aws

Guia do usuário

CloudWatch Registros da Amazon

CloudWatch Registros da Amazon: Guia do usuário

Table of Contents

O que é o Amazon CloudWatch Logs? 1
Atributos 1
AWS Serviços relacionados 3
Definição de preço 4
Conceitos 4
Faturamento e custos 5
Classes de log 7
Atributos compatíveis 7
Conceitos básicos 10
Pré-requisitos 10
Inscreva-se para um Conta da AWS 10
Criar um usuário com acesso administrativo 11
Configurar a interface de linha de comando 12
Usando o CloudWatch agente unificado 13
Usando o CloudWatch agente anterior 13
CloudWatch Pré-requisitos do agente de registros 14
Início rápido: Instalar o agente em uma instância do EC2 do Linux em execução 15
Início rápido: instalar o agente em uma instância do EC2 do Linux na inicialização 22
Início rápido: use CloudWatch registros com instâncias do Windows Server 2016 26
Início rápido: use CloudWatch registros com instâncias do Windows Server 2012 e do Windows Server 2008 37
Início rápido: instale o agente usando AWS OpsWorks 48
Relate o status do agente do CloudWatch Logs 54
Inicie o agente CloudWatch Logs 54
Pare o agente CloudWatch de registros 55
Início rápido com AWS CloudFormation 55
Trabalhando com AWS SDKs 57
Análise de dados de registro com o CloudWatch Logs Insights 59
Comandos suportados em classes de log 61
Primeiros passos: tutoriais de consulta 61
Tutorial: executar e modificar um exemplo de consulta 61
Tutorial: Executar uma consulta com uma função de agregação 64
Tutorial: Executar uma consulta que produz uma visualização agrupada por campos de log 65

Tutorial: Executar uma consulta que produz uma visualização de séries temporais 66
Logs compatíveis e campos descobertos 67
Campos em logs JSON 69
Sintaxe de consulta 71
display 74
fields 75
filtrar 75
pattern 78
diferença 79
parse 80
sort 82
stats 83
limite 90
dedup 90
unmask 91
Funções booleanas, de comparação, numéricas e de data e hora, entre outras 91
Campos contendo caracteres especiais 101
Usar aliases e comentários em consultas 101
Análise de padrões 103
Introdução à análise de padrões 103
Detalhes sobre o comando pattern 106
Compare (diff) com intervalos de tempo anteriores 107
Consultas de exemplo 109
Consultas gerais 110
Consultas de logs do Lambda 110
Consultas para logs de fluxo da Amazon VPC 111
Consultas de logs do Route 53 112
Consultas para registros CloudTrail 112
Consultas para Amazon API Gateway 114
Consultas para gateway NAT 114
Consultas para logs do servidor Apache 116
Consultas para a Amazon EventBridge 116
Exemplos do comando de análise 116
Visualize dados de log em grafos 117
Salvar e executar consultas novamente 117
Adicionar consulta ao painel ou exportar os resultados da consulta 120

Exibir as consultas em execução ou o histórico de consultas 120
Criptografe os resultados da consulta com AWS Key Management Service 121
Limites 122
Etapa 1: criar um AWS KMS key 122
Etapa 2: definir permissões na chave do KMS 123
Etapa 3: associar uma chave do KMS aos resultados da consulta 124
Etapa 4: desassociar uma chave dos resultados da consulta na conta 125
Use linguagem natural para gerar e atualizar consultas do CloudWatch Logs Insights 125
Consultas de exemplo 126
Optar por não usar seus dados para melhorar o serviço 128
Detecção de anomalias de log 129
Gravidade e prioridade de anomalias e padrões 130
Tempo de visibilidade da anomalia 130
Suprimindo uma anomalia 130
Perguntas frequentes 131
Ativar a detecção de anomalias em um grupo de registros 132
Exibir anomalias que foram encontradas 133
Crie alarmes em detectores de anomalias de log 136
Métricas publicadas por detectores de anomalias de log 139
Criptografe um detector de anomalias e seus resultados com AWS KMS 139
Limites 140
Trabalhar com grupos de logs e fluxos de logs 144
Criar um grupo de logs 144
Enviar logs a um grupo de logs 145
Visualizar dados de log 145
Usar o Live Tail para visualizar logs quase em tempo real 146
Iniciar uma sessão Live Tail 146
Pesquisar dados de log usando padrões de filtro 149
Pesquisar entradas de log usando o console 149
Pesquisar entradas de registro usando o AWS CLI 150
Passar de métricas para logs 150
Solução de problemas 151
Alterar a retenção do log de dados 152
Etiquetar grupos de logs 153
Conceitos Básicos de Tags 153
Monitorar custos usando a marcação 154

