or f3>25)`
		O próximo exemplo retorna os eventos de log em que o campo `statusCode` tem um valor entre 200 e 299.
		`fields f1 | filter statusCode like /2\d\d/`
		O próximo exemplo retorna os eventos de log que têm um `statusCode` de "300", "400" ou "500".
		`fields @timestamp, @message | filter statusCode in [300,400,500]`
		Este último exemplo retorna eventos de log que não têm campos `Type` com valores de "foo", "bar" ou "1".
		`fields @timestamp, @message | filter Type not in ["foo","bar",1]`

`stats`	Calcula estatísticas de agregação com base nos valores dos campos de log. Ao usar `stats`, também é possível usar `by` para especificar um ou mais critérios a serem usados para agrupar dados ao calcular as estatísticas. Diversos operadores estatísticos são compatíveis, incluindo `sum()`, `avg()`, `count()`, `min()` e `max()`.	O exemplo a seguir calcula o valor médio de `f1` para cada valor exclusivo de `f2`. `stats avg (f1) by f2`
`sort`	Classifica os eventos de log recuperados. As ordens crescente (`asc`) e decrescente (`desc`) são compatíveis.	O exemplo a seguir classifica os eventos retornados em ordem decrescente com base no valor de `f1` e exibe os campos `f1`, `f2` e `f3`. `fields f1, f2, f3 | sort f1 desc`

Comando	Descrição	Exemplos
limit	Especifica o número de eventos de log retornados pela consulta. É possível usar isso para limitar os resultados a um pequeno número para ver um pequeno conjunto de resultados relevantes. Também é possível usar `limit` com um número entre 1.000 e 10.000 para aumentar o número de linhas de resultado de consulta exibidas no console para um valor maior que o padrão de 1.000 linhas. Se você não especificar um limite, o padrão da consulta exibirá um máximo de 1.000 linhas.	O exemplo a seguir classifica os eventos em ordem decrescente com base no valor de `@timestamp` e exibe os campos `f1` e `f2` para os primeiros 25 eventos por ordem de classificação. Nesse caso, a ordem de classificação é por time stamp a partir do mais recente, de maneira que os 25 eventos mais recentes sejam retornados. `sort @timestamp desc | limit 25 | display f1, f2`

parse	Extrai dados de um campo de log e cria um ou mais campos efêmeros que você pode processar mais na consulta. `parse` aceita expressões glob e expressões regulares.
	
	Para analisar as expressões glob, forneça o comando `parse` com uma string constante (caracteres entre aspas únicas ou duplas) em que cada variável de texto é substituída por um asterisco (*). Isso é extraído em campos efêmeros e fornecidos como um alias após a palavra-chave `as`, na ordem posicional.
	
	Coloque expressões regulares em barras (/). Na expressão, cada parte da string correspondente que deve ser extraída é incluída em um grupo de captura nomeado. Um exemplo de um grupo de captura nomeado é `(?<name>.*)`, onde `name` é o nome e `.*` é o padrão.
		Usando esta única linha de log como um exemplo:
		
		`25 May 2019 10:24:39,474 [ERROR] {foo=2, bar=data} The error was: DataIntegrityException`
		
		As duas expressões `parse` a seguir fazem o seguinte: os campos efêmeros `level`, `config` e `exception` são criados. `level` tem um valor de `ERROR`, `config` tem um valor de `{foo=2, bar=data}` e `exception` tem um valor de `DataIntegrityException`. O primeiro exemplo usa uma expressão glob e o segundo usa uma expressão regular.
		
		`parse @message "[*] * The error was: *" as level, config, exception`
		
		`parse @message /\[(?<level>\S+)\]\s+(?<config>\{.*\})\s+The error was: (?<exception>\S+)/`
		
		O exemplo a seguir usa uma expressão regular para extrair os campos efêmeros `user2`, `method2` e `latency2` do campo de log `@message` e retorna a latência média para cada combinação exclusiva de `method2` e `user2`.
		
		`parse @message /user=(?<user2>.*?), method:(?<method2>.*?), latency := (?<latency2>.*?)/ | stats avg(latency2) by method2, user2`

Observações sobre comandos de consulta na tabela anterior

As seguintes regras, diretrizes e dicas se aplicam aos comandos de consulta na tabela anterior.

- Qualquer campo de log nomeado em uma consulta que tenha caracteres diferentes do sinal @, do ponto final (.) e de caracteres alfanuméricos deve estar entre caracteres de apóstrofo (` Por exemplo, o nome do campo `foo-bar` deve ser incluído em caracteres de apóstrofo porque inclui um caractere não alfanumérico.
- Ambos **`fields`** e **`display`** são usados para especificar os campos a serem exibidos nos resultados da consulta. As diferenças entre os dois são as seguintes:
  - Use o comando **`display`** apenas para especificar quais campos serão exibidos nos resultados. Você pode usar o comando **`fields`** com a palavra-chave as para criar novos campos efêmeros usando funções e os campos que estão no evento de log. Por exemplo, o `fields ispresent(resolverArn) as isRes` cria um campo efêmero chamado `isRes` que pode ser usado no restante da consulta. O valor de `isRes` é 0 ou 1, dependendo de `resolverArn` ser ou não um campo descoberto no evento de log.
  - Se você tiver vários comandos da **`fields`** e não incluir um comando **`display`**, os campos especificados em todos os **`fields`** `commands are displayed.`.
  - Se você tiver vários comandos da **`display`**, apenas os campos especificados no **`display`** `command are displayed.` final

Correspondências e expressões regulares no comando de filtro

Você pode usar operadores de comparação (**=**, **!=**, **<**, **<=**, **>**, **>=**), operadores boolianos (`and`, `or` e `not`) e expressões regulares no comando **`filter`**.

É possível usar `in` para testar a associação de conjunto. Coloque uma matriz com os elementos a serem verificados imediatamente após `in`. Você pode usar o `not` com o `in`. As correspondências de string usando o `in` devem ser correspondências de string completas.

Para filtrar por substrings, você pode usar `like` ou `=~` (sinal de igual seguido de um til) no comando `filter`. Para uma substring correspondente usando `like` ou `=~`, coloque sua substring entre aspas duplas ou simples. Para executar a correspondência da expressão regular, coloque a expressão para corresponder com barras. A consulta só retorna eventos de log correspondentes aos critérios definidos.

Exemplos

Os três exemplos a seguir retornam todos os eventos em que `f1` contém a palavra `Exception`. Os dois primeiros exemplos usam expressões regulares. O terceiro exemplo usa uma correspondência de substring. Os três exemplos fazem distinção entre maiúsculas e minúsculas.

```
fields f1, f2, f3 | filter f1 like /Exception/
```

```
fields f1, f2, f3 | filter f1 =~ /Exception/
```

```
fields f1, f2, f3 | filter f1 like "Exception"
```

O exemplo a seguir altera a pesquisa por "Exceção" para não diferenciar maiúsculas de minúsculas.

```
fields f1, f2, f3 | filter f1 like /(?i)Exception/
```

O exemplo a seguir usa uma expressão regular. Ela retorna todos os eventos em que `f1` é exatamente a palavra `Exception`. A consulta não diferencia maiúsculas de minúsculas.

```
fields f1, f2, f3 | filter f1 =~ /^(?i)Exception$/
```

Usar aliases em consultas

Use `as` para criar um ou mais aliases em uma consulta. Os aliases são compatíveis com os comandos `fields`, `stats` e `sort`.

Crie aliases para campos de log e para os resultados de operações e funções.

Exemplos

Os exemplos a seguir mostram o uso de aliases em comandos de consulta.

```
fields abs(myField) as AbsoluteValuemyField, myField2
```

Retorna o valor absoluto de `myField` como `AbsoluteValuemyField` e também retorna o campo `myField2`.

```
stats avg(f1) as myAvgF1 | sort myAvgF1 desc
```

Calcula a média dos valores do `f1` como `myAvgF1` e os retorna em ordem decrescente com relação a esse valor.

Usando Comentários em Consultas

Você pode comentar as linhas em uma consulta usando o caractere # As linhas que começam com o caractere # são ignoradas. Isso pode ser útil para documentar sua consulta ou ignorar temporariamente parte de uma consulta complexa para uma chamada, sem excluir essa linha.

No exemplo a seguir, a segunda linha da consulta foi ignorada.

```
fields @timestamp, @message
    # | filter @message like /delay/
    | limit 20
```

Operações e funções compatíveis

A linguagem de consulta oferece suporte a muitos tipos de operações e funções, conforme mostrado nas tabelas a seguir.

Operações de comparação

Você pode usar operações de comparação no comando `filter` e como argumentos de outras funções. As operações de comparação aceitam todos os tipos de dados como argumentos e retornam um resultado booliano.

```
= != < <= > >=
```

Operadores booleanos

Você pode usar os operadores boolianos **`and`**, **`or`** e **`not`**. Só é possível usar esses operadores boolianos em funções que retornam um valor booliano.

Operações aritméticas

Você pode usar operações aritméticas nos comandos `filter` e `fields` e como argumentos de outras funções. As operações aritméticas aceitam tipos de dados numéricos como argumentos e retornam resultados numéricos.

Operação	Descrição
`a + b`	Adição
`a - b`	Subtração
`a * b`	Multiplicação
`a / b`	Divisão
`a ^ b`	Exponenciação. `2 ^ 3` retorna `8`
`a % b`	Resto ou módulo. `10 % 3` retorna `1`

Operações numéricas

Você pode usar operações numéricas nos comandos `filter` e `fields` e como argumentos de outras funções. As operações numéricas aceitam tipos de dados numéricos como argumentos e retornam resultados numéricos.

Operação	Tipo de resultado	Descrição
`abs(a: number)`	number	Valor absoluto.
`ceil(a: number)`	number	Arredonde para o máximo (o menor inteiro maior que o valor de `a`).
`floor(a: number)`	number	Arredonde para o mínimo (o maior inteiro menor que o valor de `a`).
`greatest(a: number, ...numbers: number[])`	number	Retorna o maior valor.
`least(a: number, ...numbers: number[])`	number	Retorna o menor valor.
`log(a: number)`	number	Log natural.
`sqrt(a: number)`	number	Raiz quadrada.

Funções gerais

Você pode usar funções gerais nos comandos `filter` e `fields` e como argumentos de outras funções.

Função	Tipo de resultado	Descrição
`ispresent(fieldName: LogField)`	boolean	Retornará `true` se o campo existir.

Função	Tipo de resultado	Descrição
`coalesce(fieldName: LogField, ...fieldNames: LogField[])`	LogField	Retorna o primeiro valor não nulo da lista.

Funções de string

Você pode usar funções de string nos comandos `filter` e `fields` e como argumentos de outras funções.

Função	Tipo de resultado	Descrição
`isempty(fieldName: string)`	boolean	Retornará `true` se o campo não for encontrado ou for uma string vazia.
`isblank(fieldName: string)`	boolean	Retornará `true` se o campo não for encontrado, for uma string vazia ou só contiver espaço branco.
`concat(str: string, ...strings: string[])`	string	Concatena as strings.

`ltrim(str: string)`
 	
`ltrim(str: string, subStr: string)`
	string	Remove o espaço em branco do lado esquerdo da string. Se a função tiver um segundo argumento de string, ela removerá os caracteres de `subStr` da esquerda do `str`. Por exemplo, `ltrim("xyZfooxyZ","xyZ")` retorna `"fooxyZ"`.

`rtrim(str: string)`
 	
`rtrim(str: string, subStr: string)`
	string	Remove o espaço em branco do lado direito da string. Se a função tiver um segundo argumento de string, ela removerá os caracteres de `subStr` da direita do `str`. Por exemplo, `rtrim("xyZfooxyZ","xyZ")` retorna `"xyZfoo"`.

`trim(str: string)`
 	
`trim(str: string, subStr: string)`
	string	Remove o espaço em branco de ambos os lados da string. Se a função tiver um segundo argumento de string, ela removerá os caracteres de `subStr` de ambos os lados do `str`. Por exemplo, `trim("xyZfooxyZ","xyZ")` retorna `"foo"`.

`strlen(str: string)`	number	Retorna o tamanho da string em pontos de código Unicode.

Função	Tipo de resultado	Descrição
`toupper(str: string)`	string	Converte a string em letras maiúsculas.
`tolower(str: string)`	string	Converte a string em letras minúsculas.

`substr(str: string, startIndex: number)`
 	
`substr(str: string, startIndex: number, length: number)`
	string	Retorna uma substring do índice especificado pelo argumento de número ao final da string. Se tiver um segundo argumento de número, a função conterá o tamanho da substring a ser recuperada. Por exemplo, `substr("xyZfooxyZ",3, 3)` retorna `"foo"`.

`replace(str: string, searchValue: string, replaceValue: string)`	string	Substitui todas as instâncias de `searchValue` em `str` por `replaceValue`. Por exemplo, `replace("foo","o","0")` retorna `"f00"`.
`strcontains(str: string, searchValue: string)`	number	Retornará 1 se `str` contiver `searchValue` e 0, do contrário.

Funções de data e hora

Você pode usar funções de data e hora nos comandos `filter` e `fields` e como argumentos de outras funções. Use essas funções para criar buckets de tempo para consultas com funções de agregação.

Como parte de funções de datetime, use períodos que consistam em um número e, em seguida, `m` para minutos ou `h` para horas. Por exemplo, `10m` é 10 minutos e `1h` é uma hora.

Função	Tipo de resultado	Descrição
`bin(period: Period)`	Time stamp	Arredonda o valor de `@timestamp` para o período indicado e trunca.
`datefloor(timestamp: Timestamp, period: Period)`	Time stamp	Trunca o time stamp para o período indicado. Por exemplo, `datefloor(@timestamp, 1h)` trunca todos os valores de `@timestamp` no final.
`dateceil(timestamp: Timestamp, period: Period)`	Time stamp	Arredonda o time stamp para o período indicado e trunca. Por exemplo, `dateceil(@timestamp, 1h)` trunca todos os valores de `@timestamp` no início.
`fromMillis(fieldName: number)`	Time stamp	Interpreta o campo de entrada como o número de milissegundos desde a epoch do Unix e o converte em um time stamp.
`toMillis(fieldName: Timestamp)`	number	Converte o time stamp encontrado no campo nomeado em um número que representa os milissegundos desde a epoch do Unix.

Funções do endereço IP

Você pode usar funções de string no endereço IP dos comandos `filter` e `fields` e como argumentos de outras funções.

Função	Tipo de resultado	Descrição
`isValidIp(fieldName: string)`	boolean	Retorna `true` se o campo for um endereço válido IPv4 ou IPv6.
`isValidIpV4(fieldName: string)`	boolean	Retorna `true` se o campo for um endereço válido do IPv4.
`isValidIpV6(fieldName: string)`	boolean	Retorna `true` se o campo for um endereço válido do IPv6.
`isIpInSubnet(fieldName: string, subnet: string)`	boolean	Retorna `true` se o campo for um endereço IPv4 ou IPv6 válido dentro da sub-rede v4 ou v6 especificada. Ao especificar a sub-rede, use a notação CIDR, como `192.0.2.0/24` ou `2001:db8::/32`.
`isIpv4InSubnet(fieldName: string, subnet: string)`	boolean	Retorna `true` se o campo for um endereço IPv4 válido dentro da sub-rede v4 especificada. Ao especificar a sub-rede, use a notação CIDR, como `192.0.2.0/24`.
`isIpv6InSubnet(fieldName: string, subnet: string)`	boolean	Retorna `true` se o campo for um endereço IPv6 válido dentro da sub-rede v6 especificada. Ao especificar a sub-rede, use a notação CIDR, como `2001:db8::/32`.

Funções de agregação de estatísticas

Você pode usar funções de agregação no comando `stats` e como argumentos de outras funções.

Função	Tipo de resultado	Descrição
`avg(fieldName: NumericLogField)`	number	A média dos valores no campo especificado.

`count()`
 	
`count(fieldName: LogField)`
	number	Conta os eventos de log. `count()` (ou `count(*)`) conta todos os eventos retornados pela consulta, enquanto `count(fieldName)` conta todos os registros que incluem o nome do campo especificado.

`count_distinct(fieldName: LogField)`	number	Retorna o número de valores exclusivos do campo. Se o campo tiver cardinalidade muito alta (muitos valores exclusivos), o valor retornado por `count_distinct` será apenas uma aproximação.
`max(fieldName: LogField)`	LogFieldValue	O máximo dos valores desse campo de log nos logs consultados.

Função	Tipo de resultado	Descrição
`min(fieldName: LogField)`	LogFieldValue	O mínimo dos valores desse campo de log nos logs consultados.
`pct(fieldName: LogFieldValue, percent: number)`	LogFieldValue	Um percentil indica a posição relativa de um valor no conjunto de dados. Por exemplo, `pct(@duration, 95)` retorna o valor `@duration` em que 95% dos valores de `@duration` são menores que esse valor e 5% são maiores que esse valor.
`stddev(fieldName: NumericLogField)`	number	O desvio padrão dos valores no campo especificado.
`sum(fieldName: NumericLogField)`	number	A soma dos valores no campo especificado.

Funções não agregadas de estatísticas

Você pode usar funções de não agregação no comando `stats` e como argumentos de outras funções.

Função	Tipo de resultado	Descrição
`earliest(fieldName: LogField)`	LogField	Retorna o valor de `fieldName` do evento do log que tem o primeiro time stamp nos logs consultados.
`latest(fieldName: LogField)`	LogField	Retorna o valor de `fieldName` do evento do log que tem o último time stamp nos logs consultados.
`sortsFirst(fieldName: LogField)`	LogField	Retorna o valor do `fieldName` que aparece em primeiro lugar nos logs consultados.
`sortsLast(fieldName: LogField)`	LogField	Retorna o valor do `fieldName` que aparece em último lugar nos logs consultados.

Visualizar dados de log em gráficos

É possível usar visualizações, como gráficos de barras, gráficos de linha e gráficos de áreas empilhadas para identificar padrões em seus dados de log de forma mais eficiente. O CloudWatch Logs Insights gera visualizações para consultas que usam a função `stats` e uma ou mais funções de agregação. Para obter mais informações, consulte Aggregation Functions in the Stats Command (p. 50).

Todas essas consultas podem produzir gráficos de barras. Se sua consulta usar a função `bin()` para agrupar os dados por um campo ao longo do tempo, você também poderá ver gráficos de linhas e gráficos de áreas empilhadas.

Tópicos

Visualizar dados de séries temporais

As visualizações de séries temporais funcionam para consultas com as seguintes características:

- A consulta contém uma ou mais funções de agregação. Para obter mais informações, consulte Aggregation Functions in the Stats Command (p. 50).
- A consulta usa a função `bin()` para agrupar os dados por um campo.

Essas consultas podem produzir gráficos de linha, gráficos de área empilhada, gráficos de barras e gráficos de pizza.

Exemplos

Para obter um tutorial completo, consulte the section called “Tutorial: Executar uma consulta que produz uma visualização de séries temporais” (p. 41).

Aqui estão mais exemplos de consultas que funcionam para visualização de séries temporais.

A consulta a seguir gera uma visualização dos valores médios do campo `myfield1`, com um ponto de dados criado a cada cinco minutos. Cada ponto de dados é a agregação das médias dos valores `myfield1` dos logs dos últimos cinco minutos.

```
stats avg(myfield1) by bin(5m)
```

A consulta a seguir gera uma visualização dos três valores com base em campos diferentes, com um ponto de dados criado a cada cinco minutos. A visualização é gerada porque a consulta contém funções de agregação e usa `bin()` como o campo de agrupamento.

```
stats avg(myfield1), min(myfield2), max(myfield3) by bin(5m)
```

Restrições do gráfico de linhas e do gráfico de áreas empilhadas

Consultas que agregam informações de entrada de log, mas que não usam a função `bin()`, podem gerar gráficos de barras. No entanto, as consultas não podem gerar gráficos de linha ou gráficos de áreas empilhadas. Para obter mais informações sobre esses tipos de políticas, consulte the section called “Visualizar dados de log agrupados por campos” (p. 52).

Visualizar dados de log agrupados por campos

É possível produzir gráficos de barras para consultas que usam a função `stats` e uma ou mais funções de agregação. Para obter mais informações, consulte Aggregation Functions in the Stats Command (p. 50).

Para ver a visualização, execute sua consulta. Depois, escolha a guia Visualization (Visualização), selecione a seta ao lado de Line (Linha) e escolha Bar (Barra). As visualizações estão limitadas a até 100 barras no gráfico de barras.

Exemplos

Para obter um tutorial completo, consulte the section called “Tutorial: Executar uma consulta que produz uma visualização agrupada por campos de log” (p. 40). Os parágrafos a seguir incluem mais consultas de exemplo para a visualização por campos.

A consulta de log de fluxo da VPC a seguir localiza o número médio de bytes transferidos por sessão para cada endereço de destino.

```
stats avg(bytes) by dstAddr
```

Também é possível produzir um gráfico que inclua mais de uma barra para cada valor resultante. Por exemplo, a consulta de log de fluxo da VPC a seguir localiza o número médio e máximo de bytes transferidos por sessão para cada endereço de destino.

```
stats avg(bytes), max(bytes) by dstAddr
```

A consulta a seguir localiza o número de logs de consulta do Amazon Route 53 para cada tipo de consulta.

```
stats count(*) by queryType
```

Salvar e executar novamente as consultas do CloudWatch Logs Insights

Depois de criar uma consulta, você pode salvá-la para que possa ser executada novamente mais tarde. Suas consultas salvas são mantidas em uma estrutura de pastas para ajudá-lo a mantê-las organizadas. Você pode salvar até 1.000 consultas do CloudWatch Logs Insights, por região e por conta.

Para salvar uma consulta, você deve estar conectado a uma função que tenha a permissão `logs:PutQueryDefinition` Para ver uma lista de consultas salvas, você deve estar conectado a uma função que tenha a permissão `logs:DescribeQueryDefinitions`

Como salvar uma consulta

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.
3. No editor de consultas, crie uma consulta.
4. Escolha Salvar.

   Se você não vir um botão Save (Salvar), será necessário alterar para o novo design do console do CloudWatch Logs. Para fazer isso:

   a. No painel de navegação, escolha Log groups (Grupos de logs).
   b. Escolha Testar o novo design.
   c. No painel de navegação, escolha Insights e volte para a etapa 3 deste procedimento.
5. Insira um nome para a consulta.
6. (Opcional) Escolha uma pasta na qual deseja salvar a consulta. Selecione Create new (Criar novo) para criar uma pasta. Se você criar uma pasta, poderá usar caracteres de barra (/) no nome da pasta para definir uma estrutura de pasta. Por exemplo, dar o nome **`folder-level-1/folder-level-2`** a uma nova pasta cria uma pasta de nível superior chamada **`folder-level-1`**, com outra pasta chamada **`folder-level-2`** dentro dela. A consulta é salva em **`folder-level-2`**.
7. (Opcional) Altere os grupos de log da consulta ou o texto da consulta.
8. Escolha Salvar.

Como executar uma consulta salva

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.
3. À direita, escolha Consultas.
4. Selecione sua consulta na lista de Saved queries (Consultas salvas). Ela aparece no editor de consulta.

5. Escolha Run (Executar).

Como salvar uma nova versão de uma consulta salva

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.
3. À direita, escolha Consultas.
4. Selecione sua consulta na lista de Saved queries (Consultas salvas). Ela aparece no editor de consulta.
5. Modifique a consulta. Se você precisar executá-la para verificar seu trabalho, escolha Executar consulta.
6. Quando estiver pronto para salvar a nova versão, escolha Ações, Salvar como.
7. Insira um nome para a consulta.
8. (Opcional) Escolha uma pasta na qual deseja salvar a consulta. Selecione Create new (Criar novo) para criar uma pasta. Se você criar uma pasta, poderá usar caracteres de barra (/) no nome da pasta para definir uma estrutura de pasta. Por exemplo, dar o nome **`folder-level-1/folder-level-2`** a uma nova pasta cria uma pasta de nível superior chamada **`folder-level-1`**, com outra pasta chamada **`folder-level-2`** dentro dela. A consulta é salva em **`folder-level-2`**.
9. (Opcional) Altere os grupos de log da consulta ou o texto da consulta.
10. Escolha Salvar.

Para excluir uma consulta, você deve estar conectado a uma função que tenha a permissão `logs:DeleteQueryDefinition`

Como editar ou excluir uma consulta salva

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.
3. À direita, escolha Consultas.
4. Selecione sua consulta na lista de Saved queries (Consultas salvas). Ela aparece no editor de consulta.
5. Escolha Ações, Editar ou Ações, Excluir.

Consultas de exemplo

Esta seção inclui consultas de exemplo que mostram a eficiência do CloudWatch Logs Insights.

Consultas gerais

Encontre os 25 eventos de log adicionados mais recentemente.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

Obtenha uma lista do número de exceções por hora.

```
filter @message like /Exception/
    | stats count(*) as exceptionCount by bin(1h)
    | sort exceptionCount desc
```

Obtenha uma lista de eventos de log que não sejam exceções.

```
fields @message | filter @message not like /Exception/
```

Consultas de logs doLambda

Determine a quantidade de memória provisionada excessivamente.

```
filter @type = "REPORT"
    | stats max(@memorySize / 1000 / 1000) as provisonedMemoryMB,
        min(@maxMemoryUsed / 1000 / 1000) as smallestMemoryRequestMB,
        avg(@maxMemoryUsed / 1000 / 1000) as avgMemoryUsedMB,
        max(@maxMemoryUsed / 1000 / 1000) as maxMemoryUsedMB,
        provisonedMemoryMB - maxMemoryUsedMB as overProvisionedMB
```

Crie um relatório de latência.

```
filter @type = "REPORT" |
    stats avg(@duration), max(@duration), min(@duration) by bin(5m)
```

Consultas de logs de fluxo daAmazon VPC

Encontre as 15 principais transferências de pacotes nos hosts:

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

Encontre as 15 principais transferências de bytes para hosts em uma determinada sub-rede.

```
filter isIpv4InSubnet(srcAddr, "192.0.2.0/24")
    | stats sum(bytes) as bytesTransferred by dstAddr
    | sort bytesTransferred desc
    | limit 15
```

Encontre os endereços IP que usam o UDP como um protocolo de transferência de dados.

```
filter protocol=17 | stats count(*) by srcAddr
```

Encontre os endereços IP nos quais os registros do fluxo foram ignorados durante a janela de captura.

```
filter logStatus="SKIPDATA"
    | stats count(*) by bin(1h) as t
    | sort t
```

Consultas de logs doRoute 53

Encontre a distribuição de registros por hora por tipo de consulta.

```
stats count(*) by queryType, bin(1h)
```

Encontre os 10 resolvedores DNS com o maior número de solicitações.

```
stats count(*) as numRequests by resolverIp
    | sort numRequests desc
    | limit 10
```

Encontre o número de registros por domínio e subdomínio em que o servidor deixou de concluir a solicitação DNS.

```
filter responseCode="SERVFAIL" | stats count(*) by queryName
```

Consultas de logs doCloudTrail

Encontre o número de entradas de log de cada serviço, tipo de evento e região da AWS.

```
stats count(*) by eventSource, eventName, awsRegion
```

Encontre os hosts do Amazon EC2 que foram iniciados ou interrompidos em uma região da AWS.

```
filter (eventName="StartInstances" or eventName="StopInstances") and region="us-east-2"
```

Encontre as regiões da AWS, os nomes de usuário e o ARNs de usuários do IAM recém-criados.

```
filter eventName="CreateUser"
    | fields awsRegion, requestParameters.userName, responseElements.user.arn
```

Encontre o número de registros em que ocorreu uma exceção durante a invocação da API `UpdateTrail`.

```
filter eventName="UpdateTrail" and ispresent(errorCode)
    | stats count(*) by errorCode, errorMessage
```

Exemplos do comando de análise

Use uma expressão de glob para extrair os campos efêmeros `@user`, `@method` e `@latency` do campo de log `@message` e retornar a latência média para cada combinação exclusiva de `@method` e `@user`.

```
parse @message "user=*, method:*, latency := *" as @user,
    @method, @latency | stats avg(@latency) by @method,
    @user
```

Use uma expressão regular para extrair os campos efêmeros `@user2`, `@method2` e `@latency2` do campo de log `@message` e retornar a latência média para cada combinação exclusiva de `@method2` e `@user2`.

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(latency2) by @method2,
    @user2
```

Extrai os campos efêmeros `loggingTime`, `loggingType` e `loggingMessage`, aplica o filtro para registrar eventos que contenham as strings `ERROR` ou `INFO` e exibe apenas os campos `loggingMessage` e `loggingType` para eventos que contenham uma string `ERROR`

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
```

```
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

Adicionar consulta ao painel ou exportar resultados da consulta

Depois de executar uma consulta, você pode adicionar a consulta a um painel do CloudWatch ou copiar os resultados para a área de transferência.

As consultas adicionadas aos painéis são executadas sempre que você carrega o painel e sempre que o painel é atualizado. Essas consultas entram na conta do limite de 10 consultas do CloudWatch Logs Insights simultâneas.

Para adicionar resultados da consulta a um painel

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.
3. Escolha um ou mais grupos de logs e execute uma consulta.
4. Escolha Add to dashboard (Adicionar ao painel).
5. Selecione o painel ou escolha Create new (Criar novo) para criar um painel para os resultados da consulta.
6. Selecione o tipo de widget a ser usado para os resultados da consulta.
7. Insira um nome para o widget.
8. Escolha Add to dashboard (Adicionar ao painel).

Como copiar os resultados da consulta para a área de transferência ou fazer download dos resultados da consulta

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.
3. Escolha um ou mais grupos de logs e execute uma consulta.
4. Escolha Export results (Exportar resultados) e escolha a opção desejada.

Exibir consultas em execução ou o histórico de consultas

Exiba as consultas atualmente em andamento, bem como o histórico de consultas recente.

As consultas em execução no momento incluem as consultas adicionadas a um painel. Você está limitado a 10 consultas CloudWatch Logs Insights simultâneas por conta, incluindo consultas adicionadas aos painéis.

Para exibir o histórico de consultas recente

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Insights.
3. Escolha History (Histórico) se estiver usando o novo design do console do CloudWatch Logs. Se estiver usando o design antigo, escolha Ações, Exibir histórico de consultas para esta conta.

Uma lista das consultas recentes é exibida. Você pode executar qualquer um deles novamente selecionando a consulta e escolhendo Executar.

Em Status, o CloudWatch Logs exibe Em andamento para todas as consultas que estão sendo executadas no momento.

Trabalhar com grupos de logs e fluxos de log

Stream de log é uma sequência de eventos de log que compartilham a mesma origem. Cada origem separada de logs no CloudWatch Logs compõe um fluxo de log separado.

Um grupo logs é um grupo de fluxos de log que compartilham as mesmas configurações de retenção, monitoramento e controle de acesso. Você pode definir grupos de logs e especificar quais fluxos colocar em cada grupo. Não há limite para o número de streams de log que podem pertencer a um grupo de logs.

Use os procedimentos desta seção para trabalhar com grupos e fluxos de logs.

Criar um grupo de logs no CloudWatch Logs

Quando você instalar o agente do CloudWatch Logs em uma instância do Amazon EC2 usando as etapas em seções anteriores do Amazon CloudWatch Logs User Guide, o grupo de logs será criado como parte do processo. Você também pode criar um grupo de logs diretamente no console do CloudWatch

Para criar um grupo de logs

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Selecione Actions (Ações) e selecione Create log group (Criar grupo de logs).
4. Digite um nome para o grupo de logs e escolha Create log group (Criar grupo de logs).

Enviar logs do a um grupo de logs do

O CloudWatch Logs AWS recebe automaticamente eventos de log de vários serviços da . Você também pode enviar outros eventos de log para o CloudWatch Logs usando um dos seguintes métodos:

- Agente do CloudWatch O agente unificado do pode enviar métricas e logs para o —.CloudWatchCloudWatch Logs Para obter informações sobre como instalar e usar o agente do CloudWatch, consulte Coleta de métricas e logs de instâncias do Amazon EC2 e de servidores no local com o agente do CloudWatch no Guia do usuário do Amazon CloudWatch.
- AWS CLI—O put-log-events faz upload de lotes de eventos de log para o CloudWatch Logs.
- Programmatically A API —PutLogEvents permite que você faça upload de forma programática de lotes de eventos de log para o .CloudWatch Logs

Visualizar os dados de log enviados para o CloudWatch Logs

Você pode visualizar e percorrer dados de log fluxo por fluxo ao serem enviados para o CloudWatch Logs pelo agente do CloudWatch Logs Você pode especificar o intervalo de tempo para os dados de log a serem visualizados.

Para visualizar dados de log

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Em Grupos de logs, escolha o grupo de logs para visualizar os fluxos.
4. Na lista de grupos de logs, escolha o nome do grupo de logs que deseja visualizar.
5. Na lista de fluxos de logs, escolha o nome do fluxo de log que deseja visualizar.
6. Para alterar a forma como os dados de log são exibidos, faça o seguinte:

   - Para expandir um único evento de log, escolha a seta ao lado dele.
   - Para expandir todos os eventos de log e visualizá-los como texto simples, acima da lista de eventos de log, escolha Texto.
   - Para filtrar os eventos de log, insira o filtro de pesquisa desejado no campo de pesquisa. Para obter mais informações, consulte Criar métricas a partir de eventos de log usando filtros (p. 71).
   - Para visualizar dados de log de um intervalo de data e hora especificado, escolha a seta ao lado da data e hora ao lado do filtro de pesquisa. Para especificar um intervalo de data e hora, escolha Absolute (Absoluto). Para escolher um número predefinido de minutos, horas, dias ou semanas, escolha Relative (Relativo). Também é possível alternar entre UTC e fuso horário local.

Pesquisar dados de log usando padrões de filtro

Você pode pesquisar seus dados de log usando Sintaxe do padrão e do filtro (p. 72). Você pode pesquisar todos os fluxos de log em um grupo de logs ou também pode pesquisar fluxos de log específicos usando a AWS CLI Na execução de cada pesquisa, ela retorna para a primeira página de dados encontrada e um token para recuperar a próxima página de dados ou continuar a pesquisa. Se não houver resultados retornados, você poderá continuar a pesquisar.

Você pode definir o intervalo de tempo que deseja consultar para limitar o escopo da pesquisa. Você pode começar com um intervalo maior para ver onde se encontram as linhas de log de interesse e, em seguida, reduzir o intervalo de tempo para limitar a exibição aos logs no intervalo de tempo de interesse.

Você também pode passar diretamente de suas métricas extraídas dos logs para os logs correspondentes.

Pesquisar entradas de log usando o console

Você pode procurar entradas de log que atendam a critérios especificados usando o console.

Para pesquisar seus logs usando o console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Em Log Groups (Grupos de logs), escolha o nome do grupo de logs que contém o fluxo de logs a ser pesquisado.
4. Em Fluxos de log, escolha o nome do fluxo de log para pesquisa.
5. Em Log events (Eventos de log), insira a sintaxe de filtro a ser usada.

Para pesquisar todas as entradas de log em um intervalo de tempo usando o console

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Em Log Groups (Grupos de logs), escolha o nome do grupo de logs que contém o fluxo de logs a ser pesquisado.

4. Escolha Pesquisar grupo de logs.
5. Em Log events (Eventos de log), selecione o intervalo de data e hora e insira a sintaxe do filtro.

Pesquisar entradas de log usando a AWS CLI

Você pode procurar entradas de log que atendam a critérios especificados usando a AWS CLI.

Para pesquisar entradas de log usando a AWS CLI

Em um prompt de comando, execute o seguinte comando filter-log-events. Use `--filter-pattern` para limitar os resultados para o padrão de filtro especificado e `--log-stream-names` para limitar os resultados para o grupo de logs especificado.

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] --filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Para pesquisar entradas de log em um determinado intervalo de tempo usando a AWS CLI

Em um prompt de comando, execute o seguinte comando filter-log-events:

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--start-time 1482197400000] [--end-time 1482217558365]
 [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Passar de métricas para logs

Você pode obter entradas de log específicas de outras partes do console.

Para passar de widgets do painel para logs

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Painéis.
3. Escolha um painel.
4. No widget, escolha Exibir logs e, em seguida, escolha Exibir logs neste período. Se houver mais de um filtro de métrica, selecione um na lista. Se houver mais filtros de métrica do que podemos exibir na lista, escolha Mais filtros de métrica e selecione ou procure um filtro de métrica.

Para passar de métricas para logs

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. No campo de pesquisa na guia Todas as métricas, digite o nome da métrica e pressione Enter.
4. Selecione uma ou mais métricas nos resultados da pesquisa.
5. Escolha Ações, Exibir logs. Se houver mais de um filtro de métrica, selecione um na lista. Se houver mais filtros de métrica do que podemos exibir na lista, escolha Mais filtros de métrica e selecione ou procure um filtro de métrica.

Troubleshooting

A pesquisa leva muito tempo para ser concluída

Se você tiver uma grande quantidade de dados de log, pode demorar muito tempo para a pesquisa ser concluída. Para acelerar uma pesquisa, você pode fazer o seguinte:

- Se você está usando a AWS CLI, pode limitar a pesquisa apenas aos fluxos de logs nos quais está interessado. Por exemplo, se o grupo de logs tem 1.000 fluxos de logs, mas você deseja ver apenas três fluxos de logs que considera relevantes, use a AWS CLI para limitar a pesquisa apenas a esses fluxos de logs no grupo.
- Use um intervalo de tempo menor, mais granular, o que reduz a quantidade de dados a serem pesquisados e acelera a consulta.

Alterar a retenção do log de dados no CloudWatch Logs

Por padrão, os dados de log são armazenados no CloudWatch Logs indefinidamente. No entanto, você pode configurar quanto tempo armazenar os dados de log em um grupo de logs. Todos os dados mais antigos do que a configuração de retenção atual são automaticamente excluídos. É possível alterar a retenção de logs de cada grupo de logs a qualquer momento.

Para alterar a configuração de retenção de logs

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Localize o grupo de logs a ser atualizado.
4. Na coluna Expirar eventos depois de correspondente ao grupo de logs em questão, escolha a configuração de retenção atual, como Nunca expirar.
5. Em Edit Retention (Editar retenção), para Retention (Retenção), escolha um valor de retenção de log e selecione Ok.

Marcar grupos de logs no Amazon CloudWatch Logs

Você pode atribuir seus próprios metadados aos grupos de logs que cria no Amazon CloudWatch Logs na forma de tags. Marca é um par de chave-valor que você define para um grupo de logs. O uso de marcas é uma forma simples, mas eficiente, de gerenciar os recursos da AWS e organizar os dados, incluindo dados de faturamento.

Tópicos

Conceitos básicos de tags

Você usa a AWS CLI ou a API do CloudWatch Logs para concluir as seguintes tarefas:

- Adicione tags a um grupo de logs ao criá-lo.
- Adicione tags a um grupo de logs existente.
- Liste as tags para um grupo de logs.
- Remova tags de um grupo de logs.

Você pode usar marcas para categorizar seus grupos de logs. Por exemplo, você pode categorizá-las 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 marcas que ajude a monitorar os grupos de log por proprietário e aplicativo associado. Aqui estão alguns exemplos de marcas:

- Projeto: nome do projeto
- Proprietário: nome
- Objetivo: testes de carga
- Aplicativo: nome do aplicativo
- Ambiente: produção

Monitoramento de custos com marcação

Você pode usar marcas para categorizar e monitorar seus custos da AWS. Quando você aplica marcas aos seus recursos da AWS, incluindo grupos de logs, o relatório de alocação de custos da AWS inclui o uso e os custos agrupados por marcas. É possível aplicar tags que representem categorias de negócios (como centros de custos, nomes de aplicativos ou proprietários) para organizar seus custos de vários serviços. Para obter mais informações, consulte Usar tags de alocação de custos para relatórios de faturamento personalizados no Guia do usuário do AWS Billing and Cost Management.

Restrições de tag

As restrições a seguir se aplicam a marcas.

Restrições básicas

- O número máximo de marcas por grupo de logs é 50.
- As chaves e os valores de tags diferenciam maiúsculas de minúsculas.
- Você não pode alterar nem editar as marcas de um grupo de logs 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.
- Você não pode iniciar uma chave de marca com `aws:` porque esse prefixo é reservado para uso pela AWS. A AWS cria marcas que começam com esse prefixo em seu nome, mas você não pode editá-las nem 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: `_ . / = + - @`.

Uso de tags em grupos de logs usando a AWS CLI

Você pode adicionar, listar e remover tags usando a AWS CLI. Para obter exemplos, consulte a seguinte documentação:

create-log-group

Cria um grupo de logs. Você também pode adicionar marcas ao criar o grupo de logs.

tag-log-group

Adiciona ou atualiza as marcas para o grupo de logs especificado.

list-tags-log-group

Lista as marcas para o grupo de logs especificado.

untag-log-group

Remove as marcas do grupo de logs especificado.

Uso de tags em grupos de logs usando a API do CloudWatch Logs

Você pode adicionar, listar e remover tags usando a API do CloudWatch Logs Para obter exemplos, consulte a seguinte documentação:

CreateLogGroup

Cria um grupo de logs. Você também pode adicionar marcas ao criar o grupo de logs.

TagLogGroup

Adiciona ou atualiza as marcas para o grupo de logs especificado.

ListTagsLogGroup

Lista as marcas para o grupo de logs especificado.

UntagLogGroup

Remove as marcas do grupo de logs especificado.

Criptografar dados de log no CloudWatch Logs usando o AWS Key Management Service

Os dados do grupo de logs são sempre criptografados no CloudWatch Logs. Você também pode usar o AWS Key Management Service para essa criptografia. Se você fizer isso, a criptografia será feita usando uma chave mestra de cliente (CMK) do AWS KMS. A criptografia usando o AWS KMS é habilitada no nível do grupo de logs associando uma CMK a um grupo de logs, ou quando você cria o grupo de logs ou depois.

> **Important**
>
> O CloudWatch Logs agora oferece suporte ao contexto de criptografia, usando `kms:EncryptionContext:aws:logs:arn` como a chave e o ARN do grupo de logs como o valor dessa chave. Se você tiver grupos de logs que já criptografou com uma CMK e quiser restringir a CMK a ser usada com uma única conta e grupo de logs, atribua uma nova CMK que inclua uma condição na política do IAM Para obter mais informações, consulte Chaves do AWS KMS e contexto de criptografia (p. 68).

Depois que você associar uma CMK a um grupo de logs, todos os novos dados consumidos para o grupo de logs serão criptografados usando a CMK. Esses dados são armazenados em formato criptografado durante todo o período de retenção. O CloudWatch Logs descriptografa esses dados sempre que eles são solicitados. O CloudWatch Logs deve ter permissões para a CMK sempre que dados criptografados são solicitados.

Depois de desassociar uma CMK de um grupo de logs, o CloudWatch Logs interrompe a criptografia de dados consumidos recentemente para o grupo de logs. Todos os dados consumidos anteriormente permanecem criptografados.

> **Important**
>
> O CloudWatch Logs CMKs oferece suporte apenas a simétricas. Não use uma CMK assimétrica para criptografar os dados em seus grupos de logs. Para obter mais informações, consulte Usar chaves simétricas e assimétricas.

Limits

- Para executar as etapas a seguir, é necessário ter as seguintes permissões: `kms:CreateKey`, `kms:GetKeyPolicy` e `kms:PutKeyPolicy`.
- Depois de associar ou desassociar uma CMK de um grupo de logs, pode levar até cinco minutos para que a operação tenha efeito.
- Se você revogar o acesso do CloudWatch Logs a uma CMK associada ou excluir uma CMK associada, não será mais possível recuperar seus dados criptografados no CloudWatch Logs
- Você não pode associar uma CMK a um grupo de logs usando o console do CloudWatch

Etapa 1: criar uma CMK do AWS KMS

Para criar uma CMK do AWS KMS, use o seguinte comando create-key:

```
aws kms create-key
```

A saída contém o ID de chave e o Nome de recurso da Amazon (ARN) da CMK. A seguir está um exemplo de saída:

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
```

```
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Etapa 2: Definir permissões na CMK

Por padrão, todos os AWS KMS CMKs são privados. Somente o proprietário do recurso pode usá-la para criptografar e descriptografar dados. No entanto, proprietário do recurso pode conceder permissões para acessar a CMK a outros usuários e recursos. Com esta etapa, você dá ao serviço CloudWatch a permissão principal para usar a CMK. O principal desse serviço deve estar na mesma região da AWS onde a CMK está armazenada.

Como prática recomendada, recomendamos restringir o uso da chave somente às contas da AWS ou grupos de logs especificados.

Primeiro, salve a política padrão para sua CMK como `policy.json` usando o seguinte comando get-key-policy:

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./policy.json
```

Abra o arquivo `policy.json` em um editor de texto e adicione a seção em negrito de uma das instruções a seguir. Separe a instrução existente da nova instrução com uma vírgula. Essas declarações usam seções `Condition` para aumentar a segurança da chave AWS KMS. Para obter mais informações, consulte Chaves do AWS KMS e contexto de criptografia (p. 68).

A seção `Condition` neste exemplo restringe a chave a um único ARN de grupo de logs.

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:log-group:log-group-name"
                }
```

```
            }
        }
    ]
}
```

A seção `Condition` neste exemplo limita o uso da chave AWS KMS à conta especificada, mas ela pode ser usada para qualquer grupo de logs.

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:*"
                }
            }
        }
    ]
}
```

Por fim, adicione a política atualizada usando o seguinte comando put-key-policy:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://policy.json
```

Etapa 3: Associar um grupo de logs com uma CMK

É possível associar uma CMK a um grupo de logs ao criá-la ou posteriormente.

Para saber se um grupo de logs já tem uma CMK associada, use o seguinte comando describe-log-groups:

```
aws logs describe-log-groups --log-group-name-prefix "log-group-name-prefix"
```

Se a saída incluir um campo `kmsKeyId`, o grupo de logs será associado à chave exibida para o valor desse campo.

Para associar a CMK a um grupo de logs ao criá-lo

Use o comando create-log-group da seguinte forma:

```
aws logs create-log-group --log-group-name my-log-group --kms-key-id "key-arn"
```

Para associar a CMK a um grupo de logs existente

Use o comando associate-kms-key da seguinte forma:

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id "key-arn"
```

Etapa 4: Desassociar um grupo de logs de uma CMK

Para desassociar a CMK associada a um grupo de logs, use o seguinte comando disassociate-kms-key:

```
aws logs disassociate-kms-key --log-group-name my-log-group
```

Chaves do AWS KMS e contexto de criptografia

Para aumentar a segurança das chaves do AWS Key Management Service e dos grupos de logs criptografados, o CloudWatch Logs agora coloca o grupo de logs ARNs como parte do contexto de criptografia usado para criptografar dados de log. O contexto de criptografia é um conjunto de pares de chave/valor que são usados como dados autenticados adicionais. O contexto de criptografia permite que você use condições de política do IAM para limitar o acesso à sua chave do AWS KMS por conta da AWS e grupo de logs. Para obter mais informações, consulte Contexto de criptografia e Elementos de política JSON do IAM: condição.

Recomendamos usar chaves do CMK diferentes para cada grupo de logs criptografado.

Se você tiver um grupo de logs criptografado anteriormente e agora deseja alterar o grupo de logs para usar uma nova CMK que funcione somente para esse grupo, siga estas etapas.

Como converter um grupo de logs criptografado para usar uma CMK com uma política limitando-a a esse grupo de logs

1. Digite o comando a seguir para localizar o ARN da CMK atual do grupo de logs:

   ```
   aws logs describe-log-groups
   ```

   A saída inclui a linha a seguir. Anote o ARN. Ele será necessário na etapa 7.

   ```
   ...
   "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/01234567-89ab-
   cdef-0123-456789abcdef"
   ...
   ```

2. Digite o comando a seguir para criar uma nova CMK:

   ```
   aws kms create-key
   ```

3. Digite o comando a seguir para salvar a política da nova chave em um arquivo `policy.json`

   ```
   aws kms get-key-policy --key-id new-key-id --policy-name default --output text > ./
   policy.json
   ```

4. Use um editor de texto para abrir `policy.json` e adicionar uma expressão `Condition` à política:

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::ACCOUNT-ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:REGION:ACCOUNT-
ID:log-
group:LOG-GROUP-NAME"
                }
            }
        }
    ]
}
```

5. Digite o comando a seguir para adicionar a política atualizada à nova CMK:

```
aws kms put-key-policy --key-id new-key-ARN --policy-name default --policy file://
policy.json
```

6. Digite o comando a seguir para associar a política ao seu grupo de logs:

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id new-key-ARN
```

CloudWatch LogsO agora criptografa todos os novos dados usando a nova CMK.

7. Depois, revogue todas as permissões, exceto `Decrypt` da CMK antiga. Primeiro, digite o comando a seguir para recuperar a política antiga:

```
aws kms get-key-policy --key-id old-key-ARN --policy-name default --output text > ./
policy.json
```

8. Use um editor de texto para abrir `policy.json` e remover todos os valores da lista `Action`, exceto `kms:Decrypt*`

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
```

```
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::REGION:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt*"
            ],
            "Resource": "*"
        }
    ]
}
```

9. Digite o comando a seguir para adicionar a política atualizada à antiga CMK:

```
aws kms put-key-policy --key-id old-key-ARN --policy-name default --policy file://
policy.json
```

Criar métricas a partir de eventos de log usando filtros

Depois que o agente do CloudWatch Logs começar a publicação dos dados de log no Amazon CloudWatch, você pode começar a pesquisar e filtrar os dados de log, criando um ou mais filtros de métrica. Os filtros de métrica definem os termos e os padrões a serem procurados nos dados de log à medida que são enviados para CloudWatch Logs. CloudWatch Logs usa esses filtros de métrica para transformar os dados de log em métricas numéricas do CloudWatch que podem ser usadas para criar um gráfico ou definir um alarme. Você pode usar qualquer tipo de estatística do CloudWatch, incluindo estatísticas de percentil, ao visualizar essas métricas ou definir alarmes.

> Note
>
> As estatísticas de percentil serão compatíveis com uma métrica somente se nenhum dos valores da métrica for negativo. Se você configurar o filtro de métrica para que ele possa relatar números negativos, as estatísticas de percentil não estarão disponíveis para essa métrica quando ela tiver números negativos como valores. Para obter mais informações, consulte Percentis.

Os filtros não funcionam de forma retroativa nos dados. Eles só publicam os pontos de dados de métrica para eventos que ocorrem após a criação do filtro. Os resultados filtrados retornam as primeiras 50 linhas, que não serão exibidas se o time stamp nos resultados filtrados for anterior à criação da métrica.

Tópicos

Concepts

Cada filtro de métrica é composto dos seguintes elementos-chave:

padrão de filtro

> Descrição simbólica de como o CloudWatch Logs deve interpretar os dados em cada evento de log. Por exemplo, uma entrada de log pode conter time stamps, endereços IP, sequências de caracteres, etc. Você pode usar o padrão para especificar o que procurar no arquivo de log.

nome da métrica

> O nome da métrica do CloudWatch com o qual as informações do log monitoradas devem ser publicadas. Por exemplo, você pode publicar em uma métrica chamada ErrorCount.

namespace de métrica

> O namespace de destino da nova métrica do CloudWatch

valor da métrica

> O valor numérico para publicar a métrica cada vez que um log correspondente é encontrado. Por exemplo, se você contabilizar as ocorrências de um determinado termo, como "Erro", o valor será "1" para cada ocorrência. Se você estiver contando os bytes transferidos, poderá incrementar pelo número real de bytes encontrado no evento de log.

valor padrão

O valor indicado para o filtro de métrica durante um período quando não forem encontrados logs correspondentes. Ao definir esse valor como 0, você garante que os dados são relatados durante cada período, impedindo métricas irregulares em períodos sem dados.

Sintaxe do padrão e do filtro

Você pode usar filtros de métrica para procurar e relacionar termos, frases ou valores em seus eventos de log. Quando um filtro de métrica encontra um dos termos, frases ou valores nos seus eventos de log, você pode incrementar o valor de uma métrica do CloudWatch Por exemplo, você pode criar um filtro de métrica para pesquisar e contar a ocorrência da palavra ERRO em seus eventos de log.

Os filtros de métrica também podem extrair valores numéricos dos eventos de log limitados por espaço, como a latência das solicitações da web. Nesses exemplos, você pode aumentar o seu valor de métrica pelo valor numérico real extraído do log.

Você também pode usar operadores condicionais e curingas para criar correspondências exatas. Antes de criar um filtro de métrica, você pode testar seus padrões de pesquisa no console do CloudWatch As seções a seguir explicam a sintaxe do filtro da métrica em mais detalhes.

Correspondência de termos em eventos de log

Para procurar um termo nos seus eventos de log, use o termo como seu padrão de filtro de métrica. Você pode especificar vários termos em um filtro de métrica padrão, mas todos os termos devem aparecer em um evento de log para haver uma correspondência. Os filtros de métrica fazem distinção de maiúsculas e minúsculas.

Os termos de filtro de métrica que incluem caracteres que não sejam alfanuméricos ou sublinhado devem ser colocados entre aspas duplas ("").

Para excluir um termo, use um sinal de menos (-) antes dele.

Exemplo 1: correspondência de todos

O padrão de filtro """" corresponde todos os eventos de log.

Exemplo 2: Termo único

O padrão de filtro "ERRO" faz a correspondência das mensagens de eventos de log que contêm esse termo da seguinte forma:

- [ERRO] Ocorreu uma exceção fatal
- Saindo com ERRORCODE: -1

Exemplo 3: Incluir um termo e excluir um termo

No exemplo anterior, se você alterar o padrão de filtro para "ERRO" – "Saindo", a mensagem de evento de log "Saindo com ERRORCODE: -1" será excluída.

Exemplo 4: Vários termos

O padrão de filtro "ERRO Exceção" faz a correspondência de mensagens de eventos de log que contêm os dois termos, como o seguinte:

- [ERRO] Capturado IllegalArgumentException

- [ERRO] Exceção não tratada

O padrão de filtro "Falha ao processar a solicitação" faz a correspondência de mensagens de eventos de log que contêm todos os termos, como o seguinte:

- [AVISO] Falha ao processar a solicitação
- [ERRO] Não foi possível continuar: falha ao processar a solicitação

Correspondência de padrão OU

Você pode corresponder termos em filtros com base em texto usando a correspondência de padrão OU. Use um ponto de interrogação para OR, como `?term`.

Examine os três exemplos de eventos de log abaixo. `ERROR` corresponde aos exemplos 1 e 2. `?ERROR ? WARN` corresponde aos exemplos 1, 2 e 3, pois todos eles incluem a palavra ERROR ou a palavra WARN. `ERROR WARN` só corresponde ao exemplo 1, pois é o único contendo as duas palavras. ERRO -WARN corresponde ao exemplo 2, pois corresponde a uma string que contém ERROR, mas não contém WARN.

1. `ERROR WARN message`
2. `ERROR message`
3. `WARN message`

Você pode corresponder termos usando a correspondência de padrão OU em filtros delimitados por espaços. Com filtros delimitados por espaço, `w1` indica a primeira palavra no evento de log, `w2` indica a segunda palavra e assim por diante. Para os padrões de exemplo abaixo, [w1=ERROR, w2] corresponde aos padrões 1 e 2, pois ERROR é a primeira palavra, e [w1=ERROR || w1=WARN, w2] corresponde aos padrões 1, 2 e 3. [w1!= ERRO&&w1!= AVISO, w2] não corresponde a nenhuma das linhas porque todas contêm ERRO ou AVISO.

1. Mensagem de AVISO DE ERRO
2. Mensagem de ERRO
3. Mensagem de AVISO

Você pode corresponder termos usando a correspondência de padrão OU em filtros JSON. Para os padrões de exemplo a seguir, {$.foo = bar} corresponde ao padrão 1, {$.foo = baz } corresponde ao padrão 2 e {$.foo = bar || $.foo = baz } corresponde aos padrões 1 e 2.

1. {"foo": "bar"}
2. {"foo": "baz"}

Correspondência de termos em eventos de log JSON

Você pode extrair valores de eventos de log JSON. Para extrair valores de eventos de log JSON, você precisa criar um filtro de métrica com base em string. Strings que contêm notação científica não têm suporte. Os itens nos dados de eventos de log JSON devem corresponder exatamente ao filtro de métrica. É possível criar filtros de métrica em eventos de log JSON para indicar o seguinte:

- Ocorre um determinado evento. Por exemplo, eventName é "UpdateTrail".
- O IP está fora de uma sub-rede conhecida. Por exemplo, sourceIPAddress não está no intervalo de alguma sub-rede conhecida.
- Uma combinação de duas ou mais condições são verdadeiras. Por exemplo, o eventName é "UpdateTrail" e o recipientAccountId é 123456789012.

Uso de filtros de métrica para extrair valores de eventos de log JSON

Você pode usar filtros de métrica para extrair valores de eventos de log JSON. Um filtro de métrica verifica os logs de entrada e modifica um valor numérico quando encontra uma correspondência nos dados de log. Quando você cria um filtro de métrica, pode simplesmente incrementar uma contagem cada vez que o texto correspondente é encontrado em um log ou pode extrair valores numéricos do log e usá-los para incrementar o valor da métrica.

Correspondência de termos JSON com filtros de métricas

A sintaxe do filtro de métrica para eventos de log JSON usa o seguinte formato:

```
{ SELECTOR EQUALITY_OPERATOR STRING }
```

O filtro de métrica deve estar entre chaves { }, para indicar que essa é uma expressão JSON. O filtro de métrica contém as seguintes partes:

SELETOR

Especifica o que a propriedade JSON deve verificar. Os seletores de propriedade sempre começam com o sinal de cifrão ($), o que significa a raiz do JSON. Os seletores de propriedade são strings alfanuméricas que também dão suporte aos caracteres "-" e "_". Os elementos de matriz são indicados com a sintaxe [NUMBER] e devem seguir a uma propriedade. São exemplos: $.eventId, $.users[0], $.users[0].id, $.requestParameters.instanceId.

EQUALITY_OPERATOR

Pode ser = ou !=.

STRING

Uma string com ou sem aspas. Você pode usar o caractere curinga asterisco '*" para fazer a correspondência com qualquer texto em, antes ou depois de um termo de pesquisa. Por exemplo, *Event corresponderá a PutEvent e GetEvent. Event* corresponderá a EventId e EventName. Ev*ent só corresponderá à string real Ev*ent. Strings que consistem inteiramente em caracteres alfanuméricos não precisam de aspas. Strings que contém unicode e outros caracteres, como"@", "$","\" etc., devem estar entre aspas duplas para serem válidas.

Exemplos de filtro de métrica JSON

Veja um exemplo de JSON a seguir:

```
{
  "eventType": "UpdateTrail",
  "sourceIPAddress": "111.111.111.111",
  "arrayKey": [
        "value",
        "another value"
  ],
  "objectList": [
       {
         "name": "a",
         "id": 1
       },
       {
         "name": "b",
         "id": 2
       }
  ],
  "SomeObject": null,
  "ThisFlag": true
```

```
}
```

Os seguintes filtros corresponderiam a:

```
{ $.eventType = "UpdateTrail" }
```

Filtrar com base no tipo de evento UpdateTrail.

```
{ $.sourceIPAddress != 123.123.* }
```

Filtrar com base no endereço IP fora do prefixo 123.123 da sub-rede.

```
{ $.arrayKey[0] = "value" }
```

Filtrar com base na primeira entrada em arrayKey sendo "valor". Se arrayKey não for uma matriz, será falso.

```
{ $.objectList[1].id = 2 }
```

Filtrar com base na segunda entrada em objectList tendo uma propriedade chamada id = 2. Se objectList não for uma matriz, será falso. Se os itens em objectList não forem objetos ou não tiverem uma propriedade id, isso será falso.

```
{ $.SomeObject IS NULL }
```

Filtrar com base em SomeObject sendo definido como nulo. Isso só será verdadeiro se o objeto especificado for definido como nulo.

```
{ $.SomeOtherObject NOT EXISTS }
```

Filtrar com base em SomeOtherObject sendo inexistente. Isso só será verdadeiro se o objeto especificado não existir nos dados de log.

```
{ $.ThisFlag IS TRUE }
```

Filtros em ThisFlag sendo TRUE. Isso também funciona para filtros boolianos que verificariam o valor FALSE.

Condições compostas JSON

Você pode combinar várias condições em uma expressão composta usando OR (||) e AND (&&). Parênteses são permitidos e a sintaxe segue a ordem padrão das operações () > && > ||.

```
{
    "user": {
        "id": 1,
        "email": "John.Stiles@example.com"
    },
    "users": [
        {
         "id": 2,
         "email": "John.Doe@example.com"
        },
        {
         "id": 3,
```

```
         "email": "Jane.Doe@example.com"
        }
    ],
    "actions": [
        "GET",
        "PUT",
        "DELETE"
    ],
    "coordinates": [
        [0, 1, 2],
        [4, 5, 6],
        [7, 8, 9]
    ]
}
```

Examples

```
{ ($.user.id = 1) && ($.users[0].email = "John.Doe@example.com") }
```

Corresponde ao JSON acima.

```
{ ($.user.id = 2 && $.users[0].email = "nonmatch") || $.actions[2] = "GET" }
```

Não corresponde ao JSON acima.

```
{ $.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = nonmatch &&
 $.actions[2] = nomatch }
```

Corresponde ao JSON acima.

```
{ ($.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = nonmatch) &&
 $.actions[2] = nomatch }
```

Não corresponde ao JSON acima.

Considerações especiais JSON

O SELETOR deve apontar para um nó de valor (string ou número) no JSON. Se apontar para uma matriz ou objeto, o filtro não será aplicado porque o formato do log não corresponderá ao filtro. Por exemplo, {$.users = 1} e {$.users! = 1} não corresponderão a um evento de log no qual os usuários são um array:

```
{
  "users": [1, 2, 3]
}
```

Comparações numéricas

A sintaxe do filtro de métrica oferece suporte à correspondência precisa em comparações numéricas. As seguintes comparações numéricas são compatíveis:**<**, **>**, **>=**, **<=**, **=**, **!=**

Os filtros numéricos têm uma sintaxe de

```
{ SELECTOR NUMERIC_OPERATOR NUMBER }
```

O filtro de métrica deve estar entre chaves { }, para indicar que essa é uma expressão JSON. O filtro de métrica contém as seguintes partes:

SELETOR

Especifica o que a propriedade JSON deve verificar. Os seletores de propriedade sempre começam com o sinal de cifrão ($), o que significa a raiz do JSON. Os seletores de propriedade são strings alfanuméricas que também dão suporte aos caracteres "-" e "_". Os elementos de matriz são indicados com a sintaxe [NUMBER] e devem seguir a uma propriedade. São exemplos: $.latency, $.numbers[0], $.errorCode, $.processes[4].averageRuntime.

NUMERIC_OPERATOR

Pode ser um dos seguintes: =, !=, <, >, <= ou >=.

NUMBER

Um valor inteiro com um sinal opcional +ou –, um valor decimal com um sinal opcional + ou – ou um número em uma notação específica, que é um número inteiro ou decimal com um sinal opcional + ou –, seguido por "e", por um valor inteiro com um sinal opcional + ou –.

Exemplos:

```
{ $.latency >= 500 }
{ $.numbers[0] < 10e3 }
{ $.numbers[0] < 10e-3 }
{ $.processes[4].averageRuntime <= 55.5 }
{ $.errorCode = 400 }
{ $.errorCode != 500 }
{ $.latency > +1000 }
```

Uso de filtros de métrica para extrair valores de eventos de log delimitados por espaço

Você pode usar filtros de métrica para extrair valores de eventos de log delimitados por espaço. Os caracteres entre um par de colchetes [] ou duas aspas duplas ("") são tratados como um único campo. Por exemplo:

```
127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] "GET /apache_pb.gif HTTP/1.0" 200 1534
127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] "GET /apache_pb.gif HTTP/1.0" 500 5324
127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] "GET /apache_pb.gif HTTP/1.0" 200 4355
```

Para especificar um padrão de filtro de métrica que analise eventos delimitados por espaço, o padrão de filtro de métrica precisa especificar os campos com um nome, separados por vírgulas, com todo o padrão entre colchetes. Por exemplo: [ip, user, username, timestamp, request, status_code, bytes].

Quando você não souber o número de campos, use uma notificação abreviada com reticências (...). Por exemplo:

```
[..., status_code, bytes]
[ip, user, ..., status_code, bytes]
[ip, user, ...]
```

Você também pode adicionar condições aos seus campos, para que apenas os eventos de log correspondentes a todas as condições coincidam com os filtros. Por exemplo:

```
[ip, user, username, timestamp, request, status_code, bytes > 1000]
[ip, user, username, timestamp, request, status_code = 200, bytes]
[ip, user, username, timestamp, request, status_code = 4*, bytes]
[ip, user, username, timestamp, request = *html*, status_code = 4*, bytes]
```

É possível usar `&&` como um operador lógico AND e `||` como um operador lógico OR, conforme os exemplos a seguir:

```
[ip, user, username, timestamp, request, status_code = 4* && bytes > 1000]
[ip, user, username, timestamp, request, status_code = 403 || status_code = 404, bytes]
```

CloudWatch LogsO oferece suporte a campos de string e numéricos condicionais. Para campos de string, você pode usar os operadores = ou != com um asterisco (*).

Para campos numéricos, você pode usar os operadores >, <, >=, <=, = e !=.

Se você estiver usando um filtro delimitado por espaço, os campos extraídos mapeados para os nomes dos campos delimitados por espaço (conforme expressos no filtro) terão o valor de cada um desses campos. Se você não estiver usando um filtro delimitado por espaço, ficará vazio.

Exemplo de filtro:

```
[..., request=*.html*, status_code=4*,]
```

Exemplo de evento de log para o filtro:

```
127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /index.html HTTP/1.0\" 404 1534
```

Campos extraídos para o evento de log e padrão de filtro:

```
{
   "$status_code": "404",
   "$request": "GET /products/index.html HTTP/1.0",
   "$7": "1534",
   "$4": "10/Oct/2000:13:25:15 -0700",
   "$3": "frank",
   "$2": "-",
   "$1": "127.0.0.1"
}
```

Definir como os valores de métrica mudam quando são encontradas correspondências

Quando um filtro de métrica encontra um dos termos, frases ou valores correspondentes nos seus eventos de log, ele incrementa a contagem na métrica do CloudWatch pela quantidade que você especificar para o valor de métrica. O valor da métrica é agregado e relatado a cada minuto.

Se forem ingeridos logs durante um período de um minuto, mas nenhuma correspondência for encontrada, o valor especificado para Valor Padrão (se houver) será relatado. No entanto, se nenhum evento de log for ingerido durante um período de um minuto, nenhum valor será relatado.

Especificar um valor padrão, mesmo se esse valor for 0, ajuda a garantir que os dados serão relatadas com mais frequência, ajudando a evitar métricas irregulares quando não são encontradas correspondências.

Por exemplo, suponha que há um grupo de logs que publica dois registros a cada minuto, o valor de métrica é 1 e o valor padrão é 0. Se as correspondências forem encontradas nos dois registros de log no primeiro minuto, o valor de métrica para aquele minuto será 2. Se não houver correspondências nos registros de log publicados no segundo minuto, o valor padrão de 0 será usado para os dois registros de log e o valor de métrica para aquele minuto será 0.

Se você não especificar um valor padrão, nenhum dado será relatado para qualquer período em que não haja correspondências de padrão.

Valores numéricos de publicação encontrados em entradas de log

Em vez de apenas contar o número de itens correspondentes encontrados em logs, você também pode usar o filtro de métrica para publicar valores com base nos valores numéricos encontrados nos logs. O procedimento a seguir mostra como publicar uma métrica com a latência encontrada na solicitação JSON `metricFilter: { $.latency = * } metricValue: $.latency`.

Para publicar uma métrica com a latência em uma solicitação JSON

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Escolha `Actions`, Criar filtro de métrica.
4. Em Padrão de filtro, digite **`{ $.latency = * }`** e escolha Próximo.
5. Em Metric Name (Nome da métrica), digite myMetric.
6. Em Valor da métrica, insira **`$.latency`**.
7. Para Valor padrão, insira 0 e escolha Próximo. Especificar um valor padrão garante que os dados sejam relatados mesmo em períodos em que nenhum evento de log corresponder ao filtro. Isso evita métricas irregulares ou ausentes quando os logs são ingeridos mas não correspondem ao filtro.
8. Escolha Criar filtro de métrica.

O evento de log a seguir publicaria um valor de 50 na métrica myMetric após a criação do filtro.

```
{
"latency": 50,
"requestType": "GET"
}
```

Criação de filtros de métrica

Os exemplos a seguir mostram como criar filtros de métrica.

Exemplos

- Exemplo: Contar eventos de log (p. 79)
- Exemplo: Contar as ocorrências de um termo (p. 80)
- Exemplo: Contar códigos HTTP 404 (p. 82)
- Exemplo: Contar códigos HTTP 4xx (p. 83)
- Exemplo: Extrair campos de um log Apache (p. 84)

Exemplo: Contar eventos de log

O tipo mais simples de monitoramento de eventos de log é contar o número de eventos de log ocorridos. É possível fazer isso para manter uma contagem de todos os eventos, para criar um monitor no estilo de "pulsação" ou apenas para praticar a criação de filtros de métrica.

No exemplo de CLI a seguir, um filtro de métrica chamado MyAppAccessCount é aplicado ao grupo de logs MyApp/access.log para criar a métrica EventCount no namespace CloudWatch do MyNamespace. O filtro é configurado para fazer a correspondência de qualquer conteúdo de eventos de log e incrementar a métrica em "1".

Para criar um filtro de métrica usando o console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Escolha o nome de um grupo de logs.
4. Escolha `Actions`, Criar filtro de métrica.
5. Deixe Filter Pattern e Select Log Data to Test em branco.
6. Escolha Próximo e, em Nome do filtro, digite **`EventCount`**.
7. Em Metric Details (Detalhes da métrica), em Metric Namespace (Namespace da métrica), digite **`MyNameSpace`**.
8. Para Metric Name (Nome da métrica), digite **`MyAppEventCount`**.
9. Confirme se o Valor da métrica é 1. Isso especifica que a contagem é aumentada em 1 para cada evento de log.
10. Para Valor padrão, insira 0 e escolha Próximo. Especificar um valor padrão garante que os dados são relatados mesmo durante períodos em que não ocorrem eventos de log. Isso evita métricas irregulares nas quais os dados, às vezes, não existem.
11. Escolha Criar filtro de métrica.

Para criar um filtro de métrica usando a AWS CLI

Em um prompt de comando, execute o seguinte comando:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name EventCount \
  --filter-pattern "" \
  --metric-transformations \
  metricName=MyAppEventCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Você pode testar essa nova política postando quaisquer dados de eventos. Você deve ver pontos de dados publicados na métrica MyAppAccessEventCount.

Para publicar dados de eventos usando a AWS CLI

Em um prompt de comando, execute o seguinte comando:

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="Test event 1" \
    timestamp=1394793518000,message="Test event 2" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Exemplo: Contar as ocorrências de um termo

Os eventos de log frequentemente incluem mensagens importantes que você deseja contar, talvez sobre o êxito ou a falha de operações. Por exemplo, poderá ocorrer um erro e ele ser registrado em um arquivo de log se ocorrer uma falha em uma determinada operação. É possível monitorar essas entradas para entender a tendência dos erros.

No exemplo abaixo, um filtro de métrica é criado para monitorar o termo Erro. A política foi criada e adicionada ao grupo de logs MyApp/message.log. O CloudWatch Logs publica um ponto de dados na métrica personalizada CloudWatch do ErrorCount no namespace MyApp/message.log com um valor de "1" para cada evento que contém Erro. Se não houver um evento com a palavra Erro, será publicado um valor de 0. Ao criar gráficos com esses dados no console do CloudWatch, certifique-se de usar a estatística de soma.

Depois de criar um filtro de métrica, você pode exibir a métrica no console do CloudWatch Ao selecionar a métrica a ser exibida, selecione o namespace da métrica que corresponde ao nome do grupo de logs. Para obter mais informações, consulte Visualizar métricas disponíveis.

Para criar um filtro de métrica usando o console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Escolha o nome do grupo de logs.
4. Escolha Ações, Criar filtro de métrica.
5. Em Padrão de filtro, insira **`Error`**.

   Note

   Todas as entradas em Padrão de filtro diferenciam maiúsculas de minúsculas.
6. Para testar seu padrão de filtro, escolha Testar padrão.
7. Escolha Próximo e, na página Atribuir métrica, em Nome do filtro, digite **`MyAppErrorCount`**.
8. Em Metric Details (Detalhes da métrica), para Metric Namespace (Namespace da métrica), digite MyNameSpace.
9. Em Metric Name (Nome da métrica), digite ErrorCount.
10. Confirme se o Metric Value (Valor da métrica) é 1. Isso especifica que a contagem é aumentada em 1 para cada evento de log que contém "Erro".
11. Para Valor padrão, digite 0 e escolha Próximo.
12. Escolha Criar filtro de métrica.

Para criar um filtro de métrica usando a AWS CLI

Em um prompt de comando, execute o seguinte comando:

```
aws logs put-metric-filter \
  --log-group-name MyApp/message.log \
  --filter-name MyAppErrorCount \
  --filter-pattern 'Error' \
  --metric-transformations \
      metricName=ErrorCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Você pode testar essa nova política postando eventos que contenham a palavra "Erro" na mensagem.

Para publicar eventos usando a AWS CLI

Em um prompt de comando, execute o seguinte comando. Os padrões fazem distinções de maiúsculas e minúsculas.

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="This message contains an Error" \
```

```
    timestamp=1394793528000,message="This message also contains an Error"
```

Exemplo: Contar códigos HTTP 404

Usando o CloudWatch Logs, você pode monitorar quantas vezes os servidores Apache retornam uma resposta HTTP 404, que é o código de resposta para página não encontrada. Você pode monitorar isso para entender a frequência com que os visitantes de seu site não encontram o recurso que procuram. Suponha que seus registros de log estejam estruturados para incluir as seguintes informações para cada evento de log (visita ao site):

- Endereço IP do solicitante
- Identidade RFC 1413
- Username
- Time stamp
- Método de solicitação com o recurso solicitado e o protocolo
- Código de resposta HTTP para a solicitação
- Bytes transferidos na solicitação

Um exemplo disso pode ter a seguinte aparência:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 404 2326
```

Você pode especificar uma regra que tente fazer a correspondência de eventos dessa estrutura para erros HTTP 404, como mostrado no exemplo a seguir:

Para criar um filtro de métrica usando o console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Escolha `Actions`, Criar filtro de métrica.
4. Em Padrão de filtro, digite **`[IP, UserInfo, User, Timestamp, RequestInfo, StatusCode=404, Bytes]`**.
5. Para testar seu padrão de filtro, escolha Testar padrão.
6. Escolha Próximo e, em Nome do filtro, digite HTTP404Errors.
7. Em Detalhes da métrica, para Namespace da métrica, insira **`MyNameSpace`**.
8. Em Nome da métrica, insira **`ApacheNotFoundErrorCount`**.
9. Confirme se o Valor da métrica é 1. Isso especifica que a contagem é aumentada em 1 para cada evento 404 Error.
10. Para Valor padrão, insira 0 e escolha Próximo.
11. Escolha Criar filtro de métrica.

Para criar um filtro de métrica usando a AWS CLI

Em um prompt de comando, execute o seguinte comando:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP404Errors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=404, size]' \
  --metric-transformations \
```

```
metricName=ApacheNotFoundErrorCount,metricNamespace=MyNamespace,metricValue=1
```

Neste exemplo, os caracteres literais, como os colchetes à direita e à esquerda, aspas duplas e string de caracteres 404 foram usados. O padrão precisa fazer a correspondência com toda a mensagem de evento de log para o evento de log a ser considerado para monitoramento.

Você pode verificar a criação do filtro de métrica usando o comando describe-metric-filters Você deve ver uma saída semelhante a:

```
aws logs describe-metric-filters --log-group-name MyApp/access.log

{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404, size]"
        }
    ]
}
```

Agora você pode publicar alguns eventos manualmente:

```
aws logs put-log-events \
--log-group-name MyApp/access.log --log-stream-name hostname \
--log-events \
timestamp=1394793518000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 404 2326" \
timestamp=1394793528000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb2.gif HTTP/1.0\" 200 2326"
```

Logo depois de colocar esses exemplos de eventos de log, você pode recuperar a métrica indicada no console do CloudWatch como ApacheNotFoundErrorCount.

Exemplo: Contar códigos HTTP 4xx

Como no exemplo anterior, você pode monitorar seus logs de acesso do serviço da Web e monitorar os níveis de código de resposta HTTP. Por exemplo, você pode monitorar todos os erros no nível de HTTP 400 erros. No entanto, é possível especificar um novo filtro de métrica para cada código de retorno.

O exemplo a seguir demonstra como criar uma métrica que inclua todas as 400 respostas de código HTTP a partir de um log de acesso usando o formato de log de acesso do Apache do exemplo Exemplo: Contar códigos HTTP 404 (p. 82)

Para criar um filtro de métrica usando o console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Escolha o nome do grupo de logs para o servidor Apache.
4. Escolha `Actions`, Criar filtro de métrica.
5. Em Filter name (Filtrar nome), insira **`HTTP4xxErrors`**.

6. Em Filter pattern (Padrão de filtro), insira **`[ip, id, user, timestamp, request, status_code=4*, size]`**.
7. Para testar seu padrão de filtro, escolha Testar padrão.
8. Escolha Next (Próximo) e, em Filter name (Nome do filtro), digite **`HTTP4xxErrors`**.
9. Em Metric details (Detalhes da métrica), para Metric namespace (Namespace da métrica), insira **`MyNameSpace`**.
10. Em Metric name (Nome da métrica), insira HTTP4xxErrors.
11. Em Metric value (Valor da métrica), insira 1. Isso especifica que a contagem é aumentada em 1 para cada log que contém um erro 4xx.
12. Em Default value (Valor padrão), insira 0 e escolha Next (Próximo).
13. Escolha Criar filtro de métrica.

Para criar um filtro de métrica usando a AWS CLI

Em um prompt de comando, execute o seguinte comando:

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP4xxErrors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=4*, size]' \
  --metric-transformations \
  metricName=HTTP4xxErrors,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Você pode usar os seguintes dados em chamadas put-event para testar essa regra. Se você não removeu a regra de monitoramento no exemplo anterior, gerará duas métricas diferentes.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Exemplo: Extrair campos de um log Apache

Às vezes, em vez de contar, é útil usar os valores em eventos de log individuais para valores de métrica. Este exemplo mostra como criar uma regra de extração para criar uma métrica que meça os bytes transferidos por um servidor web Apache.

Esta regra de extração faz a correspondência dos sete campos do evento de log. O valor da métrica é o valor do sétimo token correspondente. Você pode ver a referência para o token como "$7" no campo `metricValue` da regra de extração.

Para criar um filtro de métrica usando o console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Escolha o nome do grupo de logs para o servidor Apache.
4. Escolha `Actions`, Criar filtro de métrica.
5. Em Filter pattern (Padrão de filtro), insira **`[ip, id, user, timestamp, request, status_code, size]`**.
6. Para testar seu padrão de filtro, escolha Testar padrão.
7. Escolha Next (Próximo) e, em Filter name (Nome do filtro), digite **`size`**.

8. Em Metric details (Detalhes da métrica), para Metric namespace (Namespace da métrica), insira **`MyNameSpace`**. Como esse é um novo namespace, verifique se Create new (Criar novo) está selecionado.
9. Em Metric name (Nome da métrica), insira **`BytesTransferred`**.
10. Em Metric value (Valor da métrica), insira **`$size`**.
11. Em Default value (Valor padrão), insira 0 e escolha Next (Próximo).
12. Escolha Criar filtro de métrica.

Para criar um filtro de métrica usando a AWS CLI

Em um prompt de comando, execute o seguinte comando

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
--filter-name BytesTransferred \
--filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
--metric-transformations \
metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue=$size,defaultValue=0
```

Você pode usar os seguintes dados em chamadas log-event para testar essa regra. Isso gerará duas métricas diferentes se você não removeu a regra de monitoramento no exemplo anterior.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Listagem de filtros de métrica

Você pode listar todos os filtros de métrica em um grupo de logs.

Para listar filtros de métrica usando o console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Escolha o número de filtros no painel de conteúdo, na lista de grupos de log, na coluna Metric Filters (Filtros de métrica).

   A tela Grupos de logs > Filtros para lista todos os filtros de métrica associados ao grupo de logs.

Para listar filtros de métrica usando a AWS CLI

Em um prompt de comando, execute o seguinte comando:

```
aws logs describe-metric-filters --log-group-name MyApp/access.log
```

A seguir está um exemplo de saída:

```
{
    "metricFilters": [
```

```
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404, size]"
        }
    ]
}
```

Exclusão de um filtro de métrica

Uma política é identificada por seu nome e o grupo de logs ao qual ela pertence.

Para excluir um filtro de métrica usando o console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Escolha o filtro de métrica no painel de conteúdo, na coluna Metric Filter (Filtro de métrica).
4. Na tela Filtros de métrica de logs, no filtro de métrica, escolha Excluir filtro.
5. Quando a confirmação for solicitada, escolha Yes, Delete (Sim, excluir).

Para excluir um filtro de métrica usando a AWS CLI

Em um prompt de comando, execute o seguinte comando:

```
aws logs delete-metric-filter --log-group-name MyApp/access.log \
 --filter-name MyFilterName
```

Processamento em tempo real de dados de log com assinaturas

Pode utilizar as subscrições para obter acesso a um feed em tempo real de eventos de registo de CloudWatch Logs e ter-se-ão entregue a outros serviços, tais como um Amazon Kinesis série, um Amazon Kinesis Data Firehose fluxo, ou AWS Lambda para processamento, análise ou carregamento personalizados para outros sistemas. Quando os eventos de registo são enviados para o serviço de receção, são codificados e comprimidos com o formato gzip.