Restrições de tags 154
Marcando grupos de registros usando o AWS CLI 155
Como marcar grupos de registros usando a API CloudWatch Logs 155
Criptografe dados de registro usando AWS KMS 156
Limites 157
Etapa 1: criar uma AWS KMS chave 122
Etapa 2: definir permissões na chave do KMS 123
Etapa 3: associar uma chave do KMS a um grupo de logs 143
Etapa 4: desassociar uma chave de um grupo de logs 143
Chaves do KMS e contexto de criptografia 161
Ajude a proteger dados de log confidenciais com mascaramento 164
Noções básicas sobre políticas de proteção de dados 168
Permissões de IAM necessárias para criar ou trabalhar com uma política de proteção de dados 170
Criar uma política de proteção de dados para toda a conta 176
Criar uma política de proteção de dados para um único grupo de logs 179
Exibir dados não mascarados 182
Relatórios de descobertas de auditoria 183
Tipos de dados que você pode proteger 185
Filtros de métrica 230
Conceitos 231
Sintaxe de padrões de filtros para filtros de métricas 232
Configurando valores métricos para um filtro de métrica 233
Como publicar dimensões com métricas de eventos de log 234
Usando valores em eventos de log para incrementar o valor de uma métrica 237
Criar filtros de métrica 238
Criar um filtro de métrica para um grupo de logs 239
Exemplo: contar eventos de log 240
Exemplo: contar as ocorrências de um termo 241
Exemplo: contar códigos HTTP 404 243
Exemplo: contar códigos HTTP 4xx 246
Exemplo: Extrair campos de um log Apache e atribuir dimensões 247
Listagem de filtros de métrica 249
Excluir um filtro de métrica 250
Filtros de assinatura 251
Conceitos 252

Registre filtros de assinatura em nível de grupo ... 253
Exemplo 1: filtros de assinatura com o Kinesis Data Streams ... 254
Exemplo 2: filtros de assinatura com AWS Lambda ... 260
Exemplo 3: filtros de assinatura com o Amazon Data Firehose ... 263
Filtros de assinatura em nível de conta ... 271
Exemplo 1: filtros de assinatura com o Kinesis Data Streams ... 271
Exemplo 2: filtros de assinatura com AWS Lambda ... 278
Exemplo 3: filtros de assinatura com o Amazon Data Firehose ... 282
Assinaturas entre contas e regiões ... 290
Compartilhamento de dados de log entre contas usando o Kinesis Data Streams ... 291
Compartilhamento de dados de registro entre contas usando Firehose ... 311
Assinaturas multiregionais em nível de conta usando o Kinesis Data Streams ... 325
Assinaturas multiregionais em nível de conta usando Firehose ... 343
Prevenção de ‘confused deputy’ ... 356
Prevenção de recursão de registros ... 357
Sintaxe de padrões de filtros ... 359
Expressões regulares compatíveis ... 360
Fazer a correspondência de termos usando expressões regulares ... 363
Fazer a correspondência de termos em eventos de log não estruturados ... 363
Fazer correspondência de termos em eventos de log JSON ... 367
Fazer a correspondência de termos em eventos de log delimitados por espaços ... 376
Habilitar o registro a partir de AWS serviços ... 381
Registro em log que requer permissões [v1] adicionais ... 387
Registros enviados para CloudWatch Logs ... 387
Logs enviados ao Amazon S3 ... 390
Registros enviados para o Firehose ... 394
Registro em log que requer permissões [v2] adicionais ... 395
Registros enviados para CloudWatch Logs ... 397
Logs enviados ao Amazon S3 ... 399
Registros enviados para o Firehose ... 404
Permissões específicas do serviço ... 406
Permissões específicas do console ... 407
Prevenção contra o ataque do “substituto confuso” em todos os serviços ... 408
Atualizações da política ... 409
Exportar dados de log para o Amazon S3 ... 411
Conceitos ... 412