Para começar a subscrever eventos de registo, crie o recurso de receção, tal como um Kinesis onde os eventos serão entregues. Um filtro de assinatura define o padrão de filtro a ser usado para filtragem de quais eventos de log devem ser entregues para o seu recurso da AWS, bem como informações sobre o local para onde enviar eventos de log correspondentes.

Cada grupo de registo pode ter até dois filtros de subscrição associados.

> Note
>
> Se o serviço de destino apresentar um erro de nova tentativa, como uma exceção de aceleração ou uma exceção de serviço de nova tentativa (HTTP 5xx, por exemplo), CloudWatch Logs continua a tentar a entrega até 24 horas. CloudWatch Logs (Entregar novamente) não tenta entregar novamente se o erro for um erro impossível de tentar novamente, tal como AccessDeniedException ou ResourceNotFoundException.

O CloudWatch Logs também produz métricas do CloudWatch sobre o encaminhamento de eventos de log para assinaturas. Para obter mais informações, consulte Métricas e dimensões do Amazon CloudWatch Logs.

Tópicos

Concepts

Cada filtro de assinatura é composto dos seguintes elementos-chave:

nome do grupo de logs

> O grupo de logs ao qual associar o filtro de assinatura. Todos os eventos de registo carregados para este grupo de registo estarão sujeitos ao filtro de subscrição e os que correspondem ao filtro são entregues no serviço de destino que está a receber os eventos de registo correspondentes.

padrão de filtro

> Descrição simbólica de como o CloudWatch Logs deve interpretar os dados em cada evento de log, juntamente com expressões de filtragem que restringem o que é entregue ao recurso de destino da AWS. Para obter mais informações sobre a sintaxe padrão de filtros, consulte Sintaxe do padrão e do filtro (p. 72).

arn de destino

> O nome de recurso da Amazon (ARN) do fluxo do Kinesis ou do Kinesis Data Firehose, ou da função do Lambda que você deseja usar como o destino de feed da assinatura.

arn de função

máquinas IAM função que concede CloudWatch Logs as permissões necessárias para colocar dados no destino escolhido. Essa função não é necessária para destinos do Lambda porque o CloudWatch Logs pode obter as permissões necessárias a partir das configurações de controle de acesso na própria função do Lambda.

distribuição

O método usado para distribuir dados de log no destino, quando o destino é um stream do Amazon Kinesis. Por padrão, os dados de log são agrupados por stream de log. Para obter uma distribuição uniforme, você pode agrupar os dados de log aleatoriamente.

Uso de filtros de assinatura do CloudWatch Logs

Você pode usar um filtro de assinatura com o Kinesis, o Lambda ou o Kinesis Data Firehose. Os registos enviados para um serviço de receção através de um filtro de subscrição são codificados com Base64 e comprimidos com o formato gzip.

Exemplos

- Exemplo 1: Filtros de subscrição com Kinesis (p. 88)
- Exemplo 2: Filtros de subscrição com AWS Lambda (p. 92)
- Exemplo 3: Filtros de subscrição com Amazon Kinesis Data Firehose (p. 94)

Exemplo 1: Filtros de subscrição com Kinesis

O exemplo a seguir associa um filtro de assinatura a um grupo de logs que contém eventos do AWS CloudTrail para que todas as atividades registradas feitas pelas credenciais "Root" da AWS sejam entregues a um fluxo do Kinesis denominado "RootAccess". Para obter mais informações sobre como enviar eventos do AWS CloudTrail ao CloudWatch Logs, consulte Envio de eventos do CloudTrail ao CloudWatch Logs no AWS CloudTrail User Guide.

Note

Antes de criar o fluxo do Kinesis, calcule o volume de dados de log que será gerado. Certifique-se de criar um fluxo do Kinesis com estilhaços suficientes para suportar esse volume. Se o fluxo não tiver um número suficiente de estilhaços, o fluxo de logs será limitado. Para obter mais informações sobre os limites de volume de fluxo do Kinesis, consulte Limites de fluxos de dados do Amazon Kinesis.

Para criar um filtro de assinatura para o Kinesis

1. Crie um stream do Kinesis de destino usando o seguinte comando:

```
$ C:\>  aws kinesis create-stream --stream-name "RootAccess" --shard-count 1
```

2. Aguarde até que o stream do Kinesis se torne ativo (isso pode levar um ou dois minutos). Pode usar o seguinte Kinesis descrever-stream para verificar a StreamDescription.Estado do Fluxo propriedade. Além disso, tenha em atenção a StreamDescription.APRENDIZAGEM em sequência uma vez que irá precisar dele numa etapa posterior:

```
aws kinesis describe-stream --stream-name "RootAccess"
```

Veja a seguir um exemplo de saída:

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "RootAccess",
        "StreamARN": "arn:aws:kinesis:us-east-1:123456789012:stream/RootAccess",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "340282366920938463463374607431768211455",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "49551135218688818456679503831981458784591352702181572610"
                }
            }
        ]
    }
}
```

3. Crie a função do IAM que concederá ao CloudWatch Logs permissão para colocar os dados em seu fluxo do Kinesis. Primeiro, você precisará criar uma política de confiança em um arquivo (por exemplo, `~/TrustPolicyForCWL.json`). Use um editor de texto para criar esta política. Não use o console do IAM para criá-la.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.region.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

4. Use o comando create-role para criar a função do IAM especificando o arquivo de política de confiança. Observe o valor de Role.Arn retornado, pois ele também será necessário em uma etapa posterior:

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document file://
~/TrustPolicyForCWL.json
```

Segue-se um exemplo da saída.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
```

```
    }
}
```

5. Crie uma política de permissões para definir quais ações o CloudWatch Logs pode fazer em sua conta. Primeiro, você criará uma política de permissões em um arquivo (por exemplo, `~/PermissionsForCWL.json`). Use um editor de texto para criar esta política. Não use o console do IAM para criá-la.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/RootAccess"
    }
  ]
}
```

6. Associe a política de permissões com a função usando o seguinte comando put-role-policy:

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-
Policy-For-CWL  --policy-document file://~/PermissionsForCWL.json
```

7. Quando o fluxo do Kinesis estiver no estado Active (Ativo) e você tiver criado a função do IAM, você poderá criar o filtro de assinatura do CloudWatch Logs. O filtro de assinatura inicia imediatamente o fluxo de dados de log em tempo real a partir do grupo de logs escolhido para o fluxo do Kinesis:

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "RootAccess" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:kinesis:region:123456789012:stream/RootAccess" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
```

8. Depois de configurar o filtro de assinatura, o CloudWatch Logs encaminhará todos os eventos de log de entrada que corresponderem ao padrão de filtro para o fluxo do Kinesis. Você pode verificar se isso está acontecendo obtendo um iterador de estilhaços do Kinesis e usando o comando get-records do Kinesis para buscar alguns registros do Kinesis:

```
aws kinesis get-shard-iterator --stream-name RootAccess --shard-id shardId-000000000000
 --shard-iterator-type TRIM_HORIZON
```

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiK2OSh0uP"
```

Observe que você pode precisar fazer esta chamada algumas vezes para que o Kinesis comece a retornar os dados.

É necessário esperar para ver uma resposta com um conjunto de registros. O atributo Data (Dados) em um registro do Kinesis é codificado em Base64 e compactado com o formato gzip. Você pode examinar os dados brutos na linha de comando usando os seguintes comandos Unix:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Os dados decodificados por Base64 e descompactados têm o formato JSON com a seguinte estrutura:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail",
    "logStream": "111111111111_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

Os principais elementos na estrutura de dados acima são os seguintes:

owner

O ID da conta da AWS dos dados de log de origem.

logGroup

O nome do grupo de logs dos dados de log de origem.

logStream

O nome do stream de log dos dados de log de origem.

subscriptionFilters

A lista de nomes de filtro de assinatura que corresponderam aos dados de log de origem.

messageType

As mensagens de dados usarão o tipo "DATA_MESSAGE". Às vezes, o CloudWatch Logs pode emitir registros do Kinesis com um tipo "CONTROL_MESSAGE", principalmente para verificar se o destino é acessível.

logEvents

Os dados de log reais, representados como um conjunto de registros de eventos de log. A propriedade "id" é um identificador exclusivo de cada evento de log.

Exemplo 2: Filtros de subscrição com AWS Lambda

Neste exemplo, você criará um filtro de assinatura do CloudWatch Logs que enviará dados para a sua função do AWS Lambda.

Note

Antes de criar a função do Lambda, calcule o volume de dados de log que será gerado. Lembre-se de criar uma função que suporte esse volume. Se a função não tiver um volume suficiente, o fluxo de logs será limitado. Para obter mais informações sobre os limites do Lambda, consulte Limites do AWS Lambda.

Para criar um filtro de assinatura para o Lambda

1. Criar a função do AWS Lambda

   Verifique se a função de execução do Lambda foi configurada. Para obter mais informações, consulte Passo 2.2: Criar uma Função IAM (função de execução) na AWS Lambda Developer Guide.
2. Abra um editor de texto e crie um arquivo chamado `helloWorld.js` com o seguinte conteúdo:

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString('ascii'));
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Fechar o ficheiro helloWorld.js e guarde-o com o nome `helloWorld.zip`.
4. Use o comando a seguir, no qual a função é a função de execução do Lambda configurada na primeira etapa:

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs12.x
```

5. Conceda ao CloudWatch Logs a permissão para executar sua função. Use o comando a seguir, substituindo o espaço reservado conta pela sua própria conta e o espaço reservado grupo de logs pelo grupo de logs a ser processado:

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.region.amazonaws.com" \
    --action "lambda:InvokeFunction" \
```

```
    --source-arn "arn:aws:logs:region:123456789123:log-group:TestLambda:*" \
    --source-account "123456789012"
```

6. Crie um filtro de assinatura usando o seguinte comando, substituindo o espaço reservado conta pela sua própria conta e o espaço reservado grupo de logs pelo grupo de logs a ser processado:

```
aws logs put-subscription-filter \
    --log-group-name myLogGroup \
    --filter-name demo \
    --filter-pattern "" \
    --destination-arn arn:aws:lambda:region:123456789123:function:helloworld
```

7. (Opcional) Teste usando um exemplo de evento de log. Em um prompt de comando, execute o seguinte comando, que colocará uma mensagem de log simples no stream assinado.

   Para ver o resultado de sua função do Lambda, navegue para a função do Lambda na qual você verá a saída em /aws/lambda/helloworld:

```
aws logs put-log-events --log-group-name myLogGroup --log-stream-name stream1 --log-
events "[{\"timestamp\":<CURRENT TIMESTAMP MILLIS> , \"message\": \"Simple Lambda
 Test\"}]"
```

   É necessário esperar para ver uma resposta com uma matriz do Lambda. O atributo Data (Dados) em um registro do Lambda. é codificado em Base64 e compactado com o formato gzip. A carga útil real recebida pelo Lambda está no seguinte formato `{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }`. Você pode examinar os dados brutos na linha de comando usando os seguintes comandos Unix:

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

   Os dados decodificados por Base64 e descompactados têm o formato JSON com a seguinte estrutura:

```
{
    "owner": "123456789012",
    "logGroup": "CloudTrail",
    "logStream": "123456789012_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
```

```
    ]
}
```

Os principais elementos na estrutura de dados acima são os seguintes:

owner

O ID da conta da AWS dos dados de log de origem.

logGroup

O nome do grupo de logs dos dados de log de origem.

logStream

O nome do stream de log dos dados de log de origem.

subscriptionFilters

A lista de nomes de filtro de assinatura que corresponderam aos dados de log de origem.

messageType

As mensagens de dados usarão o tipo "DATA_MESSAGE". Às vezes, o CloudWatch Logs pode emitir registros do Lambda com um tipo "CONTROL_MESSAGE", principalmente para verificar se o destino é acessível.

logEvents

Os dados de log reais, representados como um conjunto de registros de eventos de log. A propriedade "id" é um identificador exclusivo de cada evento de log.

Exemplo 3: Filtros de subscrição com Amazon Kinesis Data Firehose

Neste exemplo, você criará uma assinatura do CloudWatch Logs que enviará todos os eventos de log de entrada correspondentes aos filtros definidos no fluxo de entrega do Amazon Kinesis Data Firehose. Os dados enviados do CloudWatch Logs para o Amazon Kinesis Data Firehose já estão compactados com a compactação gzip de nível 6, de modo que você não precisa usar o fluxo de entrega do Kinesis Data Firehose.

Note

Antes de criar o fluxo do Kinesis Data Firehose, calcule o volume de dados de log que será gerado. Crie um fluxo do Kinesis Data Firehose que possa comportar esse volume. Se o fluxo não puder suportar o volume, o fluxo de logs será limitado. Para obter mais informações sobre os limites de volume de fluxo do Kinesis Data Firehose, consulte Limites de dados do Amazon Kinesis Data Firehose.

Para criar um filtro de assinatura para o Kinesis Data Firehose

1. Crie um bucket do Amazon Simple Storage Service (Amazon S3). Recomendamos que você use um bucket criado especificamente para o CloudWatch Logs. No entanto, se você desejar usar um bucket existente, vá para a etapa 2.

   Execute o comando a seguir, substituindo o espaço reservado região pela região que você deseja usar:

```
aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
 LocationConstraint=region
```

Veja a seguir um exemplo de saída:

```
{
    "Location": "/my-bucket"
}
```

2. Crie a função do IAM que concederá ao Amazon Kinesis Data Firehose permissão para colocar os dados em seu bucket do Amazon S3.

   Para obter mais informações, consulte Controlar acesso com o Amazon Kinesis Data Firehose no Guia do desenvolvedor do Amazon Kinesis Data Firehose.

   Primeiro, use um editor para criar uma política de confiança em um arquivo `~/TrustPolicyForFirehose.json` como a seguir, substituindo account-id pelo ID de sua conta da AWS:

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

3. Use o comando create-role para criar a função do IAM especificando o arquivo de política de confiança. Observe o valor de Role.Arn retornado, pois você precisará dele em uma etapa posterior:

```
aws iam create-role \
      --role-name FirehosetoS3Role \
      --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Crie uma política de permissões para definir quais ações o Kinesis Data Firehose pode fazer em sua conta. Primeiro, use um editor de texto para criar uma política de permissões em um arquivo `~/PermissionsForFirehose.json`:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
```

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

5. Associe a política de permissões com a função usando o seguinte comando put-role-policy:

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-Policy-
For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Criar um destino Kinesis Data Firehose saída de fornecimento da seguinte forma, substituindo os valores do marcador de posição para RoleARN e BucketARN com a função e o desejo ARNs que criou:

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::my-bucket"}'
```

Observe que o Kinesis Data Firehose usa automaticamente um prefixo no formato de tempo AAAA/MM/DD/HH (UTC) para objetos entregues do Amazon S3. Você pode especificar um prefixo extra a ser incluído na frente do prefixo de formato de tempo. Se o prefixo terminar com uma barra (/), ele aparecerá como uma pasta no bucket do Amazon S3.

7. Aguarde até que o stream fique ativo (isso pode levar alguns minutos). Você pode usar o Kinesis Data Firehose descrever-delivery-stream para verificar a DeliveryStreamDescription.EstadoEsquemaDeEntrega propriedade. Além disso, tenha em atenção a DeliveryStreamDescription.APRENDIZAGEM da Entrega uma vez que irá precisar dele numa etapa posterior:

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-east-1:123456789012:deliverystream/
my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::my-bucket",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
```

```
                    }
                }
            }
        ]
    }
}
```

8. Crie a função do IAM que concederá ao CloudWatch Logs permissão para colocar dados no seu fluxo de entrega do Kinesis Data Firehose. Primeiro, use um editor de texto para criar uma política de confiança em um arquivo `~/TrustPolicyForCWL.json`:

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.region.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

9. Use o comando create-role para criar a função do IAM especificando o arquivo de política de confiança. Observe o valor de Role.Arn retornado, pois você precisará dele em uma etapa posterior:

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. Crie uma política de permissões para definir quais ações o CloudWatch Logs pode fazer em sua conta. Primeiro, use um editor de texto para criar um arquivo de política de permissões (por exemplo, `~/PermissionsForCWL.json`):

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:123456789012:*"]
      }
    ]
}
```

11. Associe a política de permissões com a função usando o comando put-role-policy:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name Permissions-
Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Quando o fluxo de entrega do Amazon Kinesis Data Firehose estiver no estado ativo e você tiver criado a função do IAM, você poderá criar o filtro de assinatura do CloudWatch Logs. O filtro de assinatura inicia imediatamente o fluxo de dados de log em tempo real a partir do grupo de logs escolhido para o fluxo do entrega do Amazon Kinesis Data Firehose:

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "Destination" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:firehose:region:123456789012:deliverystream/my-delivery-
stream" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
```

13. Depois de configurar o filtro de assinatura, o CloudWatch Logs encaminhará todos os eventos de log de entrada que corresponderem ao padrão de filtro para o fluxo de entrega do Amazon Kinesis Data Firehose. Seus dados começarão a aparecer no Amazon S3 com base no intervalo de buffer de tempo definido em seu fluxo de entrega do Amazon Kinesis Data Firehose. Quando tiver passado tempo suficiente, você poderá conferir seus dados verificando o bucket do Amazon S3.

```
aws s3api list-objects --bucket 'my-bucket' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2015-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-
a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-
stream-2015-10-29-00-35-40-7cc92023-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'my-bucket' --key 'firehose/2015/10/29/00/my-delivery-
stream-2015-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250' testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2015 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
```

```
}
```

Os dados no objeto do Amazon S3 são compactados com o formato gzip. Você pode examinar os dados brutos na linha de comando usando o seguinte comando Unix:

```
zcat testfile.gz
```

Compartilhamento de dados de log entre contas com assinaturas

Você pode colaborar com um proprietário de outra conta da AWS e receber seus eventos de log nos recursos da AWS, como um fluxo do Amazon Kinesis (isso é conhecido como compartilhamento de dados entre contas). Por exemplo, esses dados de eventos de log podem ser lidos em um fluxo do Amazon Kinesis centralizado para realizar processamento personalizado e análise. O processamento personalizado é especialmente útil quando você colabora e analisa dados entre várias contas. Por exemplo, um grupo de segurança de informações de uma empresa pode querer analisar dados de detecção de invasões em tempo real ou comportamentos anormais para que possa realizar uma auditoria de contas em todas as divisões da empresa, coletando seus logs de produção federados para processamento central. Um fluxo em tempo real de dados de eventos entre essas contas pode ser montado e entregue aos grupos de segurança de informações que podem usar o Kinesis para anexar os dados aos seus sistemas analíticos de segurança existentes.

Os fluxos do Kinesis são atualmente o único recurso com suporte como destino para assinaturas entre contas.

Para compartilhar dados de log entre contas, você precisa estabelecer um remetente e um destinatário dos dados de log:

- Remetente dos dados de log — obtém as informações de destino do destinatário e permite que o CloudWatch Logs saiba que está pronto para enviar seus eventos de log para o destino especificado. Nos procedimentos apresentados no restante desta seção, o remetente dos dados de log é mostrado com um número de uma conta fictícia da AWS 111111111111.
- Log data recipient (Destinatário dos dados de log) — configura um destino que encapsula um fluxo do Kinesis e permite que o CloudWatch Logs saiba que o destinatário deseja receber dados de log. O destinatário, então, compartilha as informações sobre seu destino com o remetente. Nos procedimentos apresentados no restante desta seção, o destinatário dos dados de log é mostrado com um número de uma conta fictícia da AWS 999999999999.

Para começar a receber eventos de log entre usuários de contas, o destinatário dos dados de log primeiro cria um destino do CloudWatch Logs. Cada destino consiste nos seguintes elementos-chave:

Nome do destino

O nome do destino que você deseja criar.

ARN de destino

O Nome de recurso da Amazon (ARN) do recurso da AWS que você deseja usar como o destino do feed de assinatura.

ARN de função

Uma função do AWS Identity and Access Management (IAM) que concede ao CloudWatch Logs as permissões necessárias para colocar dados em um fluxo escolhido do Kinesis.

Política de acesso

Um documento de política do IAM (no formato JSON, gravado usando a gramática de políticas do IAM) que controla o conjunto de usuários que têm permissão para gravar em seu destino.

O grupo de logs e o destino devem estar na mesma região da AWS. No entanto, o recurso da AWS para o qual o destino aponta pode estar localizado em uma região diferente.

Tópicos

Criar um destino

Important

As etapas deste procedimento devem ser processadas na conta destinatária dos dados do log.

Para este exemplo, a conta destinatária dos dados de log tem um ID de conta da AWS de 999999999999. Já o ID de conta da AWS do remetente de dados de log é 111111111111.

Este exemplo cria um destino utilizando um Kinesis sequência chamada RecipientStreame uma função que permite CloudWatch Logs para escrever dados no mesmo.

Para criar um destino

1. Crie um fluxo de destino no Kinesis. Em um prompt de comando, digite:

```
aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
```

2. Aguarde até que o fluxo do Kinesis fique ativo. Você pode usar o A cinese descreve o fluxo para verificar a StreamDescription.Estado do Fluxo propriedade. Além disso, tome nota da StreamDescription.APRENDIZAGEM em sequência porque será passado para CloudWatch Logs mais tarde:

```
aws kinesis describe-stream --stream-name "RecipientStream"
{
  "StreamDescription": {
    "StreamStatus": "ACTIVE",
    "StreamName": "RecipientStream",
    "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
    "Shards": [
      {
        "ShardId": "shardId-000000000000",
        "HashKeyRange": {
          "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702l8EXAMPLE"
        }
      }
    ]
  }
```

```
}
```

Pode levar um ou dois minutos para o seu stream aparecer no estado ativo.

3. Crie a função do IAM que concederá ao CloudWatch Logs a permissão para colocar os dados em seu fluxo do Kinesis. Primeiro, você precisará criar uma política de confiança em um arquivo ~/TrustPolicyForCWL.json. Use um editor de texto para criar esse arquivo de política, não use o console do IAM.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.region.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

4. Use o comando aws iam create-role para criar a função do IAM especificando o arquivo de política de confiança. Anote o valor Role.Arn retornado porque ele também será transmitido para o CloudWatch Logs posteriormente:

```
aws iam create-role \
    --role-name CWLtoKinesisRole \
    --assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Crie uma política de permissões para definir quais ações o CloudWatch Logs pode executar em sua conta. Primeiro, você usará um editor de texto para criar uma política de permissões em um arquivo ~/PermissionsForCWL.json:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Associe a política de permissões à função usando o comando aws iam put-role-policy:

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
```

```
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. Quando o fluxo do Kinesis estiver no estado ativo e você tiver criado a função do IAM, você poderá criar o destino do CloudWatch Logs.

   a. Esta etapa não associará uma política de acesso ao seu destino e só é a primeira etapa das duas concluirá uma criação de destino. Anote o DestinationArn que for retornado na carga útil:

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

   b. Depois que a etapa 7a for concluída, na conta destinatária dos dados de log, associe uma política de acesso ao destino. Esta política permite que a conta remetente dos dados de log (111111111111 conta) acesse o destino na conta destinatária dos dados de log (999999999999). Você pode usar um editor de texto para colocar essa política no arquivo ~/AccessPolicy.json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

      Note

      Se várias contas estiverem enviando logs para esse destino, cada conta de remetente deverá ser listada separadamente na política. Esta política não oferece suporte à especificação de `*` como `Principal` ou ao uso da chave global `aws:PrincipalOrgId`.

   c. Isso cria uma política que define quem tem acesso de gravação ao destino. Essa política deve especificar a ação logs:PutSubscriptionFilter para acessar o destino. Os usuários entre contas usará a ação PutSubscriptionFilter para enviar eventos de log ao destino:

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

      Esta política de acesso permite aos utilizadores na Conta AWS com ID 111111111111 ligarem para PutSubscriptionFilter para o destino com ARN:aws:logs:*region*:999999999999:destino:testeDestino. Qualquer tentativa de chamada de outro utilizador PutSubscriptionFilter neste destino serão rejeitados.

Para validar os privilégios de um usuário com base em uma política de acesso, consulte Uso do validador de políticas no Guia do usuário do IAM.

Criar um filtro de assinatura

Depois de criar um destino, a conta destinatária dos dados de log pode compartilhar o ARN do destino (arn:aws:logs:us-east-1:999999999999:destination:testDestination) com outras contas da AWS para que elas possam enviar seus eventos de log para o mesmo destino. Esses outros usuários de contas de envio criam um filtro de assinatura em seus respectivos grupos de log para esse destino. O filtro de assinatura filtrar inicia imediatamente o fluxo de dados de log em tempo real a partir do grupo de logs escolhido para o destino especificado.

No exemplo a seguir, um filtro de assinatura é criado em uma conta remetente. O filtro está associado a um grupo de logs que contém eventos do AWS CloudTrail de modo que todas as atividades registradas feitas pelas credenciais "Root" da AWS sejam entregues ao destino que você criou acima. Esse destino encapsula um stream do Kinesis chamado "RecipientStream". Para obter mais informações sobre como enviar eventos do AWS CloudTrail ao CloudWatch Logs, consulte Envio de eventos do CloudTrail ao CloudWatch Logs no AWS CloudTrail User Guide.

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "RecipientStream" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

O grupo de logs e o destino devem estar na mesma região da AWS. No entanto, o destino pode apontar para um recurso da AWS, como um fluxo do Kinesis, que está localizado em uma região diferente.

Validação do fluxo de eventos de log

Depois de criar o filtro de assinatura, o CloudWatch Logs encaminha todos os eventos de log de entrada correspondentes ao padrão de filtro para o stream do Kinesis que é encapsulado no stream de destino denominado "RecipientStream". O proprietário do destino pode verificar se isso está acontecendo usando o comando aws kinesis get-shard-iterator para capturar um estilhaço do Kinesis e o comando aws kinesis get-records para buscar alguns registros do Kinesis:

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

Note

Pode ser necessário executar o comando get-records algumas vezes para que o Kinesis comece a retornar dados.

Você deve ver uma resposta com um conjunto de registros do Kinesis. O atributo de dados no registro do Kinesis é compactado no formato gzip e, depois, codificado em Base64. Você pode examinar os dados brutos na linha de comando usando o seguinte comando Unix:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Os dados decodificados por Base64 e descompactados têm o formato JSON com a seguinte estrutura:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail",
    "logStream": "111111111111_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}"
        }
    ]
}
```

Os principais elementos nesta estrutura de dados são os seguintes:

owner

O ID da conta da AWS dos dados de log de origem.

logGroup

O nome do grupo de logs dos dados de log de origem.

logStream

O nome do stream de log dos dados de log de origem.

subscriptionFilters

A lista de nomes de filtro de assinatura que corresponderam aos dados de log de origem.

messageType

As mensagens de dados usarão o tipo "DATA_MESSAGE". Às vezes, o CloudWatch Logs pode emitir registros do Kinesis com um tipo "CONTROL_MESSAGE", principalmente para verificar se o destino é acessível.

logEvents

Os dados de log reais, representados como um conjunto de registros de eventos de log. A propriedade ID é um identificador exclusivo de cada evento de log.

Modificação da associação de destino no tempo de execução

Pode haver situações em que você precise adicionar ou remover a associação de alguns usuários de um destino que você possua. Você pode usar o PutDestinationPolicy no seu destino com a nova política de acesso. No exemplo a seguir, uma conta 111111111111 recém-adicionada é impedida de enviar mais dados de log e a conta 222222222222 é ativada.

1. Busque a política que está atualmente associada ao destino testDestination e anote a AccessPolicy:

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisRole",
     "DestinationArn": "arn:aws:logs:region:222222222222:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:222222222222:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:123456789012:destination:testDestination\"}] }"
   }
 ]
}
```

2. Atualize a política para refletir que a conta 111111111111 foi interrompida e a conta 222222222222 está habilitada. Coloque esta política no arquivo ~/NewAccessPolicy.json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:222222222222:destination:testDestination"
    }
  ]
}
```

3. Chamada PutDestinationPolicy para associar a política definida na NewAccessPolicy.json ficheiro com o destino:

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

Por fim, isso desativará os eventos de log do ID da conta 111111111111. Registar eventos a partir da ID de conta 222222222222 começam a fluir para o destino assim que o proprietário da conta 222222222222 cria um filtro de subscrição utilizando PutSubscriptionFilter.

Habilitar o registro em log de determinados serviços da AWS

Alguns serviços da AWS usam uma infraestrutura comum para enviar seus logs aos seguintes destinos:

- Amazon CloudWatch Logs
- Amazon Simple Storage Service
- Amazon Kinesis Data Firehose

Para permitir que os serviços da AWS listados na tabela a seguir enviem seus logs para esses destinos, você deve ter determinadas permissões.

Além disso, determinadas permissões devem ser concedidas ao AWS para habilitar o envio dos logs. O AWS pode criar essas permissões automaticamente quando os logs são configurados ou você mesmo pode criá-los antes de configurar o registro.

Se você optar por configurar automaticamente as permissões e as políticas de recursos necessárias do AWS quando você ou alguém em sua organização configurar o envio de logs, o usuário que estiver configurando o envio de logs deverá ter determinadas permissões, conforme explicado posteriormente nesta seção. Como alternativa, você mesmo pode criar as políticas de recursos e os usuários que configuram o envio de logs não precisam de tantas permissões.

A tabela a seguir resume a quais tipos de logs e destinos de log as informações desta seção se aplicam.

Tipo de log	CloudWatch Logs (p. 108)	Amazon S3 (p. 109)	Kinesis Data Firehose (p. 110)
Amazon API Gateway logs de acesso	✓		
Amazon Chime logs de métrica de qualidade de mídia e logs de mensagens SIP	✓		
CloudFront: logs de acesso		✓	
AWS Global Accelerator Logs de fluxo da		✓	
Amazon MSK Logs de agente do	✓	✓	✓
Logs de firewall de rede da AWS	✓	✓	✓
Load balancer de rede logs de acesso		✓	
Amazon Route 53 Logs de consulta do resolvedor do	✓	✓	✓
Amazon SageMaker Eventos de operador do	✓		
Arquivos de feed de dados da instância spot do EC2		✓	
AWS Step Functions Histórico do fluxo de trabalho do	✓		

Tipo de log	CloudWatch Logs (p. 108)	Amazon S3 (p. 109)	Kinesis Data Firehose (p. 110)
AWS Storage Gateway logs de auditoria e logs de integridade	✓		
Amazon Virtual Private Cloud Logs de fluxo da		✓	

As seções a seguir fornecem mais detalhes para cada um desses destinos.

Logs enviados para oCloudWatch Logs

> **Important**
>
> Quando você configura os tipos de log na lista a seguir para serem enviados ao CloudWatch Logs, o AWS cria ou altera as políticas de recursos associadas ao grupo de logs que recebe os logs, se necessário. Continue lendo esta seção para ver os detalhes.

Esta seção se aplica quando os seguintes tipos de logs são enviados ao CloudWatch Logs:

- Amazon API GatewayLogs de acesso do
- AWS Storage Gateway logs de auditoria e logs de integridade
- Amazon Chime logs de métrica de qualidade de mídia e logs de mensagens SIP
- Logs do agente doAmazon Managed Streaming for Apache Kafka
- AWS Logs de firewall de rede
- Amazon Route 53 logs de consulta do resolvedor
- Eventos do operador doAmazon SageMaker
- Histórico do fluxo de trabalho expresso do AWS Step Functions e histórico do fluxo de trabalho padrão

Permissões de usuário

Para poder configurar o envio de qualquer um desses tipos de logs para o CloudWatch Logs pela primeira vez, você deve estar conectado a uma conta com as seguintes permissões.

- `logs:CreateLogDelivery`
- `logs:PutResourcePolicy`
- `logs:DescribeResourcePolicies`
- `logs:DescribeLogGroups`

Se qualquer um desses tipos de logs já estiver sendo enviado a um grupo de logs no CloudWatch Logs, para configurar o envio de outro desses tipos de logs para esse mesmo grupo de logs, você só precisará da permissão `logs:CreateLogDelivery`.

Política de recursos do grupo de logs

O grupo de logs ao qual os logs estão sendo enviados deve ter uma política de recursos que inclua determinadas permissões. Se o grupo de logs atualmente não tiver uma política de recurso e o usuário que configura o registro em log tiver as permissões `logs:PutResourcePolicy`, `logs:DescribeResourcePolicies` e `logs:DescribeLogGroups` para o grupo de logs, o AWS criará automaticamente a seguinte política para ele quando você começar a enviar os logs para o CloudWatch Logs.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSLogDeliveryWrite20150319",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "delivery.logs.amazonaws.com"
        ]
      },
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:0123456789:log-group:my-log-group:log-stream:*"
      ]
    }
  ]
}
```

Se o grupo de logs tiver uma política de recursos, mas essa política não contiver a instrução mostrada na política anterior e o usuário que estiver configurando o registro em log tiver as permissões `logs:PutResourcePolicy`, `logs:DescribeResourcePolicies` e `logs:DescribeLogGroups` para o grupo de logs, essa instrução será anexada à política de recursos do grupo de logs.

Considerações sobre o limite de tamanho da política de recursos do grupo de logs

Esses serviços devem listar cada grupo de logs para o qual estão enviando logs na política de recursos, e as políticas de recursos do CloudWatch Logs estão limitadas a 5.120 caracteres. Um serviço que envia logs para um grande número de grupos de logs pode se deparar com esse limite.

Para mitigar isso, o CloudWatch Logs monitora o tamanho das políticas de recursos usadas pelo serviço que está enviando logs e, quando detecta que uma política se aproxima do limite de tamanho de 5.120 caracteres, o CloudWatch Logs habilita automaticamente `/aws/vendedlogs/*` na política de recursos desse serviço. É possível começar a usar grupos de logs com nomes que começam com `/aws/vendedlogs/` como os destinos dos logs desses serviços.

Logs enviados para oAmazon S3

> **Important**
>
> Quando você configura os tipos de log na lista a seguir para serem enviados ao Amazon S3, o AWS cria ou altera as políticas de recursos associadas ao bucket do S3 que está recebendo os logs, se necessário. Continue lendo esta seção para ver os detalhes.

Esta seção se aplica quando os seguintes tipos de logs são enviados ao Amazon S3:

- CloudFront logs de acesso e logs de acesso de streaming. O CloudFront usa um modelo de permissões diferente dos outros serviços nessa lista. Para obter mais informações, consulte Permissões necessárias para configurar o registro em log padrão e acessar seus arquivos de log.
- Feed de dados da instância spot doAmazon EC2
- AWS Global AcceleratorLogs de fluxo do
- Logs do agente doAmazon Managed Streaming for Apache Kafka
- Load balancer de redeLogs de acesso do
- AWS Logs de firewall de rede
- Amazon Virtual Private CloudLogs de fluxo do

Permissões de usuário

Para poder configurar o envio de qualquer um desses tipos de logs para o Amazon S3 pela primeira vez, você deve estar conectado a uma conta com as seguintes permissões.

- `logs:CreateLogDelivery`
- `S3:GetBucketPolicy`
- `S3:PutBucketPolicy`

Se qualquer um desses tipos de logs já estiver sendo enviado a um bucket do Amazon S3, para configurar o envio de outro desses tipos de logs para o mesmo bucket, você só precisará ter a permissão `logs:CreateLogDelivery`.

Política de recursos de bucket do S3

O bucket do S3 ao qual os logs estão sendo enviados deve ter uma política de recursos que inclua determinadas permissões. Se, no momento, o bucket não tiver uma política de recursos e o usuário que estiver configurando o registro em log tiver as permissões `S3:GetBucketPolicy` e `S3:PutBucketPolicy` para o bucket, o AWS criará automaticamente a seguinte política para ele quando você começar a enviar os logs para o Amazon S3.

```
{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite20150319",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::my-bucket"
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-bucket/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Se o bucket tiver uma política de recursos, mas essa política não contiver a instrução mostrada na política anterior e o usuário que configurou o registro em log tiver as permissões `S3:GetBucketPolicy` e `S3:PutBucketPolicy` para o bucket, essa instrução será anexada à política de recursos do bucket.

Logs enviados para oKinesis Data Firehose

Esta seção se aplica quando os seguintes tipos de logs são enviados ao Kinesis Data Firehose:

- Logs do agente doAmazon Managed Streaming for Apache Kafka
- AWS Logs de firewall de rede
- Amazon Route 53 logs de consulta do resolvedor

Permissões de usuário

Para poder configurar o envio de qualquer um desses tipos de logs para o Kinesis Data Firehose pela primeira vez, você deve estar conectado a uma conta com as seguintes permissões.

- `logs:CreateLogDelivery`
- `firehose:TagDeliveryStream`
- `iam:CreateServiceLinkedRole`

Se qualquer um desses tipos de logs já estiver sendo enviado ao Kinesis Data Firehose, para configurar o envio de outro desses tipos de logs para o Kinesis Data Firehose, você precisará ter apenas as permissões `logs:CreateLogDelivery` e `firehose:TagDeliveryStream`.

IAM Funções do usadas para permissões

Como o Kinesis Data Firehose não usa políticas de recursos, o AWS usa funções do IAM ao configurar esses logs para serem enviados ao Kinesis Data Firehose. O AWS cria uma função vinculada ao serviço chamada AWSServiceRoleForLogDelivery. Essa função vinculada ao serviço inclui as permissões a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch",
                "firehose:ListTagsForDeliveryStream"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "firehose:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

Essa função vinculada ao serviço concede permissão a todos os fluxos de entrega do Kinesis Data Firehose que têm a tag `LogDeliveryEnabled` definida como `true`. O AWS dá essa tag ao fluxo de entrega de destino quando você configura o registro em log.

Essa função vinculada ao serviço também tem uma política de confiança que permite que o principal de serviço do `delivery.logs.amazonaws.com` assuma a função vinculada ao serviço necessária. Essa política de confiança é a seguinte:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
```

```
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Enviar registos diretamente para Amazon S3 ou Kinesis Data Firehose

Alguns AWS os serviços podem publicar registos diretamente para Amazon S3 ou Kinesis Data Firehose. Desta forma, se o seu principal requisito para os registos for o armazenamento ou processamento num destes serviços, pode facilmente ter o serviço que produz os registos, envie-os directamente para Amazon S3 ou Kinesis Data Firehose sem configurar infra-estrutura adicional.

Os logs do Amazon S3 são publicados em um bucket especificado por você. Um ou mais arquivos de log são criados a cada cinco minutos no bucket especificado.

Mesmo quando os registos são publicados diretamente para Amazon S3 ou Kinesis Data Firehose, CloudWatch Logs aplicam-se taxas. Para mais informações, consulte Registos vencidos no Registos separador Amazon CloudWatch Preços.

Os logs a seguir podem ser publicados diretamente no Amazon S3:

- VPC Flow Logs Para obter mais informações, consulte Publicação de logs de fluxo no Amazon S3 no Guia do usuário da Amazon VPC.
- Logs de fluxo do AWS Global Accelerator. Para obter mais informações, consulte Publicar logs de fluxo no Amazon S3 no Guia do desenvolvedor do AWS Global Accelerator.

Os logs a seguir podem ser publicados diretamente no Kinesis Data Firehose:

- Logs do Amazon Managed Streaming for Apache Kafka Para mais informações, consulte Iniciar sessão no Guia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

Exportação de dados de log para o Amazon S3

Você pode exportar dados de log dos grupos de log para um bucket do Amazon S3 e usar esses dados em processamento e análise personalizados ou para carregar em outros sistemas.

Exportar dados de registo para Amazon S3 pães que estão encriptados por AWS KMS não é suportado.

Para iniciar o processo de exportação, você deve criar um bucket do S3 para armazenar os dados de log exportados. Você pode armazenar os arquivos exportados em seu bucket do Amazon S3 e definir regras de ciclo de vida do Amazon S3 para arquivar ou excluir arquivos exportados automaticamente.

A exportação de buckets do S3 que são criptografados com AES-256 é compatível. A exportação de buckets do S3 que são criptografados com SSE-KMS não é compatível. Para obter mais informações, consulte Como habilitar a criptografia padrão em um bucket do S3?

Você pode exportar logs de vários grupos de log ou vários intervalos de tempo para o mesmo bucket do S3. Para separar dados de log para cada tarefa de exportação, você pode especificar um prefixo que será usado como o prefixo de chaves do Amazon S3 para todos os objetos exportados.

Pode levar até 12 horas para os dados de log se tornarem disponíveis para exportação. Para obter informações sobre análise quase em tempo real de dados de log, consulte Analisar dados de log com o CloudWatch Logs Insights (p. 35) ou Processamento em tempo real de dados de log com assinaturas (p. 87).

> **Note**
>
> A partir de 15 de fevereiro de 2019, a exportação para o recurso da Amazon S3 exige que os chamadores tenham acesso `s3:PutObject` ao bucket de destino.

Tópicos

Concepts

Antes de começar, familiarize-se com os seguintes conceitos de exportação:

nome do grupo de logs

: O nome do grupo de logs associado a uma tarefa de exportação. Os dados de log neste grupo serão exportados para o bucket especificado do Amazon S3.

de (timestamp)

: Um timestamp obrigatório expresso como o número de milissegundos desde 1º de janeiro de 1970 00:00:00 UTC. Todos os eventos de log no grupo de logs que foram ingeridos após esse período serão exportados.

a (timestamp)

: Um timestamp obrigatório expresso como o número de milissegundos desde 1º de janeiro de 1970 00:00:00 UTC. Todos os eventos de log no grupo de logs que foram ingeridos antes desse período serão exportados.

bucket de destino

O nome do bucket do Amazon S3 associado a uma tarefa de exportação. Esse bucket é usado para exportar os dados de log do grupo de logs especificado.

prefixo de destino

Um atributo opcional que é usado como o prefixo de chave do S3 para todos os objetos exportados. Isso ajuda a criar uma organização do tipo pasta em seu bucket.

Exportação de dados de log usando o console do Amazon S3

No exemplo a seguir, você usará o console do Amazon CloudWatch para exportar todos os dados de um grupo de logs do Amazon CloudWatch Logs chamado `my-log-group` para um bucket do Amazon S3 chamado `my-exported-logs`.

Exportar dados de registo para Amazon S3 pães que estão encriptados por AWS KMS não é suportado.

Etapa 1 Criar um bucket do Amazon S3

Recomendamos que você use um bucket criado especificamente para o CloudWatch Logs. No entanto, se você desejar usar um bucket existente, vá para a etapa 2.

Note

O bucket do Amazon S3 deve residir na mesma região que os dados de log a serem exportados. O CloudWatch Logs não oferece suporte à exportação de dados para buckets do Amazon S3 em uma região diferente.

Para criar um bucket do Amazon S3

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Se necessário, altere a região. Na barra de navegação, selecione a região em que seu CloudWatch Logs reside.
3. Escolha Create Bucket (Criar bucket).
4. Para Bucket Name (Nome do bucket), digite um nome para o bucket.
5. Em Region (Região), selecione a região onde os dados do CloudWatch Logs residem.
6. Selecione Create (Criar).

Etapa 2. Criar um IAM Utilizador com acesso total a Amazon S3 e CloudWatch Logs

Nas etapas a seguir, você criará o usuário do IAM com permissões necessárias.

Para criar o usuário necessário do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. Selecione Users (Usuários), Add user (Adicionar usuário).
3. Introduza um nome de utilizador, como *`CWLExportUser`*.
4. Selecione Programmatic access (Acesso programático) e AWS Management Console access (Acesso ao Console de Gerenciamento da AWS).

5. Escolha Autogenerated password (Senha geada automaticamente) ou Custom password (Senha personalizada).
6. Selecione Next (Próximo). Permissões
7. Escolha Attach existing policies directly (Associar políticas existentes diretamente) e associe as políticas `AmazonS3FullAccess` e `CloudWatchLogsFullAccess` ao usuário. É possível usar a caixa de pesquisa para localizar as políticas.
8. Selecione Next (Próximo). Etiquetas, Seguinte: Revisão, e depois Criar utilizador.

Etapa 3 Definir permissões num Amazon S3 Balde

Por padrão, todos os objetos e buckets do Amazon S3 são privados. Somente o proprietário do recurso e a conta da AWS que criou o bucket podem acessar o bucket e todos os objetos que ele contém. No entanto, o proprietário do recurso pode optar por conceder permissões de acesso a outros recursos e usuários ao criar uma política de acesso padrão.

Quando você define a política, é recomendável incluir uma string gerada aleatoriamente como o prefixo para o bucket, para que apenas os streams de log desejados sejam exportados para o bucket.

Para definir permissões em um bucket do Amazon S3

1. No console do Amazon S3, escolha o bucket que você criou na etapa 1.
2. Escolha Permissions (Permissões), Bucket policy (Política de bucket).
3. No Bucket Policy Editor (Editor de políticas de bucket), adicione uma das políticas a seguir. Altere `my-exported-logs` para o nome de seu bucket do S3 e `random-string` para uma string de caracteres gerada aleatoriamente. Certifique-se de especificar corretamente o endpoint de região para Principal.

   - Se o bucket estiver em sua conta, adicione a seguinte política.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      }
    ]
}
```

   - Se o bucket estiver em outra conta, use a seguinte política. Ela inclui uma instrução adicional utilizando o usuário do IAM que você criou na etapa anterior.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
```

```
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
            "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
            "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
            "Principal": { "AWS": "arn:aws:iam::SendingAccountID:user/CWLExportUser" }
        }
    ]
}
```

4. Escolha Salvar para definir a política que você acabou de adicionar como política de acesso em seu bucket. Essa política permite que o CloudWatch Logs exporte dados de log para o seu bucket do Amazon S3. O proprietário do bucket tem permissões completas sobre todos os objetos exportados.

   Warning

   Se o bucket existente já tiver uma ou mais políticas anexadas, adicione as declarações de acesso do CloudWatch Logs a essa política ou a essas políticas. Recomendamos avaliar o conjunto resultante de permissões para ter certeza de que elas são apropriadas para os usuários que acessarão o bucket.

Etapa 4. Criar uma tarefa de exportação

Nesta etapa, você criará a tarefa de exportação para exportar os logs de um grupo de logs.

Para exportar dados para o Amazon S3 usando o console do CloudWatch

1. Inicie sessão como IAM utilizador que criou em Passo 2: Criar um IAM Utilizador com acesso total a Amazon S3 e CloudWatch Logs.
2. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação, escolha Log groups (Grupos de logs).
4. Na tela Grupos de logs, escolha o nome do grupo de logs.
5. Escolha Ações, Exportar dados para o Amazon S3.
6. Na tela Exportar dados para o Amazon S3, em Definir exportação de dados, defina o período dos dados a serem exportados usando De e Até.
7. Se o seu grupo de logs tiver vários streams de log, você poderá fornecer um prefixo de stream de logs para limitar os dados do grupo de logs para um stream específico. Escolha Advanced (Avançado) e, depois, em Stream prefix (Prefixo do stream), digite o prefixo do stream de logs.
8. Em Choose S3 bucket (Escolher bucket do S3), escolha a conta associada ao bucket do Amazon S3.
9. Em S3 bucket name (Nome do bucket do S3), escolha um bucket do Amazon S3.
10. Em Prefixo do bucket do S3, insira a string gerada aleatoriamente que você especificou na política do bucket.
11. Escolha Exportar para exportar seus dados de log para o Amazon S3.
12. Para visualizar o status dos dados de log exportados para o Amazon S3, escolha Actions (Ações) e, depois, View all exports to Amazon S3 (Exibir todas as exportações para o Amazon S3).

Exportação de dados de log para o Amazon S3 com a AWS CLI

No exemplo seguinte, utiliza uma tarefa de exportação para exportar todos os dados de um CloudWatch Logs grupo de registo nomeado `my-log-group` para um Amazon S3 balde chamado `my-exported-logs`. Este exemplo pressupõe que já criou um grupo de registo chamado `my-log-group`.

Exportar dados de registo para Amazon S3 pães que estão encriptados por AWS KMS não é suportado.

Etapa 1 Criar um bucket do Amazon S3

Recomendamos que você use um bucket criado especificamente para o CloudWatch Logs. No entanto, se você desejar usar um bucket existente, vá para a etapa 2.

> Note
>
> O bucket do Amazon S3 deve residir na mesma região que os dados de log a serem exportados. O CloudWatch Logs não oferece suporte à exportação de dados para buckets do Amazon S3 em uma região diferente.

Para criar um bucket do Amazon S3 com a AWS CLI

Em um prompt de comando, execute o seguinte comando create-bucket, em que `LocationConstraint` é a região onde você está exportando dados de log.

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

Veja a seguir um exemplo de saída:

```
{
    "Location": "/my-exported-logs"
}
```

Etapa 2. Criar um IAM Utilizador com acesso total a Amazon S3 e CloudWatch Logs

Nas etapas a seguir, você criará o usuário do IAM com permissões necessárias.

Para criar o usuário e atribuir permissões

1. Crie o usuário do IAM inserindo o seguinte comando.

   ```
   aws iam create-user --user-name CWLExportUser
   ```

2. Associe as políticas gerenciadas do IAM ao usuário do IAM que você acabou de criar.

   ```
   export S3POLICYARN=$(aws iam list-policies --query 'Policies[?
   PolicyName==`AmazonS3FullAccess`].{ARN:Arn}' --output text)
   ```

   ```
   export CWLPOLICYARN=$( aws iam list-policies --query 'Policies[?
   PolicyName==`CloudWatchLogsFullAccess`].{ARN:Arn}' --output text)
   ```

```
aws iam attach-user-policy --user-name CWLExportUser --policy-arn $S3POLICYARN
```

```
aws iam attach-user-policy --user-name CWLExportUser --policy-arn $CWLPOLICYARN
```

3. Confirme se as duas políticas gerenciadas estão associadas.

```
aws iam list-attached-user-policies --user-name CWLExportUser
```

4. Configure o seu AWS CLI para incluir o IAM credenciais do *CWLExportUser* Usuário do IAM Para obter mais informações, consulte Configuração da AWS CLI.

Etapa 3 Definir permissões num Amazon S3 Balde

Por padrão, todos os objetos e buckets do Amazon S3 são privados. Somente o proprietário do recurso e a conta que criou o bucket podem acessar o bucket e todos os objetos que ele contém. No entanto, o proprietário do recurso pode optar por conceder permissões de acesso a outros recursos e usuários ao criar uma política de acesso padrão.

Para definir permissões em um bucket do Amazon S3

1. Crie um arquivo chamado `policy.json` e adicione a seguinte política de acesso, mudando `Resource` para o nome do seu bucket do S3 e `Principal` para o endpoint da região onde você está exportando dados de log. Use um editor de texto para criar este arquivo de política. Não use o console do IAM.

   - Se o bucket estiver em sua conta, use a seguinte política.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "s3:GetBucketAcl",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs",
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        }
    ]
}
```

   - Se o bucket estiver em outra conta, use a seguinte política. Ela inclui uma instrução adicional utilizando o usuário do IAM que você criou na etapa anterior.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
```

```
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
            "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
            "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
            "Principal": { "AWS": "arn:aws:iam::SendingAccountID:user/CWLExportUser" }
        }
    ]
}
```

- Se o bucket estiver em outra conta e você estiver usando uma função do IAM em vez de um usuário do IAM, use a política a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
            "Action": "s3:GetBucketAcl",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs",
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
            "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
            "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
            "Principal": { "AWS": "arn:aws:iam::SendingAccountID:role/CWLExportUser" }
        }
    ]
}
```

2. Defina a política que você acabou de adicionar como política de acesso ao seu bucket usando o comando put-bucket-policy. Essa política permite que o CloudWatch Logs exporte dados de log para o seu bucket do Amazon S3. O proprietário do bucket terá permissões completas sobre todos os objetos exportados.

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

   Warning

   Se o bucket existente já tiver uma ou mais políticas anexadas, adicione as declarações de acesso do CloudWatch Logs a essa política ou a essas políticas. Recomendamos avaliar

o conjunto resultante de permissões para ter certeza de que elas são apropriadas para os usuários que acessarão o bucket.

Etapa 4. Criar uma tarefa de exportação

Depois de criar a tarefa de exportação para exportar os logs de um grupo de logs, a tarefa poderá demorar de alguns segundos a algumas horas, dependendo do tamanho dos dados a serem exportados.

Para criar uma tarefa de exportação usando a AWS CLI

Em um prompt de comando, use o seguinte comando create-export-task para criar a tarefa de exportação.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-group-09-10-2015"
 --log-group-name "my-log-group" --from 1441490400000 --to 1441494000000 --destination "my-
exported-logs" --destination-prefix "export-task-output"
```

Veja a seguir um exemplo de saída:

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Etapa 5. Descrever as tarefas de exportação

Depois de criar uma tarefa de exportação, você pode obter o status atual da tarefa.

Para descrever as tarefas de exportação usando a AWS CLI

No prompt de comando, use o seguinte comando describe-export-tasks.

```
aws logs --profile CWLExportUser describe-export-tasks --task-id "cda45419-90ea-4db5-9833-
aade86253e66"
```

Veja a seguir um exemplo de saída:

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "RUNNING",
         "message": "Started Successfully"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "tTo": 1441494000000
   }]
}
```

Você pode usar o comando `describe-export-tasks` de três maneiras diferentes:

- Sem filtros: Mostra todas as suas tarefas de exportação, em ordem inversa de criação.
- Filtro na ID da tarefa: Lista a tarefa de exportação, se existir, com a ID especificada.
- Filtro no estado da tarefa: Lista as tarefas de exportação com o estado especificado.

Por exemplo, use o comando a seguir para filtrar com base no status `FAILED`.

```
aws logs --profile CWLExportUser describe-export-tasks --status-code "FAILED"
```

Veja a seguir um exemplo de saída:

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "completionTime": 1441498600000
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "FAILED",
         "message": "FAILED"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "to": 1441494000000
   }]
}
```

Etapa 6. Cancelar uma tarefa de exportação

Você pode cancelar uma tarefa de exportação se ela estiver no estado `PENDING` ou `RUNNING`.

Para cancelar uma tarefa de exportação usando a AWS CLI

No prompt de comando, use o seguinte comando cancel-export-task:

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-
aade86253e66"
```

Você pode usar o comando describe-export-tasks para verificar se a tarefa foi cancelada com êxito.

Streaming dados do CloudWatch Logs para Amazon Elasticsearch Service

Você pode configurar um grupo de logs do CloudWatch Logs para fazer streaming de dados recebidos para o cluster do Amazon Elasticsearch Service (Amazon ES) em tempo quase real por meio de uma assinatura do CloudWatch Logs Para obter mais informações, consulte Processamento em tempo real de dados de log com assinaturas (p. 87).

Dependendo da quantidade de dados de log que estão sendo enviados por streaming, você pode definir um limite de execução simultânea no nível da função. Para obter mais informações, consulte Limite de execução simultânea no nível da função.

> Note
>
> O streaming de grandes volumes de dados do CloudWatch Logs para o Amazon ES pode resultar em altas cobranças de uso. Recomendamos que você crie um orçamento no console de Faturamento e Gerenciamento de Custos. Para obter mais informações, consulte Gerenciamento de seus custos com orçamentos.