Exportação de dados de log para o Amazon S3 usando o console. ... 413
Exportação para a mesma conta ... 414
Exportação entre contas ... 421
Exporte dados de log para o Amazon S3 usando o AWS CLI ... 429
Exportação para a mesma conta ... 430
Exportação entre contas ... 437
Descrever tarefas de exportação ... 446
Cancelar uma tarefa de exportação ... 447
Streaming de dados para o OpenSearch serviço ... 448
Pré-requisitos ... 448
Inscrever um grupo de registros no OpenSearch Serviço ... 449
Exemplos de código ... 451
Ações ... 452
AssociateKmsKey ... 453
CancelExportTask ... 454
CreateExportTask ... 456
CreateLogGroup ... 457
CreateLogStream ... 460
DeleteLogGroup ... 462
DeleteSubscriptionFilter ... 464
DescribeExportTasks ... 470
DescribeLogGroups ... 471
DescribeSubscriptionFilters ... 475
GetQueryResults ... 481
PutSubscriptionFilter ... 483
StartLiveTail ... 489
StartQuery ... 501
Cenários ... 504
Executar uma consulta grande ... 505
Exemplos entre serviços ... 520
Usar eventos programados para invocar uma função do Lambda ... 520
Segurança ... 522
Proteção de dados ... 523
Criptografia em repouso ... 524
Criptografia em trânsito ... 524
Gerenciamento de identidade e acesso ... 524

Autenticação 525
Controle de acesso 525
Visão geral do gerenciamento de acesso 525
Usar políticas baseadas em identidade (políticas do IAM) 531
CloudWatch Referência de permissões de registros 544
Usar funções vinculadas ao serviço 550
Validação de conformidade 552
Resiliência 553
Segurança da infraestrutura 554
Endpoints da VPC de interface 554
Disponibilidade 555
Criação de um VPC endpoint para registros CloudWatch 555
Testando a conexão entre sua VPC e Logs CloudWatch 555
Controle do acesso ao seu endpoint CloudWatch VPC do Logs 556
Compatibilidade com chaves de contexto da VPC 557
Registrando operações de API e console com AWS CloudTrail 558
CloudWatch Registra as informações em CloudTrail 558
Informações de geração de consultas em CloudTrail 560
Noções básicas sobre entradas de arquivos de log do 562
Referência do agente 564
Arquivo de configuração do agente 564
Usando o agente CloudWatch Logs com proxies HTTP 570
Compartimentalizando os arquivos de configuração do agente CloudWatch Logs 571
CloudWatch Perguntas frequentes sobre o agente de registros 572
Monitorando o uso com CloudWatch métricas 576
CloudWatch Métricas de registros 576
Dimensões para métricas CloudWatch de registros 580
CloudWatch Registra métricas de uso do serviço 581
Cotas de serviço 584
Gerenciando suas cotas do serviço CloudWatch Logs 590
Histórico do documento 592
AWS Glossário 600
..... dci

O que é o Amazon CloudWatch Logs?

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

CloudWatch O Logs permite que você centralize os registros de todos os seus sistemas, aplicativos e AWS serviços que você usa, em um único serviço altamente escalável. Em seguida, você pode facilmente visualizá-los, pesquisá-los em busca de códigos de erro ou padrões específicos, filtrá-los com base em campos específicos ou arquivá-los com segurança para futuras análises. CloudWatch Os registros permitem que você veja todos os seus registros, independentemente de sua origem, como um fluxo único e consistente de eventos ordenados por horário.

CloudWatch O Logs também permite consultar seus registros com uma linguagem de consulta poderosa, auditar e mascarar dados confidenciais em registros e gerar métricas a partir de registros usando filtros ou um formato de registro incorporado.

CloudWatch O Logs oferece suporte a duas classes de log. Os grupos de CloudWatch registros na classe de registros padrão são compatíveis com todos os recursos de CloudWatch registros. Os grupos de CloudWatch registros na classe de registros Logs Infrequent Access incorrem em taxas de ingestão mais baixas e oferecem suporte a um subconjunto dos recursos da classe Standard. Para ter mais informações, consulte Classes de log.

Atributos

- Duas classes de registro para flexibilidade — o CloudWatch Logs oferece duas classes de registro para que você possa ter uma opção econômica para registros que você acessa com pouca frequência. Você também tem uma opção completa para registros que exigem monitoramento em tempo real ou outros recursos. Para ter mais informações, consulte Classes de log.
- Consulte seus dados de registro — Você pode usar o CloudWatch Logs Insights para pesquisar e analisar interativamente seus dados de registro. Você pode realizar consultas para ajudá-lo a responder com mais eficiência e eficácia aos problemas operacionais. CloudWatch O Logs Insights inclui uma linguagem de consulta específica 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. Exemplos de consultas estão incluídos para vários tipos de registros de AWS serviço. Para começar, consulte o Análise de dados de registro com o CloudWatch Logs Insights.