Prerequisites

Antes de começar, crie um domínio do Amazon ES O domínio do Amazon ES pode ter acesso público ou acesso à VPC, mas você não poderá modificar o tipo de acesso depois que o domínio for criado. Você pode querer revisar as configurações do domínio do Amazon ES mais tarde e modificar a configuração do cluster com base na quantidade de dados que seu cluster processará.

Para obter mais informações sobre o Amazon ES, consulte oGuia do desenvolvedor do Amazon Elasticsearch Service.

Para criar um domínio do Amazon ES

Em um prompt de comando, use o seguinte comando create-elasticsearch-domain:

```
aws es create-elasticsearch-domain --domain-name my-domain
```

Inscrever um grupo de logs no Amazon ES

Você pode usar o console do CloudWatch para inscrever um grupo de logs no Amazon ES.

Para inscrever um grupo de logs no Amazon ES

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Log groups (Grupos de logs).
3. Escolha o nome do grupo de logs.

4. Escolha Actions (Ações), Create Elasticsearch subscription filter (Criar filtro de assinatura do Elasticsearch).
5. Escolha se deseja fazer streaming para um cluster nessa conta ou em outra conta.
6. Em Cluster do Amazon ES, escolha o cluster que você criou na etapa anterior.
7. Sob Função do Lambda, em Função de execução do Lambda para IAM, escolha a função do IAM que o Lambda deve usar ao executar chamadas para o Amazon ES e selecione Next (Próximo).

   A função do IAM escolhida deve atender a estes requisitos:

   - Ela deve possuir `lambda.amazonaws.com` na relação de confiança.
   - Ela deve incluir a seguinte política:

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Action": [
                   "es:*"
               ],
               "Effect": "Allow",
               "Resource": "arn:aws:es:region:account-id:domain/target-domain-name/*"
           }
       ]
   }
   ```

   - Se o domínio do Amazon ES de destino usar o acesso à VPC, a função deverá ter a política AWSLambdaVPCAccessExecutionRole anexada. Essa política gerenciada pela Amazon concede ao Lambda acesso à VPC do cliente, permitindo que o Lambda grave no endpoint do Amazon ES na VPC.
8. Em Formato de log, escolha um formato de log.
9. Em Padrão de filtro de assinatura, digite os termos ou o padrão a ser localizado nos eventos de log. Dessa forma, você garante que enviará somente os dados que o interessam para o cluster do Amazon ES Para obter mais informações, consulte Criar métricas a partir de eventos de log usando filtros (p. 71).
10. (Opcional) Em Select Log Data to Test (Selecionar dados de log para testar), selecione um fluxo de logs e escolha Test Pattern (Testar padrão) para verificar se o filtro de pesquisa está retornando os resultados esperados.
11. Selecione Iniciar streaming.

Serviços da AWS que publicam logs no CloudWatch Logs

Os seguintes serviços da AWS publicam métricas no CloudWatch Logs. Para obter informações sobre os logs que esses serviços enviam, consulte a documentação no link.

Serviço .	Documentação
Amazon API Gateway	Configuração CloudWatch Início de sessão da API API Gateway
Amazon Aurora MySQL	Publicação Amazon Aurora MySQL Registos para Amazon CloudWatch Logs
AWS CloudHSM	Monitorização AWS CloudHSM Audit Logs in (Registos de auditoria) Amazon CloudWatch Logs
AWS CloudTrail	Monitorização CloudTrail Ficheiros de registo com Amazon CloudWatch Logs
Amazon Cognito	Criação da função do IAM do CloudWatch Logs
Amazon Connect	Registro em log e monitoramento Amazon Connect
AWS DataSync	Permitir DataSync para carregar registos à Amazon CloudWatch Grupos de registo
AWS Elastic Beanstalk	Utilizar Elastic Beanstalk com Amazon CloudWatch Logs
Amazon Elastic Container Service	Como usar o CloudWatch Logs com instâncias de contêiner
Amazon Elastic Kubernetes Service	Amazon Amazon Elastic Kubernetes Service Controlo do plano de controlo
AWS Fargate	Como usar o driver de log awslogs
AWS Glue	Registro contínuo para tarefas do AWS Glue
AWS IoT	Monitorização com CloudWatch Logs
AWS Lambda	Aceder Amazon CloudWatch Logs para AWS Lambda
Amazon MQ	Configurar Amazon MQ publicar registos gerais e de auditoria para Amazon CloudWatch Logs
AWS OpsWorks	Usar o Amazon CloudWatch Logs com o AWS OpsWorks Stacks
Amazon Relational Database Service	Publicar registos postgresql para CloudWatch Logs
AWS RoboMaker	Robomaker de robô AWS para nós ROS com suporte offline

Serviço .	Documentação
Amazon Route 53	Registo e monitorização na Amazon Route 53
Amazon SageMaker	Registo Amazon SageMaker Eventos com Amazon CloudWatch
Amazon Simple Notification Service	Visualização CloudWatch Logs
Amazon VPC	VPC Flow Logs

Segurança no Amazon CloudWatch Logs

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se beneficiará de um datacenter e de uma arquitetura de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

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

- Segurança da nuvem – A AWS é responsável pela proteção da infraestrutura que executa serviços da AWS na nuvem da AWS. A AWS também fornece serviços que você pode usar com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos Programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam ao Amazon WorkSpaces, consulte Produtos da AWS no escopo por programa de conformidade.
- Segurança na nuvem – Sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade dos dados, os requisitos da empresa e as leis e os regulamentos aplicáveis

Esta documentação ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon CloudWatch Logs. Ela mostra como configurar o Amazon CloudWatch Logs para atender aos seus objetivos de segurança e conformidade. Saiba também como usar outros produtos da AWS que ajudam você a monitorar e proteger os recursos do CloudWatch Logs.

Tópicos

Proteção de dados no Amazon CloudWatch Logs

O modelo de responsabilidade compartilhada da AWS se aplica à proteção de dados no Amazon CloudWatch Logs. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Esse conteúdo inclui as tarefas de configuração e gerenciamento de segurança dos serviços da AWS que você usa. Para obter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para obter mais informações sobre a proteção de dados na Europa, consulte o blog AWS Shared Responsibility Model and GDPR no Blog de segurança da AWS.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da conta da AWS e configure as contas de usuário individuais com o AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da AWS. Recomendamos TLS 1.2 ou posterior.
- Configure a API e o registro em log das atividades do usuário com o AWS CloudTrail.
- Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão nos serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados pessoais que são armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para obter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que você nunca coloque informações de identificação confidenciais, como números de conta dos seus clientes, em campos de formato livre, como um campo Name (Nome). Isso inclui quando você trabalhar com o CloudWatch Logs ou outros serviços da AWS usando o console, a API, a AWS CLI ou os AWS SDKs. Todos os dados inseridos por você no CloudWatch Logs ou em outros serviços podem ser separados para inclusão em logs de diagnóstico. Ao fornecer um URL para um servidor externo, não inclua informações de credenciais no URL para validar a solicitação a esse servidor.

Criptografia em repouso

CloudWatch LogsO protege os dados em repouso usando criptografia. Todos os grupos de logs são criptografados. Por padrão, o serviço do CloudWatch Logs gerencia as chaves de criptografia no lado do servidor.

Se quiser gerenciar as chaves usadas para criptografar e descriptografar os logs, use as chaves mestras do cliente (CMK) do AWS Key Management Service. Para obter mais informações, consulte Criptografar dados de log no CloudWatch Logs usando o AWS Key Management Service (p. 64).

Criptografia em trânsito

CloudWatch LogsO usa criptografia de ponta a ponta de dados em trânsito. O CloudWatch Logs gerencia as chaves de criptografia no lado do servidor.

Identity and Access Management para o Amazon CloudWatch Logs

O acesso ao Amazon CloudWatch Logs exige credenciais que a AWS pode usar para autenticar suas solicitações. Essas credenciais devem ter permissões para acessar os recursos da AWS, como recuperar dados do CloudWatch Logs sobre seus recursos de nuvem. As seções a seguir fornecem detalhes sobre como é possível usar o AWS Identity and Access Management (IAM) e o CloudWatch Logs para ajudar a proteger seus recursos controlando quem pode acessá-los.

Authentication

Você pode acessar a AWS como qualquer um dos seguintes tipos de identidades:

- &Usuário raiz da conta da AWS – quando se cadastrar na AWS, você fornece um endereço de e-mail e uma senha que é associada à sua conta da AWS. Estas são suas credenciais raiz e fornecem acesso total a todos os seus recursos da AWS.

  Important

  Por motivos de segurança, recomendamos que você use as credenciais raiz para criar somente um usuário administrador, que é um usuário do IAM com permissões totais à sua conta da AWS. Em seguida, use esse usuário administrador para criar outros usuários e funções do IAM com permissões limitadas. Para obter mais informações, consulte Melhores práticas do IAM e Criar um usuário e grupo administrativo no Guia do usuário do IAM.

- IAM usuário – Um IAM usuário é simplesmente uma identidade na qual sua conta da AWS possui permissões personalizadas específicas (por exemplo, para visualizar métricas no CloudWatch Logs). Você pode usar um nome e uma senha de usuário IAM para fazer o login em páginas da Web seguras da AWS, como Console de gerenciamento da AWS, Fóruns de discussão AWS ou AWS Support Center.

  Além de uma senha e um nome do usuário, você também pode gerar chaves de acesso para cada usuário. Você pode usar essas chaves ao acessar serviços da AWS de forma programática, seja com um dos vários SDKs ou usando a AWS Command Line Interface (AWS CLI). As ferramentas de SDK e de CLI usam as chaves de acesso para assinar sua solicitação de forma criptográfica. Se não utilizar as ferramentas AWS, tem de assinar o pedido. CloudWatch Logs supports Assinatura Versão 4, um protocolo para autenticar pedidos de API de entrada. Para obter mais informações sobre a autenticação de solicitações, consulte Processo de cadastramento do Signature versão 4 na AWS General Reference.

- Função do IAM – Uma função do IAM é outra identidade do IAM que você pode criar na sua conta que tenha permissões específicas. É semelhante a um usuário do IAM, mas não está associada a uma pessoa específica. Uma função do IAM permite obter chaves de acesso temporárias que podem ser usadas para acessar os serviços e recursos da AWS. As funções do IAM com credenciais temporária são úteis nas seguintes situações:

  - Acesso de usuário federado – Em vez de criar um usuário do IAM, você pode usar identidades já existente de usuário de AWS Directory Service, o diretório de usuário da sua companhia ou um provedor de identidades da web. Eles são conhecidos como usuários federados. A AWS aloca uma função a um usuário federado quando o acesso é solicitado através de um provedor de identidades. Para obter mais informações sobre usuários federados, consulte Usuários federados e funções no Guia do usuário do IAM.

  - Acesso entre contas – Você pode usar uma função do IAM em sua conta para conceder, a outra conta da AWS, permissões de acesso aos recursos da sua conta. Para um exemplo, consulte Tutorial: Delegar acesso a contas AWS utilizando funções IAM no Guia do usuário do IAM.

  - Acesso ao serviço da AWS – você pode usar uma função do IAM na sua conta para conceder permissões de serviço da AWS para acessar os recursos da sua conta. Por exemplo, você pode criar uma função que permita ao Amazon Redshift acessar um bucket do Amazon S3 em seu nome e carregar dados armazenados no bucket em um cluster do Amazon Redshift. Para obter mais informações, consulte Criar uma função para delegar permissões a um serviço da AWS no Guia do usuário do IAM.

  - Aplicativos executados no Amazon EC2 – Em vez de armazenar chaves de acesso na instância do EC2 a serem usadas em aplicativos em execução na instância e fazer solicitações de API da

AWS, você pode usar uma função do IAM para gerenciar credenciais temporárias para esses aplicativos. Para atribuir uma função de AWS a uma instância de EC2 e disponibilizá-la para todas as suas aplicações, crie um perfil de instância que esteja anexado à instância. Um perfil de instância contém a função e permite que programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Usar funções para aplicativos no Amazon EC2 no Guia do usuário do IAM.

Controle de acesso

É possível ter credenciais válidas para autenticar suas solicitações. No entanto, a menos que tenha permissões, não é possível criar nem acessar os recursos do CloudWatch Logs. Por exemplo, você deve ter permissões para criar streams de log, criar grupos de logs, etc.

As seções a seguir descrevem como gerenciar permissões para o CloudWatch Logs. Recomendamos que você leia a visão geral primeiro.

Visão geral do gerenciamento de permissões de acesso aos seus recursos do CloudWatch Logs

Cada recurso da AWS é de propriedade de uma conta da AWS, e as permissões para criar ou acessar um recurso são regidas por políticas de permissões. Um administrador de conta pode anexar políticas de permissões a identidades do IAM (ou seja, usuários, grupos e funções), e alguns serviços (como o AWS Lambda) também oferecem suporte à anexação de políticas de permissões a recursos.

Note

Um administrador da conta (ou usuário do IAM administrador) é um usuário com privilégios de administrador. Para obter mais informações, consulte Melhores práticas do IAM no Guia do usuário do IAM.

Ao conceder permissões, você decide quem recebe as permissões, os recursos relacionados às permissões concedidas e as ações específicas que deseja permitir nesses recursos.

Tópicos

Recursos e operações do CloudWatch Logs

No CloudWatch Logs, os principais recursos são grupos de logs, fluxos de log e destinos. O CloudWatch Logs não oferece suporte a sub-recursos (outros recursos para uso com o recurso principal).

Esses recursos e sub-recursos têm Nomes de recursos da Amazon (ARNs) exclusivos associados a eles, conforme mostrado na tabela a seguir.

Tipo de recurso	Formato do nome de recurso da Amazon (ARN)
Grupo de logs	arn:aws:registos:*region*:*account-id*:grupo log:*log_group_name*
Stream de log	arn:aws:registos:*region*:*account-id*:grupo log:*log_group_name*:log-stream:*log-stream-name*
Destino	arn:aws:registos:*region*:*account-id*destination*destination_name*

Para obter mais informações sobre ARNs, consulte ARNs no Guia do usuário do IAM. Para obter mais informações sobre ARNs do CloudWatch Logs, consulte Nomes de recurso da Amazon (ARNs) e namespaces do serviço da AWS no Referência geral do Amazon Web Services. Para obter um exemplo de uma política que abranja o CloudWatch Logs, consulte Uso de políticas baseadas em identidade (Políticas do IAM) para o CloudWatch Logs (p. 134).

O CloudWatch Logs fornece um conjunto de operações para funcionar com recursos do CloudWatch Logs. Para ver uma lista das operações disponíveis, consulte Referência de permissões CloudWatch Logs (p. 139).

Entender a propriedade de recursos

A conta da AWS é proprietária dos recursos criados na conta, independentemente de quem os criou. Mais especificamente, o proprietário do recurso é a conta da AWS da entidade principal (ou seja, a conta-raiz, um usuário do IAM ou uma função do IAM) que autentica a solicitação de criação de recursos. Os exemplos a seguir ilustram como isso funciona.

- Se você usar as credenciais da conta-raiz da sua conta da AWS para criar um grupo de logs, sua conta da AWS será a proprietária do recurso do CloudWatch Logs.
- Se você criar um usuário do IAM na sua conta da AWS e conceder permissões para criar recursos do CloudWatch Logs para esse usuário, ele poderá criar recursos do CloudWatch Logs. No entanto, a sua conta da AWS, à qual o usuário pertence, é a proprietária dos recursos do CloudWatch Logs.
- Se você criar uma função do IAM em sua conta da AWS com permissões para criar recursos do CloudWatch Logs, qualquer pessoa que possa assumir a função poderá criar recursos do CloudWatch Logs. Sua conta da AWS, à qual a função pertence, é a proprietária dos recursos do CloudWatch Logs.

Gerenciamento do acesso aos recursos

A política de permissões descreve quem tem acesso a quê. A seção a seguir explica as opções disponíveis para a criação das políticas de permissões.

> **Note**
>
> Esta seção aborda como usar o IAM no contexto do CloudWatch Logs. Não são fornecidas informações detalhadas sobre o serviço do IAM. Para obter a documentação completa do IAM, consulte O que é o IAM? no Guia do usuário do IAM. Para obter informações sobre a sintaxe e as descrições de política do IAM, consulte Referência de política do IAM no Guia do usuário do IAM.

As políticas anexadas a uma identidade do IAM são conhecidas como políticas baseadas em identidade (políticas do IAM) e as políticas anexadas a um recurso são conhecidas como políticas baseadas em recurso. O CloudWatch Logs oferece suporte para políticas baseadas em identidade e políticas baseadas em recurso para destinos, que são usadas para permitir assinaturas entre contas. Para obter mais informações, consulte Compartilhamento de dados de log entre contas com assinaturas (p. 99).

Tópicos

Aprovações do grupo de registos e perceções do contribuidor

As Perceções do Contribuidor são uma característica de CloudWatch que lhe permite analisar dados de grupos de registos e criar séries temporais que apresentam dados de contribuidor. É possível ver métricas sobre os principais colaboradores, o número total de colaboradores exclusivos e o uso deles. Para obter mais informações, consulte Usar o Contributor Insights para analisar dados de alta cardinalidade.

Quando concede um utilizador ao `cloudwatch:PutInsightRule` e `cloudwatch:GetInsightRuleReport` permissões, esse utilizador pode criar uma regra que avalia qualquer grupo de registo em CloudWatch Logs e depois ver os resultados. Os resultados podem conter dados de contribuidor para esses grupos de registo. Certifique-se de que concede estas permissões apenas aos utilizadores que devem conseguir visualizar estes dados.

Políticas baseadas em identidade (políticas do IAM)

Você pode anexar políticas a identidades do IAM. Por exemplo, você pode fazer o seguinte:

- Associar uma política de permissões a um usuário ou a um grupo na conta – para conceder a um usuário permissões para visualizar logs no console do CloudWatch Logs é possível associar uma política de permissões a um usuário ou ao grupo a que o usuário pertence.
- Anexar uma política de permissões a uma função (conceder permissões entre contas) – você poderá anexar uma política de permissões com base em identidade a uma função do IAM para conceder permissões entre contas. Por exemplo, o administrador na Conta A pode criar uma função para conceder permissões entre contas a outra conta da AWS (por exemplo, Conta B) ou um serviço da AWS da seguinte forma:
  1. Um administrador da Conta A cria uma função do IAM e anexa uma política de permissões à função que concede permissões em recursos da Conta A.
  2. Um administrador da Conta A anexa uma política de confiança à função identificando a Conta B como a principal, que pode assumir a função.
  3. O administrador da Conta B poderá delegar permissões para assumir a função para todos os usuários na Conta B. Isso permite que os usuários na Conta B criem ou acessem os recursos na Conta A. O principal na política de confiança também poderá ser um serviço da AWS principal se você quiser conceder a um serviço da AWS permissões para assumir a função.

  Para obter mais informações sobre como usar o IAM para delegar permissões, consulte Gerenciamento de acesso no Guia do usuário do IAM.

Veja a seguir um exemplo de política que concede permissões para as ações `logs:PutLogEvents`, `logs:CreateLogGroup` e `logs:CreateLogStream` em todos os recursos no us-east-1. Para grupos de logs, o CloudWatch Logs oferece suporte à identificação de recursos específicos usando ARNs de recursos (também chamados de permissões de recursos) para algumas das ações de API. Se você deseja incluir todos os grupos de log, especifique o caractere curinga (*).

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"",
         "Effect":"Allow",
         "Action":[
            "logs:PutLogEvents",
```

```
            "logs:CreateLogGroup",
            "logs:CreateLogStream"
         ],
         "Resource":"arn:aws:logs:us-east-1:*:*"
      }
   ]
}
```

Para obter mais informações sobre como usar políticas baseadas em identidade com o CloudWatch Logs, consulte Uso de políticas baseadas em identidade (Políticas do IAM) para o CloudWatch Logs (p. 134). Para obter mais informações sobre usuários, grupos, funções e permissões, consulte Identidades (usuários, grupos e funções) no Guia do usuário do IAM.

Políticas baseadas em recursos

O CloudWatch Logs é compatível com políticas baseadas em recursos para destinos, que você pode usar para ativar assinaturas entre contas. Para obter mais informações, consulte Criar um destino (p. 100). É possível criar destinos usando a API PutDestination, e você pode adicionar uma política de recursos ao destino usando a API PutDestination. O exemplo a seguir permite que outra conta da AWS com o ID da conta 111122223333 assine seus grupos de logs no destino `arn:aws:logs:us-east-1:123456789012:destination:testDestination`.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111122223333"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-east-1:123456789012:destination:testDestination"
    }
  ]
}
```

Especificar elementos da política: Acções, efeitos e princípios

Para cada recurso do CloudWatch Logs, o serviço define um conjunto de operações da API. Para conceder permissões a essas operações da API, o CloudWatch Logs define um conjunto de ações que podem ser especificadas em uma política. Algumas operações da API podem exigir permissões para mais de uma ação a fim de realizar a operação da API. Para obter mais informações sobre os recursos e operações da API, consulte Recursos e operações do CloudWatch Logs (p. 130) e Referência de permissões CloudWatch Logs (p. 139).

Estes são os elementos de política básicos:

- Recurso – Você usa um Amazon Resource Name (ARN) para identificar o recurso ao qual a política se aplica. Para obter mais informações, consulte Recursos e operações do CloudWatch Logs (p. 130).
- Ação – use palavras-chave de ação para identificar operações de recursos que você deseja permitir ou negar. Por exemplo, a permissão `logs.DescribeLogGroups` permite que o usuário execute a operação `DescribeLogGroups`.
- Efeito – você especifica o efeito, permitir ou negar, quando o usuário solicita a ação específica. Se você não conceder explicitamente acesso para permitir um recurso, o acesso estará implicitamente negado. Você também pode negar explicitamente o acesso a um recurso, o que pode fazer para ter a certeza de que um usuário não consiga acessá-lo, mesmo que uma política diferente conceda acesso.
- Principal – em políticas baseadas em identidade (IAM políticas), o usuário ao qual a política está anexada é implicitamente o principal. Para as políticas baseadas em recursos, você especifica quais

usuários, contas, serviços ou outras entidades deseja que recebam permissões (aplica-se somente a políticas baseadas em recursos). O CloudWatch Logs oferece suporte a políticas baseadas em recurso para destinos.

Para saber mais sobre a sintaxe e as descrições de políticas do IAM, consulte Referência de política do IAM da AWS no Guia do usuário do IAM.

Para obter uma tabela que mostre todas as ações da API do CloudWatch Logs e os recursos a que elas se aplicam, consulte Referência de permissões CloudWatch Logs (p. 139).

Especificar condições em uma política

Ao conceder permissões, você pode usar a linguagem da política de acesso para especificar as condições quando uma política deve entrar em vigor. Por exemplo, convém que uma política só seja aplicada após uma data específica. Para obter mais informações sobre como especificar condições em uma linguagem de política, consulte Condição no Guia do usuário do IAM.

Para expressar condições, você usa chaves de condição predefinidas. Para obter uma lista de chaves de contexto aceitas pelos serviços da AWS e uma lista de chaves de política gerais da AWS, consulte Ações de serviços e chaves de contexto de condição da AWS e Chaves de contexto de condição do IAM e globais no Guia do usuário do IAM.

Uso de políticas baseadas em identidade (Políticas do IAM) para o CloudWatch Logs

Este tópico fornece exemplos de políticas baseadas em identidade em que um administrador de conta pode anexar políticas de permissões a identidades do IAM (ou seja, usuários, grupos e funções).

> Important
>
> Recomendamos analisar primeiro os tópicos introdutórios que explicam os conceitos básicos e as opções disponíveis para gerenciar o acesso aos recursos do CloudWatch Logs. Para obter mais informações, consulte Visão geral do gerenciamento de permissões de acesso aos seus recursos do CloudWatch Logs (p. 130).

Este tópico abrange o seguinte:

Veja a seguir um exemplo de política de permissões:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "arn:aws:logs:*:*:*"
    ]
  }
```

```
  ]
}
```

Essa política tem uma instrução que concede permissões para criar grupos e streams de log para fazer upload de eventos de log para streams de log e listar detalhes sobre streams de log.

O caractere curinga (*) no final do valor `Resource` significa que a instrução dá permissões para as ações `logs:CreateLogGroup`, `logs:CreateLogStream`, `logs:PutLogEvents` e `logs:DescribeLogStreams` em qualquer grupo de logs. Para limitar essa permissão a um determinado grupo de logs, substitua o caractere curinga (*) no ARN do recurso pelo ARN do grupo de logs específico. Para obter mais informações sobre as seções em uma declaração de política do IAM, consulte Referência a elementos de políticas do IAM no Guia do usuário do IAM. Para obter uma lista que mostre todas as ações do CloudWatch Logs, consulte Referência de permissões CloudWatch Logs (p. 139).

Permissões necessárias para usar o console do CloudWatch

Para que um usuário trabalhe com o CloudWatch Logs no console do CloudWatch, ele deve ter um conjunto mínimo de permissões que lhe permita descrever outros recursos da AWS em sua conta da AWS. Para usar o CloudWatch Logs no console do CloudWatch, você deve ter permissões dos seguintes serviços:

- CloudWatch
- CloudWatch Logs
- Amazon ES
- IAM
- Kinesis
- Lambda
- Amazon S3

Se você criar uma política do IAM que seja mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para os usuários com essa política do IAM. Para garantir que esses usuários ainda consigam usar o console do CloudWatch, associe também a política gerenciada `CloudWatchReadOnlyAccess` ao usuário, conforme descrito em Políticas gerenciadas (predefinidas) da AWS do CloudWatch Logs (p. 136).

Não é necessário conceder permissões mínimas do console para usuários que fazem chamadas somente à AWS CLI ou à API da CloudWatch Logs.

O conjunto completo de permissões necessárias para trabalhar com o console do CloudWatch para um usuário que não está usando o console para gerenciar assinaturas de log inclui:

- cloudwatch:getMetricData
- cloudwatch:listMetrics
- logs:cancelExportTask
- logs:createExportTask
- logs:createLogGroup
- logs:createLogStream
- logs:deleteLogGroup
- logs:deleteLogStream
- logs:deleteMetricFilter
- logs:deleteQueryDefinition
- logs:deleteRetentionPolicy
- logs:deleteSubscriptionFilter

- logs:describeExportTasks
- logs:describeLogGroups
- logs:describeLogStreams
- logs:describeMetricFilters
- logs:describeQueryDefinitions
- logs:describeSubscriptionFilters
- logs:filterLogEvents
- logs:getLogEvents
- logs:putMetricFilter
- logs:putQueryDefinition
- logs:putRetentionPolicy
- logs:putSubscriptionFilter
- logs:testMetricFilter

Para um usuário que também usará o console para gerenciar assinaturas de log, as seguintes permissões também são necessárias:

- es:describeElasticsearchDomain
- es:listDomainNames
- iam:attachRolePolicy
- iam:createRole
- iam:getPolicy
- iam:getPolicyVersion
- iam:getRole
- iam:listAttachedRolePolicies
- iam:listRoles
- kinesis:describeStreams
- kinesis:listStreams
- lambda:addPermission
- lambda:createFunction
- lambda:getFunctionConfiguration
- lambda:listAliases
- lambda:listFunctions
- lambda:listVersionsByFunction
- lambda:removePermission
- s3:listBuckets

Políticas gerenciadas (predefinidas) da AWS do CloudWatch Logs

A AWS aborda muitos casos de uso comuns fornecendo políticas autônomas do IAM que são criadas e administradas pela AWS. As políticas gerenciadas concedem permissões necessárias para casos de uso comuns, de maneira que você possa evitar a necessidade de investigar quais permissões são necessárias. Para obter mais informações, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

As seguintes políticas gerenciadas pela AWS, que você pode associar a usuários na sua conta, são específicas do CloudWatch Logs:

- CloudWatchLogsFullAccess – Concede acesso total ao CloudWatch Logs.
- CloudWatchLogsReadOnlyAccess – Concede acesso somente leitura ao CloudWatch Logs.

> Note
>
> Você pode analisar essas políticas de permissões fazendo login no console do IAM e procurando políticas específicas.

(Também é possível criar suas próprias políticas do IAM personalizadas a fim de conceder permissões para ações e recursos do CloudWatch Logs). Você pode anexar essas políticas personalizadas a usuários ou grupos do IAM que exijam essas permissões.

Exemplos de política gerenciada pelo cliente

Nesta seção, você encontrará exemplos de políticas de usuário que concedem permissões para várias ações do CloudWatch Logs. Essas políticas funcionam quando você está usando a API do CloudWatch Logs, AWS SDKs ou a AWS CLI.

Exemplos:

- Exemplo: = 1. Permitir acesso total a CloudWatch Logs (p. 137)
- Exemplo: = 2. Permitir acesso só de leitura a CloudWatch Logs (p. 137)
- Exemplo 3 Permitir acesso a um grupo de registo (p. 138)

Exemplo: = 1. Permitir acesso total a CloudWatch Logs

A seguinte política permite que um usuário acesse todas as ações do CloudWatch Logs.

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

Exemplo: = 2. Permitir acesso só de leitura a CloudWatch Logs

O AWS fornece uma política CloudWatchLogsReadOnlyAccess que permite o acesso somente leitura aos dados do CloudWatch Logs. Esta política inclui as seguintes permissões.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents"
            ],
```

```
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Exemplo 3 Permitir acesso a um grupo de registo

A política a seguir permite que um usuário leia e grave eventos de log em um grupo de logs especificado.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName:*"
      }
   ]
}
```

Usar tags e políticas do IAM para controle no nível do grupo de logs

Você pode conceder aos usuários acesso a determinados grupos de logs enquanto os impede de acessar outros grupos de logs. Para fazer isso, marque seus grupos de logs e use políticas do IAM que fazem referência a essas tags.

Para obter mais informações sobre como marcar grupos de logs, consulte Marcar grupos de logs no Amazon CloudWatch Logs (p. 62).

Quando você marca grupos de logs, você pode conceder uma política do IAM a um usuário para permitir o acesso a apenas os grupos de logs com determinada tag. Por exemplo, a instrução de política a seguir concede acesso a apenas grupos de logs com o valor `Green` para a chave de tag `Team`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "logs:ResourceTag/Team": "Green"
                }
            }
        }
    ]
}
```

Para obter mais informações sobre o uso de declarações de política do IAM, consulte Controle de acesso usando políticas no Guia do usuário do IAM.

Referência de permissões CloudWatch Logs

Ao configurar o Controle de acesso (p. 130) e escrever políticas de permissões que você pode associar a uma identidade do IAM (políticas com base em identidade), você pode usar a tabela a seguir como referência. A tabela lista cada operação da API do CloudWatch Logs e as ações correspondentes para as quais você pode conceder permissões para executar a ação. Você especifica as ações no campo `Action` das políticas. Para o campo `Resource`, você pode especificar o ARN de um grupo de logs ou fluxo de logs ou especificar `*` para representar todos os recursos do CloudWatch Logs.

Você pode usar as chaves de condição em toda a AWS nas suas políticas do CloudWatch Logs para expressar condições. Para obter uma lista completa das chaves de toda a AWS, consulte Chaves de contexto globais do AWS e de condição do IAM no Guia do usuário do IAM.

> **Note**
>
> Para especificar uma ação, use o prefixo `logs:` seguido do nome da operação da API. Por exemplo, . `logs:CreateLogGroup`, `logs:CreateLogStream`, ou `logs:*` (para todos CloudWatch Logs ações).

Operações de API e permissões necessárias para ações no CloudWatch Logs

CloudWatch Logs Operações de API	Permissões necessárias (Ações da API):
CancelExportTask	`logs:CancelExportTask` Necessária para cancelar uma tarefa de exportação pendente ou em execução.
CreateExportTask	`logs:CreateExportTask` Necessária para exportar dados de um grupo de logs para um bucket do Amazon S3.
CreateLogGroup	`logs:CreateLogGroup` Necessária para criar um novo grupo de logs.
CreateLogStream	`logs:CreateLogStream` Necessária para criar um novo stream de logs em um grupo de logs.
DeleteDestination	`logs:DeleteDestination` Necessária para excluir um destino de log e desativar seus filtros de assinatura.
DeleteLogGroup	`logs:DeleteLogGroup` Necessária para excluir um grupo de logs e eventos de log arquivados associados.
DeleteLogStream	`logs:DeleteLogStream` Necessária para excluir um stream de logs e eventos de log arquivados associados.
DeleteMetricFilter	`logs:DeleteMetricFilter` Necessária para excluir um filtro de métrica associado a um grupo de logs.

CloudWatch Logs Operações de API	Permissões necessárias (Ações da API):
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Necessária para excluir uma definição de consulta salva no CloudWatch Logs Insights.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Necessário para excluir uma política de recursos do CloudWatch Logs.
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Necessária para excluir uma política de retenção do grupo de logs.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Necessária para excluir o filtro de assinatura associado a um grupo de logs.
DescribeDestinations	`logs:DescribeDestinations` Necessária para visualizar todos os destinos associado à conta.
DescribeExportTasks	`logs:DescribeExportTasks` Necessária para visualizar todas as tarefas de exportação associadas à conta.
DescribeLogGroups	`logs:DescribeLogGroups` Necessária para visualizar todos os grupos de logs associados à conta.
DescribeLogStreams	`logs:DescribeLogStreams` Necessária para visualizar todos os streams de logs associados a um grupo de logs.
DescribeMetricFilters	`logs:DescribeMetricFilters` Necessária para visualizar todas as métricas associadas a um grupo de logs.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Necessário para ver a lista de definições de consulta salvas no CloudWatch Logs Insights.
DescribeQueries	`logs:DescribeQueries` Necessária para ver a lista de consultas do CloudWatch Logs Insights que estão agendadas, em execução ou foram executadas recentemente.

CloudWatch Logs Operações de API	Permissões necessárias (Ações da API):
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Necessário para exibir uma lista de políticas de recursos do CloudWatch Logs.
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Necessária para visualizar todos os filtros de assinatura associados a um grupo de logs.
FilterLogEvents	`logs:FilterLogEvents` Necessária para classificar eventos de log por padrão de filtros de grupos.
GetLogEvents	`logs:GetLogEvents` Necessária para recuperar eventos de log a partir de um stream de logs.
GetLogGroupFields	`logs:GetLogGroupFields` Necessária para recuperar a lista de campos incluídos nos eventos de log em um grupo de log.
GetLogRecord	`logs:GetLogRecord` Necessário para recuperar os detalhes de um único evento de log.
GetQueryResults	`logs:GetQueryResults` Necessária para recuperar os resultados das consultas do CloudWatch Logs Insights.
ListTagsLogGroup	`logs:ListTagsLogGroup` Necessária para listar as tags associadas a um grupo de log.
PutDestination	`logs:PutDestination` Necessária para criar ou atualizar um fluxo de log de destino (como um fluxo do Kinesis).
PutDestinationPolicy	`logs:PutDestinationPolicy` Necessária para criar ou atualizar uma política de acesso associada a um destino de log existente.
PutLogEvents	`logs:PutLogEvents` Necessária para carregar um lote de eventos de log para um stream de log.
PutMetricFilter	`logs:PutMetricFilter` Necessária para criar ou atualizar um filtro de métrica e associá-lo a um grupo de logs.

CloudWatch Logs Operações de API	Permissões necessárias (Ações da API):
PutQueryDefinition	`logs:PutQueryDefinition` Necessária para salvar uma consulta no CloudWatch Logs Insights.
PutResourcePolicy	`logs:PutResourcePolicy` Necessário para criar uma política de recursos do CloudWatch Logs.
PutRetentionPolicy	`logs:PutRetentionPolicy` Necessária para definir o número de dias nos quais manter os eventos de log (retenção) em um grupo de logs.
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Necessária para criar ou atualizar um filtro de assinatura e associá-lo a um grupo de logs.
StartQuery	`logs:StartQuery` Necessário para iniciar consultas do CloudWatch Logs Insights.
StopQuery	`logs:StopQuery` Necessária para interromper uma consulta do CloudWatch Logs Insights que está em andamento.
TagLogGroup	`logs:TagLogGroup` Obrigatório para adicionar ou atualizar as tags do grupo de logs.
TestMetricFilter	`logs:TestMetricFilter` Necessária para testar um padrão de filtro em relação a uma amostra de mensagens de eventos de log.

Usar funções vinculadas ao serviço do CloudWatch Logs

O Amazon CloudWatch Logs usa funções vinculadas ao serviço do AWS Identity and Access Management (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao CloudWatch Logs. As funções vinculadas a serviços são predefinidas pelo CloudWatch Logs e incluem todas as permissões que o serviço requer para chamar outros serviços da AWS em seu nome.

Uma função ligada ao serviço torna a configuração CloudWatch Logs mais eficiente porque não é necessário adicionar manualmente as permissões necessárias. CloudWatch Logs define as permissões das suas funções ligadas ao serviço e, salvo definição em contrário, apenas CloudWatch Logs pode

assumir essas funções. As permissões definidas incluem a política de confiança e a política de permissões. A política de permissões não pode ser anexada a qualquer outro IAM entidade.

Para obter informações sobre outros produtos que oferecem suporte a funções vinculadas a serviços, consulte Produtos da AWS que funcionam com o IAM. Procure os serviços que têm Yes na coluna Função vinculada ao serviço. Escolha um Sim com um link para exibir a documentação da função vinculada a serviço desse serviço.

Permissões da função vinculada ao serviço para o CloudWatch Logs

CloudWatch Logs utiliza a função de serviço associada perigosserviceroleforlogdelivery. CloudWatch Logs utiliza esta função ligada ao serviço para escrever registos diretamente para Kinesis Data Firehose. Para obter mais informações, consulte Enviar registos diretamente para Amazon S3 ou Kinesis Data Firehose (p. 113).

O perigosserviceroleforlogdelivery função relacionada com o serviço confia nos seguintes serviços para assumir a função:

- `CloudWatch Logs`

A política de permissões da função permite que o CloudWatch Logs conclua as seguintes ações nos recursos especificados:

- Ação: `firehose:PutRecord` e `firehose:PutRecordBatch` em todos Kinesis Data Firehose com uma etiqueta com um `LogDeliveryEnabled` com um valor de `True`. Esta etiqueta está automaticamente ligada a um Kinesis Data Firehose quando criar uma subscrição para entregar os registos Kinesis Data Firehose.

Tem de configurar permissões para permitir uma IAM entidade para criar, editar ou eliminar uma função ligada ao serviço. Esta entidade pode ser um utilizador, grupo ou função. Para obter mais informações, consulte Permissões da função vinculada ao serviço no Guia do usuário do IAM.

Criação de uma função vinculada a um serviço do CloudWatch Logs

Não é necessário criar manualmente uma função ligada ao serviço. Quando configurar os registos para ser enviado directamente para um Kinesis Data Firehose no Console de gerenciamento da AWS, o AWS CLI, ou o AWS API, CloudWatch Logs cria a função de serviço para si.

Se você excluir essa função vinculada ao serviço e precisar criá-la novamente, poderá usar esse mesmo processo para recriar a função em sua conta. Quando configurar novamente os registos para ser enviado directamente para um Kinesis Data Firehose fluxo, CloudWatch Logs cria a função de serviço para si novamente.

Edição de uma função vinculada ao serviço do CloudWatch Logs

CloudWatch Logs não permite editar perigosserviceroleforlogdelivery, ou qualquer outra função ligada ao serviço, depois de a criar. Não pode alterar o nome da função porque várias entidades podem referenciar a função. No entanto, você poderá editar a descrição da função usando o IAM. Para obter mais informações, consulte Edição de uma função vinculada a serviço no Guia do usuário do IAM.

Exclusão de uma função vinculada ao serviço do CloudWatch Logs

Se você não precisar mais usar um recurso ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de exclui-la manualmente.

> Note
>
> Se o serviço CloudWatch Logs estiver usando a função quando você tenta excluir os recursos, a exclusão poderá falhar. Se isso acontecer, aguarde alguns minutos e tente a operação novamente.

Para eliminar CloudWatch Logs recursos utilizados pelo AWSServiceRoleForLogDelivery função associada ao serviço

- Parar de enviar os registos diretamente para Kinesis Data Firehose correntes.

Para eliminar manualmente a função ligada ao serviço utilizando IAM

Utilize o IAM consola, a AWS CLI, ou o AWS API para eliminar o perigosserviceroleforlogdelivery função associada ao serviço. Para mais informações, consulte Eliminar uma função ligada ao serviço

Regiões com suporte a funções vinculadas a serviço do CloudWatch Logs

O CloudWatch Logs oferece suporte a funções vinculadas ao serviço em todas as regiões da AWS em que o serviço esteja disponível. Para obter mais informações, consulte Regiões e endpoints do CloudWatch Logs.

Validação de conformidade do Amazon CloudWatch Logs

Auditores independentes avaliam a segurança e a conformidade do Amazon CloudWatch Logs como parte de vários programas de conformidade da AWS. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

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

Você pode fazer download de relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte Fazer download de relatórios no AWS Artifact.

Sua responsabilidade com relação à conformidade ao usar o Amazon CloudWatch Logs é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade da empresa e pelas leis e regulamentos aplicáveis. A AWS fornece os seguintes recursos para ajudar com a conformidade:

- Guias Quick Start de segurança e conformidade – esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em conformidade e segurança na AWS.
- Whitepaper Arquitetura para segurança e conformidade com HIPAA – esse whitepaper descreve como as empresas podem usar a AWS para criar aplicativos em conformidade com a HIPAA.
- Recursos de conformidade da AWS – esta coleção de manuais e guias pode ser aplicada ao seu setor e local.

- Avaliar recursos com regras no AWS Config Developer Guide – AWS Config; avalia como as configurações de recursos estão em conformidade com as práticas internas, diretrizes do setor e regulamentos.
- AWS Security Hub – esse serviço da AWS fornece uma visão abrangente do estado da segurança na AWS que ajuda você a verificar sua conformidade com padrões e melhores práticas de segurança do setor.

Resiliência no Amazon CloudWatch Logs

A infraestrutura global da AWS é criada com base em regiões e zonas de disponibilidade da AWS. As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, altas taxas de transferência e redes altamente redundantes. Com as zonas de disponibilidade, você pode projetar e operar aplicativos e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte Infraestrutura global da AWS.

Segurança da infraestrutura no Amazon CloudWatch Logs

Como um serviço gerenciado, o Amazon CloudWatch Logs é protegido pelos procedimentos de segurança da rede global da AWS que estão descritos no whitepaper Amazon Web Services: visão geral dos processos de segurança.

Use chamadas de API publicadas pela AWS para acessar o Amazon CloudWatch Logs por meio da rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos TLS 1.2 ou posterior. Os clientes também devem ter suporte a pacotes de criptografia com sigilo de encaminhamento perfeito (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece suporte a 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.

Usar o CloudWatch Logs com VPC endpoints de interface

Se você usa a Amazon Virtual Private Cloud (Amazon VPC) para hospedar seus recursos da AWS, pode estabelecer uma conexão privada entre a VPC e o CloudWatch Logs. Você pode usar essa conexão para enviar logs para o CloudWatch Logs sem enviá-los por meio da Internet.

A Amazon VPC é um serviço da AWS que você pode usar para executar os recursos da AWS em uma rede virtual definida por você. Com a VPC, você tem controle sobre as configurações de rede, como o intervalo de endereços IP, sub-redes, tabelas de rotas e gateways de rede. Para conectar sua VPC ao CloudWatch Logs, você define um VPC endpoint de interface para o CloudWatch Logs. Esse tipo

de endpoint permite que você conecte a VPC aos serviços da AWS. O endpoint fornece conectividade confiável e dimensionável com o CloudWatch Logs sem a necessidade de um gateway de Internet, da instância de conversão de endereço de rede (NAT) ou de uma conexão VPN. Para obter mais informações, consulte O que é o Amazon VPC no Guia do usuário da Amazon VPC.

Os VPC endpoints de interface são desenvolvidos pelo AWS PrivateLink, uma tecnologia da AWS que permite a comunicação privada entre os serviços da AWS usando uma interface de rede elástica com endereços IP privados. Para obter mais informações, consulte Novo – AWS PrivateLink para serviços da AWS.

As etapas a seguir são para usuários da Amazon VPC. Para obter mais informações, consulte Conceitos básicos no Guia do usuário da Amazon VPC.

Availability

CloudWatch LogsNo momento, o oferece suporte a VPC endpoints nas seguintes regiões:

- Leste dos EUA (Ohio)
- Leste dos EUA (Norte da Virgínia)
- Oeste dos EUA (Norte da Califórnia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Hong Kong)
- Ásia Pacífico (Mumbai)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Cingapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Paris)
- América do Sul (São Paulo)
- AWS GovCloud (Leste dos EUA)
- AWS GovCloud (US-West)

Criar um VPC endpoint para o CloudWatch Logs

Para começar a usar o CloudWatch Logs com sua VPC, crie um VPC endpoint de interface para o CloudWatch Logs. O serviço a ser escolhido é com.amazonaws.***Region***.logs Você não precisa alterar nenhuma configuração do CloudWatch Logs. Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

Como testar a conexão entre a VPC e o CloudWatch Logs

Depois de criar o endpoint, você pode testar a conexão.

Para testar a conexão entre a VPC e o endpoint do CloudWatch Logs

1. Conecte-se a uma instância do Amazon EC2 que reside na VPC. Para obter informações sobre como se conectar, consulte Conectar-se à sua instância do Linux ou Conectar-se à sua instância do Windows na documentação do Amazon EC2.
2. Na instância, use a AWS CLI para criar uma entrada de log em um de seus grupos de logs existentes.

   Primeiro, crie um arquivo JSON com um evento de log. O timestamp deve ser especificado como o número de milissegundos após 1º de janeiro de 1970 00:00:00 UTC.

   ```
   [
     {
       "timestamp": 1533854071310,
       "message": "VPC Connection Test"
     }
   ]
   ```

   Em seguida, use o comando `put-log-events` para criar a entrada de log:

   ```
   aws logs put-log-events --log-group-name LogGroupName --log-stream-name LogStreamName
    --log-events file://JSONFileName
   ```

   Se a resposta ao comando incluir `nextSequenceToken`, o comando terá sido bem-sucedido e o VPC endpoint estará funcionando.

Controlar o acesso ao VPC endpoint do CloudWatch Logs

Uma política de endpoint de VPC é uma política de recursos do IAM que você anexa a um endpoint quando cria ou modifica o endpoint. Se você não anexar uma política quando criar um endpoint, anexaremos uma política padrão que permita o acesso total ao serviço. Uma política de endpoint não substitui políticas de usuário do IAM ou políticas de serviço específicas. É uma política separada para controlar o acesso do endpoint ao serviço especificado.

Políticas de endpoint devem ser gravadas em formato JSON.

Para obter mais informações, consulte Controle do acesso a serviços com VPC endpoints no Guia do usuário da Amazon VPC.

Veja a seguir um exemplo de uma política de endpoint para o CloudWatch Logs. Essa política permite que os usuários se conectem ao CloudWatch Logs por meio da VPC para criar logs e enviar streams ao CloudWatch Logs e impede que eles executem outras ações do CloudWatch Logs

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
```

```
}
```

Como modificar a política de VPC endpoint para o CloudWatch Logs

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel de navegação, selecione Endpoints.
3. Se você ainda não tiver criado o endpoint para o CloudWatch Logs, selecione Create Endpoint (Criar endpoint). Em seguida, selecione com.amazonaws.***Region***.logs e escolha Create endpoint.
4. Selecione o arquivo com.amazonaws.***Region***.logs endpoint e escolha a guia Policy na metade inferior da tela.
5. Escolha Edit Policy (Editar política) e faça as alterações na política.

Suporte para chaves de contexto da VPC

O CloudWatch Logs `aws:SourceVpc` oferece suporte às chaves de contexto `aws:SourceVpce` e VPCs que podem limitar o acesso a específicos ou a VPC endpoints específicos. Essas chaves funcionam somente quando o usuário está usando VPC endpoints. Para obter mais informações, consulte Chaves disponíveis para alguns serviços no Guia do usuário do IAM.

Registrar chamadas de API do Amazon CloudWatch Logs no AWS CloudTrail

O Amazon CloudWatch Logs é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações executadas por um usuário, uma função ou um serviço da AWS no CloudWatch Logs. O CloudTrail captura as chamadas de API feitas por ou em nome de sua conta da AWS. As chamadas capturadas incluem as chamadas do console do CloudWatch e as chamadas de código para as operações da API do CloudWatch Logs. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail a um bucket do Amazon S3, incluindo eventos para o CloudWatch Logs. Se não configurar uma trilha, você ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history. Com as informações coletadas pelo CloudTrail, você pode determinar a solicitação feita para o CloudWatch Logs, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre o CloudTrail, incluindo como configurá-lo e habilitá-lo, consulte o AWS CloudTrail User Guide.

Tópicos

Informações sobre o CloudWatch Logs no CloudTrail

O CloudTrail está habilitado na sua conta da AWS ao criá-la. Quando ocorre a atividade do evento com suporte no CloudWatch Logs, ela é registrada em um evento do CloudTrail junto com outros eventos de serviços da AWS no Event history (Histórico de eventos). Você pode visualizar, pesquisar e fazer download de eventos recentes em sua conta da AWS. Para obter mais informações, consulte Visualizar eventos com o histórico de eventos do CloudTrail.

Para obter um registro contínuo de eventos em sua conta da AWS, incluindo eventos para o CloudWatch Logs, crie uma trilha. Uma trilha permite CloudTrail para fornecer arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões da AWS. A trilha registra eventos de todas as regiões na partição da AWS e fornece os arquivos de log para o bucket do Amazon S3 que você especificar. Além disso, é possível configurar outros serviços da AWS para analisar mais profundamente e agir sobre os dados de evento coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- CloudTrail Serviços compatíveis e integrações do
- Configuração de notificações do Amazon SNS para o CloudTrail
- Recebimento de arquivos de log do CloudTrail de várias regiões e Recebimento de arquivos de log do CloudTrail de várias contas

O CloudWatch Logs oferece suporte ao registro em log das ações a seguir como eventos nos arquivos de log do CloudTrail:

- CancelExportTask
- CreateExportTask
- CreateLogGroup
- CreateLogStream
- DeleteDestination
- DeleteLogGroup
- DeleteLogStream
- DeleteMetricFilter
- DeleteRetentionPolicy
- DeleteSubscriptionFilter
- PutDestination
- PutDestinationPolicy
- PutMetricFilter
- PutRetentionPolicy
- PutSubscriptionFilter
- StartQuery
- StopQuery
- TestMetricFilter

Somente elementos de solicitação são registrados no CloudTrail para estas ações de API do CloudWatch Logs:

- DescribeDestinations
- DescribeExportTasks
- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries
- DescribeSubscriptionFilters
- GetLogGroupFields
- GetLogRecord

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 da raiz ou do AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail.

Noções básicas das entradas dos arquivos de log

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log em um bucket do Amazon S3 que você especificar. Os arquivos de log do CloudTrail contêm uma ou mais entradas

de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e hora da ação, parâmetros de solicitação, e assim por diante. arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas da API pública. Assim, elas não são exibidas em nenhuma ordem específica.

A entrada do arquivo de log a seguir mostra que um usuário chamou a ação CreateExportTask do CloudWatch Logs.

```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

Referência do agente do CloudWatch Logs

> **Important**
>
> Esta referência é para o mais antigo CloudWatch Logs agente, que está no caminho para a desmarcação. Recomendamos vivamente que utilize o CloudWatch em vez de. Para mais informações sobre esse agente, consulte Recolher métricas e registos de Amazon EC2 e servidores nas instalações com o CloudWatch agente.

O agente do CloudWatch Logs fornece uma forma automatizada de enviar dados de log para o CloudWatch Logs a partir de instâncias do Amazon EC2. O agente inclui os seguintes componentes:

- Um plug-in para a AWS CLI que envia dados de log para o CloudWatch Logs.
- Um script (daemon) que inicia o processo para enviar dados para o CloudWatch Logs.
- Um trabalho cron que garante que o daemon esteja sempre em execução.

Arquivo de configuração do agente

O arquivo de configuração do agente do CloudWatch Logs descreve informações necessárias ao agente do CloudWatch Logs. A seção [general] do arquivo de configuração do agente define configurações comuns que se aplicam a todos os streams de log. A seção [logstream] define as informações necessárias para enviar um arquivo local para um stream de logs remoto. Você pode ter mais de uma seção [logstream], mas cada uma deve ter um nome exclusivo dentro do arquivo de configuração, por exemplo, [logstream1], [logstream2] e assim por diante. O valor [logstream] junto com a primeira linha de dados no arquivo de log definem a identidade do arquivo de log.

```
[general]
state_file = value
logging_config_file = value
use_gzip_http_content_encoding = [true | false]

[logstream1]
log_group_name = value
log_stream_name = value
datetime_format = value
time_zone = [LOCAL|UTC]
file = value
file_fingerprint_lines = integer | integer-integer
multi_line_start_pattern = regex | {datetime_format}
initial_position = [start_of_file | end_of_file]
encoding = [ascii|utf_8|..]
buffer_duration = integer
batch_count = integer
batch_size = integer

[logstream2]
...
```

state_file

Especifica onde o arquivo de estado está armazenado.

logging_config_file

(Opcional) Especifica a localização do arquivo de configuração de log do agente. Se você não especifica um arquivo de configuração de log do agente aqui, o arquivo padrão awslogs.conf é usado. A localização do arquivo padrão é `/var/awslogs/etc/awslogs.conf` se você instalou o agente com um script, e `/etc/awslogs/awslogs.conf` se você instalou o agente com RPM. O arquivo está no formato de arquivo de configuração Python (https://docs.python.org/2/library/logging.config.html#logging-config-fileformat). Registradores com os seguintes nomes podem ser personalizados.

```
cwlogs.push
cwlogs.push.reader
cwlogs.push.publisher
cwlogs.push.event
cwlogs.push.batch
cwlogs.push.stream
cwlogs.push.watcher
```

O exemplo a seguir altera o nível de leitor e editor para AVISO enquanto o valor padrão é INFO.

```
[loggers]
keys=root,cwlogs,reader,publisher

[handlers]
keys=consoleHandler

[formatters]
keys=simpleFormatter

[logger_root]
level=INFO
handlers=consoleHandler

[logger_cwlogs]
level=INFO
handlers=consoleHandler
qualname=cwlogs.push
propagate=0

[logger_reader]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.reader
propagate=0

[logger_publisher]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.publisher
propagate=0

[handler_consoleHandler]
class=logging.StreamHandler
level=INFO
formatter=simpleFormatter
args=(sys.stderr,)

[formatter_simpleFormatter]
```

```
format=%(asctime)s - %(name)s - %(levelname)s - %(process)d - %(threadName)s -
 %(message)s
```

use_gzip_http_content_encoding

Quando definido como verdadeiro (padrão), permite que a codificação de conteúdo gzip http envie cargas úteis compactadas para o CloudWatch Logs. Isso reduz o uso da CPU, reduz o NetworkOut e diminui a latência. Para desativar este recurso, adicione use_gzip_http_content_encoding = false à seção [general] do arquivo de configuração do agente do CloudWatch Logs e, em seguida, reinicie o agente.

Note

Essa configuração só está disponível no awscli-cwlogs versão 1.3.3 e posterior.

log_group_name

Especifica o grupo de logs de destino. Um grupo de logs é criado automaticamente, caso ele ainda não exista. Os nomes dos grupos de log podem ter entre 1 e 512 caracteres. Os caracteres permitidos incluem a-z, A-Z, 0-9, "_" (sublinhado), "-" (hífen), "/" (barra) e "." (ponto).

log_stream_name

Especifica o stream de logs de destino. Você pode usar uma cadeia de caracteres literal ou as variáveis predefinidas ({instance_id}, {hostname}, {ip_address}) ou uma combinação de ambas, para definir um nome de stream de log. Um stream de logs é criado automaticamente, caso ele ainda não exista.

datetime_format

Especifica como o carimbo de data e hora é extraído de logs. O timestamp é usado para recuperar eventos de log e gerar métricas. A hora atual será usada para todos os eventos de log se datetime_format não for fornecido. Se o valor datetime_format fornecido for inválido para uma determinada mensagem de log, o carimbo de data e hora do último evento de log com um carimbo de data/hora analisado com êxito será usado. Se não existir nenhum evento de log anterior, a hora atual será usada.

Os códigos datetime_format comuns estão listados a seguir. Você também pode usar qualquer código datetime_format compatíveis com Python, datetime.strptime(). O desvio de fuso horário (%z) também é compatível, embora não seja suportado até o python 3.2, [+ -] HHMM sem dois-pontos (:). Para obter mais informações, consulte strftime() and strptime() Behavior.

y Ano sem o século como um número decimal preenchido com zero. 00, 01, ..., 99

%Y : ano com século como número decimal.1970, 1988 2001, 2013

b Mês como nome abreviado do locale. Jan, Fev, ..., Dez (en_US);

%B Mês como nome completo do locale. Janeiro, fevereiro,..., dezembro (en_US);

%-m mês como um número decimal preenchido com zeros 01, 02, ..., 12

%-d dia do mês como um número decimal preenchido com zeros 01, 02, ..., 31

%H Hora (relógio de 24 horas) como um número decimal de formato zero. 00, 01, ..., 23

%I: Hora (relógio de 12 horas) como um número decimal de formato zero. 01, 02, ..., 12

<p> : equivalente de local de AM ou PM.

%-M Minuto como um número decimal com formato zero. 00, 01, ..., 59

%-S Segundo como um número decimal de formato zero. 00, 01, ..., 59

f Microsegundo como número decimal, almofadado zero à esquerda. 000000, ..., 999999

%z Desvio UTC no formulário+HHMM ou -HHMM. +0000, -0400, +1030

Exemplo de formatos:

```
Syslog: '%b %d %H:%M:%S', e.g. Jan 23 20:59:29

Log4j: '%d %b %Y %H:%M:%S', e.g. 24 Jan 2014 05:00:00

ISO8601: '%Y-%m-%dT%H:%M:%S%z', e.g. 2014-02-20T05:20:20+0000
```

time_zone

Especifica o fuso horário do carimbo de data e hora do evento de log. Os dois valores suportados são UTC e LOCAL. O padrão é LOCAL, que será usado se o fuso horário não puder ser considerado com base em datetime_format.

--file

Especifica os arquivos de log que você deseja enviar para o CloudWatch Logs. O arquivo pode apontar para um arquivo específico ou vários arquivos (usando curingas como /var/log/system.log*). Somente o arquivo mais recente é enviado ao CloudWatch Logs com base no tempo de modificação do arquivo. Recomendamos usar caracteres curinga para especificar uma série de arquivos do mesmo tipo, como access_log.2014-06-01-01, access_log.2014-06-01-02, etc., mas não vários tipos de arquivos, como access_log_80 e access_log_443. Para especificar vários tipos de arquivos, adicione outra entrada de stream de log ao arquivo de configuração para que cada tipo de arquivo de log vá para um stream de log diferente. Arquivos compactados não têm suporte.

file_fingerprint_lines

Especifica o intervalo de linhas para identificar um arquivo. Os valores válidos são um ou dois números delimitados por traço, como "1", "2-5". O valor padrão é "1" para que a primeira linha seja usada para calcular a impressão digital. As linhas de impressão digital não são enviadas ao CloudWatch Logs a menos que todas as linhas especificadas estejam disponíveis.

multi_line_start_pattern

Especifica o padrão para identificar o início de uma mensagem de log. Uma mensagem de log é feita de uma linha em conformidade com o padrão e as seguintes linhas que não correspondem ao padrão. Os valores válidos são expressão regular ou {datetime_format}. Ao usar {datetime_format}, a opção datetime_format deve ser especificada. O valor padrão é "^ [^\s]" para que qualquer linha que comece com um caractere diferente de espaço feche a mensagem de log anterior e inicie uma nova mensagem de log.

initial_position

Especifica onde começar a ler dados (start_of_file ou end_of_file). O padrão é start_of_file. É usado somente se não há estado persistente para esse stream de logs.

encoding

Especifica a codificação do arquivo de log para que o arquivo possa ser lido corretamente. O padrão é utf_8. As codificações suportadas pelo codecs.decode () Python podem ser usadas aqui.

> Warning
>
> A especificação incorreta da codificação pode causar a perda de dados porque os caracteres que não podem ser decodificados são substituídos por algum outro caractere.

Veja a seguir algumas codificações comuns:

```
ascii, big5, big5hkscs, cp037, cp424, cp437, cp500, cp720, cp737, cp775,
cp850, cp852, cp855, cp856, cp857, cp858, cp860, cp861, cp862, cp863, cp864,
```

```
cp865, cp866, cp869, cp874, cp875, cp932, cp949, cp950, cp1006, cp1026,
cp1140, cp1250, cp1251, cp1252, cp1253, cp1254, cp1255, cp1256, cp1257,
cp1258, euc_jp, euc_jis_2004, euc_jisx0213, euc_kr, gb2312, gbk, gb18030,
hz, iso2022_jp, iso2022_jp_1, iso2022_jp_2, iso2022_jp_2004, iso2022_jp_3,
iso2022_jp_ext, iso2022_kr, latin_1, iso8859_2, iso8859_3, iso8859_4,
iso8859_5, iso8859_6, iso8859_7, iso8859_8, iso8859_9, iso8859_10,
iso8859_13, iso8859_14, iso8859_15, iso8859_16, johab, koi8_r, koi8_u,
mac_cyrillic, mac_greek, mac_iceland, mac_latin2, mac_roman, mac_turkish,
ptcp154, shift_jis, shift_jis_2004, shift_jisx0213, utf_32, utf_32_be,
utf_32_le, utf_16, utf_16_be, utf_16_le, utf_7, utf_8, utf_8_sig
```

buffer_duration

Especifica o intervalo de tempo para o processamento em lote de eventos de log. O valor mínimo é 5000 ms e valor padrão é 5000 ms.

batch_count

Especifica o número máximo de eventos de log em um lote, até 10.000. O valor padrão é 10000.

batch_size

Especifica o tamanho máximo de eventos de log em um lote, em bytes, até 1.048.576 bytes. O valor de padrão é de 1048576 bytes. Esse tamanho é calculado como a soma de todas as mensagens de evento em UTF-8, mais 26 bytes para cada evento de log.

Uso do agente do CloudWatch Logs com proxies HTTP

Você pode usar o agente do CloudWatch Logs com proxies HTTP.

Note

Os proxies HTTP são suportados no awslogs-agent-setup.py versão 1.3.8 ou posterior.

Para usar o agente do CloudWatch Logs com proxies HTTP

1. Execute um destes procedimentos:

   a. Para uma nova instalação do agente do CloudWatch Logs, execute os seguintes comandos:

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
   setup.py -O
   ```

   ```
   sudo python awslogs-agent-setup.py --region us-east-1 --http-proxy http://your/
   proxy --https-proxy http://your/proxy --no-proxy 169.254.169.254
   ```

   Para manter o acesso ao serviço de metadados do Amazon EC2 em instâncias do EC2, use --no-proxy 169.254.169.254 (recomendado). Para obter mais informações, consulte Metadados de instância e dados de usuário no Guia do usuário do Amazon EC2 para instâncias do Linux.

   Nos valores de `http-proxy` e `https-proxy`, você especifica a URL completa.

   b. Para uma instalação existente do agente do CloudWatch Logs edite /var/awslogs/etc/proxy.conf e adicione seus proxies:

   ```
   HTTP_PROXY=
   ```

```
HTTPS_PROXY=
NO_PROXY=
```

2. Reinicie o agente para que as alterações sejam aplicadas:

```
sudo service awslogs restart
```

Se você está usando o Amazon Linux 2, use o seguinte comando para reiniciar o agente:

```
sudo service awslogsd restart
```

Compartimentalização de arquivos de configuração do agente do CloudWatch Logs

Se você estiver usando o awslogs-agent-setup.py versão 1.3.8 ou posterior com o awscli-cwlogs 1.3.3 ou posterior, poderá importar diferentes configurações de fluxo para vários componentes de forma independente entre si por meio da criação de arquivos de configuração adicionais no diretório /var/awslogs/etc/config/. Quando o agente do CloudWatch Logs é iniciado, ele inclui todas as configurações de fluxo nesses arquivos de configuração adicionais. As propriedades de configuração na seção [general] devem ser definidas no arquivo de configuração principal (/var/awslogs/etc/awslogs.conf) e são ignoradas em todos os arquivos de configuração adicionais encontrados em /var/awslogs/etc/config/.

Se você não tem um diretório /var/awslogs/etc/config/, pois o agente foi instalado com RPM, use o /etc/awslogs/config/ no lugar dele.

Reinicie o agente para que as alterações sejam aplicadas:

```
sudo service awslogs restart
```

Se você está usando o Amazon Linux 2, use o seguinte comando para reiniciar o agente:

```
sudo service awslogsd restart
```

Perguntas frequentes sobre o agente do CloudWatch Logs

Quais tipos de rotações de arquivos são compatíveis?

Os mecanismos de rotação de arquivos a seguir são suportados:

- Renomear arquivos de log existentes com um sufixo numérico e, em seguida, recriar o arquivo de log original em branco. Por exemplo, /var/log/syslog.log é renomeado como /var/log/syslog.log.1. Se /var/log/syslog.log.1 já existir de uma rotação anterior, ele será renomeado como /var/log/syslog.log.2.
- Truncando o arquivo de log original após a criação de uma cópia. Por exemplo, /var/log/syslog.log é copiado em /var/log/syslog.log.1 e /var/log/syslog.log é truncado. Pode haver perda de dados nesse caso, então, tome cuidado ao usar esse mecanismo de rotação de arquivos.
- Criando um novo arquivo com um padrão comum como o antigo. Por exemplo, /var/log/syslog.log.2014-01-01 permanece e /var/log/syslog.log.2014-01-02 é criado.

A impressão digital (ID de origem) do arquivo é calculada aplicando hash à chave de stream de logs e à primeira linha do conteúdo do arquivo. Para substituir esse comportamento, a opção file_fingerprint_lines pode ser usada. Quando a rotação de arquivos acontece, o novo arquivo deve ter um novo conteúdo e o arquivo antigo não deve ter conteúdo anexado; o agente envia o novo arquivo depois de ler o arquivo antigo.

Como posso determinar qual versão do agente estou usando?

Se você tiver usado um script de configuração para instalar o agente do CloudWatch Logs, poderá usar o /var/awslogs/bin/awslogs-version.sh para verificar qual versão do agente você está usando. Ele imprime a versão do agente e suas principais dependências. Se você tiver usado o yum para instalar o agente do CloudWatch Logs, poderá usar "yum info awslogs" e "yum info aws-cli-plugin-cloudwatch-logs" para verificar a versão do agente do CloudWatch Logs e o plug-in.

Como ficam as entradas de log convertidas em eventos de log?

Os eventos de log contêm duas propriedades: o carimbo de data e hora de quando o evento ocorreu, e a mensagem de log bruta. Por padrão, qualquer linha que comece com um caractere diferente de espaço fecha a mensagem de log anterior, se houver, e inicia uma nova mensagem de log. Para substituir este comportamento, o multi_line_start_pattern pode ser usado, e qualquer linha que esteja em conformidade com o padrão inicia uma nova mensagem de log. O padrão pode ser qualquer regex ou "{datetime_format}". Por exemplo, se a primeira linha de cada mensagem de log contiver um carimbo de data e hora, como '2014-01-02T13:13:01Z', o multi_line_start_pattern poderá ser definido como '\d\{4\}-\d\{2\}-\d\{2\}T\d\{2\}:\d\{2\}:\d\{2\}Z'. Para simplificar a configuração, a variável ‘{datetime_format}' poderá ser usada se a variável datetime_format option for especificada. Para o mesmo exemplo, se datetime_format estiver definido como '%Y-%m-%dT%H:%M:%S%z', então multi_line_start_pattern poderá ser simplesmente '{datetime_format}'.

A hora atual será usada para todos os eventos de log se datetime_format não for fornecido. Se datetime_format fornecido for inválido para uma determinada mensagem de log, será usado o carimbo de data e hora do último evento de log com um carimbo de data/hora analisado com êxito. Se não existir nenhum evento de log anterior, a hora atual será usada. Uma mensagem de aviso é registrada quando um evento de log retorna para a hora atual ou a hora do evento de log anterior.

Os carimbos de data e hora são usados para recuperar eventos de log e gerar métricas, portanto, se você especificar o formato incorreto, os eventos de log poderão se tornar não recuperáveis e gerar métricas incorretas.

Como os eventos de log são armazenadas em lote?

Um lote fica cheio e é publicado quando qualquer uma das seguintes condições é atendida:

1. O tempo de buffer_duration terminou desde que o primeiro evento de log foi adicionado.
2. Menos do que batch_size de eventos de log foram acumulados, mas adicionar o novo evento de log excede o batch_size.
3. O número de eventos de log atingiu batch_count.
4. Os eventos de log do lote não abrangem mais do que 24 horas, mas adicionar o novo evento de log excede a restrição de 24 horas.

O que faz com que entradas de log, eventos de log ou lotes sejam ignorados ou truncados?

Para acompanhar a restrição da operação `PutLogEvents`, os seguintes problemas podem fazer com que um evento de log ou lote seja ignorado.

> **Note**
>
> O agente do CloudWatch Logs grava um aviso em seu log quando os dados são ignorados.

1. Se o tamanho de um evento de log exceder 256 KB, ele será ignorado completamente.
2. Se o carimbo de data e hora do evento de log for de mais do que 2 horas no futuro, o evento de log será ignorado.

3. Se o carimbo de data e hora do evento de log for de mais do que 14 dias no passado, o evento de log será ignorado.
4. Se qualquer evento de log for mais antigo do que o período de retenção do grupo de logs, o lote inteiro será ignorado.
5. Se o lote de eventos de log em uma única solicitação `PutLogEvents` abranger mais de 24 horas, ocorrerá falha na operação `PutLogEvents`.

A interrupção do agente causa perda/duplicações de dados?

Não, desde que o arquivo de estado esteja disponível e nenhuma rotação de arquivos tenha ocorrido desde a última execução. O agente do CloudWatch Logs pode começar de onde parou e continuar a enviar os dados de log.

Posso apontar arquivos de log diferentes do mesmo host ou de hosts diferentes para o mesmo stream de logs?

A configuração de várias fontes de log para enviar dados a um único stream de logs não é suportada.

Quais chamadas de API o agente pode fazer (ou quais ações devo adicionar à minha política do IAM)?

O CloudWatch Logs agente requer o `CreateLogGroup`, `CreateLogStream`, `DescribeLogStreams`, e `PutLogEvents` operações. Se você estiver usando o agente mais recente, o `DescribeLogStreams` não será necessário. Consulte a política de exemplo do IAM a seguir.

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

Não quero que o agente do CloudWatch Logs crie grupos de log nem fluxos de log automaticamente. Como posso impedir que o agente recrie grupos e fluxos de log?

No seu IAM política, pode restringir o agente apenas às seguintes operações: `DescribeLogStreams`, `PutLogEvents`.

Antes de revogar as permissões `CreateLogGroup` e `CreateLogStream` do agente, certifique-se de criar os grupos de log e fluxos de log que você deseja que o agente use. O agente de logs não pode criar fluxos de log em um grupo de log que você criou, a menos que ele tenha as permissões `CreateLogGroup` e `CreateLogStream`.

Quais logs devo procurar ao solucionar problemas?

O log de instalação do agente está em `/var/log/awslogs-agent-setup.log` e o log do agente está em `/var/log/awslogs.log`.

Monitoramento do uso com métricas do CloudWatch

O CloudWatch Logs envia métricas para o Amazon CloudWatch a cada minuto.

CloudWatch LogsMétricas do

O namespace `AWS/Logs` inclui as métricas a seguir.

Métrica	Descrição
`IncomingBytes`	O volume de eventos de log em bytes descompactados enviados para o CloudWatch Logs. Quando usado com a dimensão `LogGroupName`, este é o volume de eventos de log em bytes descompactados carregados no grupo de logs. Dimensões válidas: LogGroupName Estatística válida: soma Unidades: bytes
`IncomingLogEvents`	O número de eventos de log carregados no CloudWatch Logs. Quando usado com a dimensão `LogGroupName`, esse é o número de eventos de log carregados no grupo de logs. Dimensões válidas: LogGroupName Estatística válida: soma Unidades: nenhuma
`ForwardedBytes`	O volume de eventos de log em bytes compactados encaminhados para o destino de inscrição. Dimensões válidas: LogGroupName, DestinationType, FilterName Estatística válida: soma Unidades: bytes
`ForwardedLogEvents`	O número de eventos de log encaminhados para o destino de inscrição. Dimensões válidas: LogGroupName, DestinationType, FilterName Estatística válida: soma Unidades: nenhuma

Métrica	Descrição
`DeliveryErrors`	O número de eventos de log para os quais CloudWatch Logs recebeu um erro ao encaminhar dados para o destino da inscrição. Dimensões válidas: LogGroupName, DestinationType, FilterName Estatística válida: soma Unidades: nenhuma
`DeliveryThrottling`	O número de eventos de log para os quais CloudWatch Logs foi acumulado ao encaminhar dados para o destino da inscrição. Dimensões válidas: LogGroupName, DestinationType, FilterName Estatística válida: soma Unidades: nenhuma

Dimensões para métricas do CloudWatch Logs

As dimensões que você pode usar com as métricas do CloudWatch Logs são listadas abaixo.

Dimensão	Descrição
`LogGroupName`	O nome do grupo de logs do CloudWatch Logs para os quais exibir métricas.
`DestinationType`	O destino da inscrição para os dados do CloudWatch Logs, que pode ser AWS Lambda, Amazon Kinesis Data Streams ou Amazon Kinesis Data Firehose.
`FilterName`	O nome do filtro de inscrição que está encaminhando dados do grupo de logs para o destino. O filtro de inscrição é automaticamente convertido pelo CloudWatch para ASCII e os caracteres sem suporte são substituídos por um ponto de interrogação (?).

CloudWatch LogsCotas do

CloudWatch LogsO tem as seguintes cotas:

Recurso	Cota padrão
Tamanho do lote	1 MB (máximo). Não é possível alterar esta cota.
Arquivamento de dados	Até 5 GB de arquivamento de dados gratuito. Não é possível alterar esta cota.
CreateLogStream	50 transações por segundo (TPS/conta/região), após as quais as transações são limitadas. É possível solicitar um aumento da cota.
DescribeLogGroups	5 transações por segundo (TPS/conta/região). É possível solicitar um aumento da cota.
DescribeLogStreams	5 transações por segundo (TPS/conta/região). É possível solicitar um aumento da cota.
Campos de log descobertos	CloudWatch Logs InsightsO pode descobrir um máximo de 1.000 campos de eventos de log em um grupo de logs. Não é possível alterar esta cota. Para obter mais informações, consulte Logs compatíveis e campos descobertos (p. 36).
Campos de log extraídos em logs JSON	CloudWatch Logs InsightsO pode extrair um máximo de 100 campos de eventos de log de um log JSON. Não é possível alterar esta cota. Para obter mais informações, consulte Logs compatíveis e campos descobertos (p. 36).
Tamanho do evento	256 KB (máximo). Não é possível alterar esta cota.
Tarefa de exportação	Uma tarefa de exportação ativa (em execução ou pendente) por vez, por conta. Não é possível alterar esta cota.
FilterLogEvents	5 transações por segundo (TPS)/conta/região. Não é possível alterar esta cota.
GetLogEvents	10 solicitações por segundo, por conta, por região. Não é possível alterar esta cota. Se você estiver processando novos dados continuamente, recomendamos assinaturas. Se você precisar de dados históricos, recomendamos exportar os dados para o Amazon S3.
Dados recebidos	Até 5 GB de dados recebidos gratuitos. Não é possível alterar esta cota.

Recurso	Cota padrão
Grupos de logs	1,000,000 grupos de log por conta, por região. É possível solicitar um aumento da cota. Não há cota para o número de streams de log que podem pertencer a um grupo de logs.
Filtros de métrica	100 por grupo de logs. Não é possível alterar esta cota.
Métricas de formato de métrica incorporadas	100 métricas por evento de log e 9 dimensões por métrica. Para obter mais informações sobre o formato de métrica incorporada, consulte Especificação: formato de métrica incorporada no Guia do usuário do Amazon CloudWatch.
PutLogEvents	5 solicitações por segundo por fluxo de logs. Solicitações adicionais são limitadas. Não é possível alterar esta cota. O tamanho máximo do lote de uma solicitação PutLogEvents é 1 MB. 800 transações por segundo por conta e por região, exceto para as regiões a seguir, onde a cota é de 1500 transações por segundo por conta e por região: Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon) e Europa (Irlanda). É possível solicitar um aumento da cota.
Tempo limite de execução da consulta	As consultas no CloudWatch Logs Insights expiram após 15 minutos. Esse limite de tempo não pode ser alterado.
Grupos de logs consultados	Um máximo de 20 grupos de logs pode ser consultado em uma única consulta do CloudWatch Logs Insights. Não é possível alterar esta cota.
Simultaneidade da consulta	Um máximo de 10 consultas do CloudWatch Logs Insights simultâneas, inclusive consultas que foram adicionadas aos painéis. É possível solicitar um aumento da cota.
Disponibilidade dos resultados da consulta	Os resultados de uma consulta podem ser recuperados durante 7 dias. Esse tempo de disponibilidade não pode ser alterado.
Consultar resultados exibidos no console	Por padrão, até 1.000 linhas de resultados de consulta são exibidas no console. É possível usar o comando **`limit`** em uma consulta a fim de aumentar para até 10.000 linhas. Para obter mais informações, consulte CloudWatch Logs InsightsSintaxe da consulta do (p. 41).
Políticas de recursos	Até 10 políticas de recursos do CloudWatch Logs por região por conta. Não é possível alterar esta cota.
Consultas salvas	Você pode salvar até 1.000 consultas do CloudWatch Logs Insights, por região e por conta. Não é possível alterar esta cota.
Filtros de assinatura	2 por grupo de logs. Não é possível alterar esta cota.

Histórico do documento

A tabela a seguir descreve as mudanças importantes em cada versão do Guia de usuário do CloudWatch Logs a partir de junho de 2018. Para receber notificações sobre atualizações dessa documentação, você pode se inscrever em um feed RSS.

update-history-change	update-history-description	update-history-date
CloudWatch Logs Insights lançado (p. 164)	Use o CloudWatch Logs Insights para pesquisar e analisar interativamente os dados de log. Para obter mais informações, consulte Analisar dados de log com o CloudWatch Logs Insights no Amazon CloudWatch Logs User Guide	November 27, 2018
Suporte para endpoints do Amazon VPC (p. 164)	Agora, você pode estabelecer uma conexão privada entre sua VPC e o CloudWatch Logs. Para obter mais informações, consulte Usar CloudWatch Logs com os VPC endpoints de interface no Amazon CloudWatch Logs User Guide.	June 28, 2018

A tabela a seguir descreve as alterações importantes feitas no Guia do usuário do Amazon CloudWatch Logs.

Alteração	Descrição	Data de lançamento
VPC endpoints de interface	Em algumas regiões, você pode usar um VPC endpoint de interface para evitar que o tráfego entre sua Amazon VPC e o CloudWatch Logs saia da rede da Amazon. Para obter mais informações, consulte Usar o CloudWatch Logs com VPC endpoints de interface (p. 145).	7 de março de 2018
Logs de consulta de DNS do Route 53	Você pode usar o CloudWatch Logs para armazenar logs sobre as consultas de DNS recebidas pelo Route 53. Para obter mais informações, consulte O que é Amazon CloudWatch Logs? (p. 1) ou Registro de consultas de DNS no Guia do desenvolvedor do Amazon Route 53.	7 de setembro de 2017
Marcar grupos de logs	Você pode usar marcas para categorizar seus grupos de logs. Para obter mais informações, consulte Marcar grupos de logs no Amazon CloudWatch Logs (p. 62).	13 de dezembro de 2016
Melhorias no console	Você pode navegar de gráficos de métricas para os grupos de logs associados. Para obter mais informações, consulte Passar de métricas para logs (p. 61).	7 de novembro de 2016

Alteração	Descrição	Data de lançamento
Melhorias no uso do console	Aprimorada a experiência para facilitar a pesquisa, a filtragem e a solução de problemas. Por exemplo, agora você pode filtrar seus dados de log para um intervalo de data e hora. Para obter mais informações, consulte Visualizar os dados de log enviados para o CloudWatch Logs (p. 59).	29 de agosto de 2016
Inclusão de suporte ao AWS CloudTrail para Amazon CloudWatch Logs e novas métricas do CloudWatch Logs	Inclusão de suporte ao AWS CloudTrail para o CloudWatch Logs. Para obter mais informações, consulte Registrar chamadas de API do Amazon CloudWatch Logs no AWS CloudTrail (p. 149).	10 de março de 2016
Inclusão de suporte à exportação do CloudWatch Logs para o Amazon S3	Inclusão de suporte à exportação dos dados do CloudWatch Logs para o Amazon S3. Para obter mais informações, consulte Exportação de dados de log para o Amazon S3 (p. 114).	7 de dezembro de 2015
Inclusão de suporte aos eventos registrados do AWS CloudTrail no Amazon CloudWatch Logs	Você pode criar alarmes no CloudWatch e receber notificações de determinada atividade de API conforme capturada pelo CloudTrail e usar a notificação para solução de problemas.	10 de novembro de 2014
Adicionado o suporte para Amazon CloudWatch Logs	Você pode usar o Amazon CloudWatch Logs para monitorar, armazenar e acessar o sistema, o aplicativo e os arquivos de log personalizados em suas instâncias do Amazon Elastic Compute Cloud (Amazon EC2) ou em outras origens. Em seguida, você pode recuperar os dados de log associados do CloudWatch Logs usando o console do Amazon CloudWatch, os comandos do CloudWatch Logs na interface da linha de comando da AWS ou o SDK do CloudWatch Logs. Para obter mais informações, consulte O que é Amazon CloudWatch Logs? (p. 1).	10 de julho de 2014

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the AWS General Reference.

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.