- Detectar e depurar usando o Live Tail: use o Live Tail para solucionar incidentes rapidamente, visualizando uma lista de streaming de novos eventos de log à medida que eles são consumidos. É possível visualizar, filtrar e realçar logs consumidos quase em tempo real, o que ajuda a detectar e resolver problemas rapidamente. Os logs podem ser filtrados com base em termos especificados, e você também pode realçar logs que contêm termos específicos para ajudar a encontrar rapidamente o que está procurando. Para ter mais informações, consulte Usar o Live Tail para visualizar logs quase em tempo real.
- Monitore registros de instâncias do Amazon EC2 — Você pode usar CloudWatch registros para monitorar aplicativos e sistemas usando dados de log. Por exemplo, o CloudWatch Logs pode rastrear o número de erros que ocorrem nos registros do seu aplicativo e enviar uma notificação sempre que a taxa de erros exceder um limite especificado por você. CloudWatch O Logs usa seus dados de registro para monitoramento; portanto, nenhuma alteração no código é necessária. Por exemplo, você pode monitorar registros de aplicativos em busca de termos literais específicos (como "NullReferenceException“) ou contar o número de ocorrências de um termo literal em uma posição específica nos dados de registro (como códigos de status “404" em um registro de acesso do Apache). Quando o termo que você está procurando é encontrado, o CloudWatch Logs reporta os dados para uma CloudWatch métrica que você especifica. Os dados de log são criptografados em trânsito e em repouso. Para começar, consulte o Introdução ao CloudWatch Logs.
- Monitore eventos AWS CloudTrail registrados — Você pode criar alarmes CloudWatch e receber notificações de atividades específicas da API, conforme capturadas por, CloudTrail e usar a notificação para solucionar problemas. Para começar, consulte Enviar CloudTrail eventos para CloudWatch registros no Guia do AWS CloudTrail usuário.
- Auditar e mascarar dados confidenciais: se você tiver dados confidenciais em seus logs, poderá ajudar a protegê-los com políticas de proteção de dados. Essas políticas permitem auditar e mascarar dados confidenciais. Se você ativar a proteção de dados, por padrão, os dados confidenciais que corresponderem aos identificadores de dados selecionados serão mascarados. Para ter mais informações, consulte Ajude a proteger dados de log confidenciais com mascaramento.
- Retenção de logs - por padrão, os logs são mantidos por tempo indeterminado e nunca saem da validade. 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 registro — Você pode usar o CloudWatch Logs para armazenar seus dados de log em um armazenamento altamente durável. O agente de CloudWatch registros facilita o envio rápido de dados de registro rotativos e não rotacionados de um host para o serviço de registro. Em seguida, você poderá acessar os dados de log brutos quando forem necessários.

- Registrar consultas DNS do Route 53 — Você pode usar CloudWatch os registros para registrar informações sobre as consultas de DNS que o Route 53 recebe. Para mais informações, consulte Registrar consultas de DNS no Guia do desenvolvedor do Amazon Route 53.

AWS Serviços relacionados

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

- AWS CloudTrailé um serviço da web que permite monitorar as chamadas feitas para a API CloudWatch Logs da sua conta, incluindo chamadas feitas pelo AWS Management Console, AWS Command Line Interface (AWS CLI) e outros serviços. Quando o CloudTrail registro está ativado, CloudTrail captura chamadas de API em sua conta e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. 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 AWS CloudTrail, consulte O que é AWS CloudTrail? no Guia do AWS CloudTrail usuário. Para obter um exemplo do tipo de dados que são CloudWatch gravados em arquivos de CloudTrail log, consulteOperações da API CloudWatch Logging Logs e do console em AWS CloudTrail.
- AWS Identity and Access Management (IAM) é um serviço web que ajuda você a controlar com segurança o acesso aos AWS recursos para seus usuários. 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 o IAM? no Manual do usuário do IAM.
- O 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 aplicação, 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.
- O AWS Lambda é um serviço da Web que você pode usar para criar aplicações que respondam rapidamente a novas informações. Carregue código da aplicação como funções Lambda, e o Lambda executará seu código em uma infraestrutura de computação de alta disponibilidade e efetuará 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 AWS Lambda? no Guia do AWS Lambda desenvolvedor.

Definição de preço

Ao se inscrever AWS, você pode começar a usar o CloudWatch Logs gratuitamente usando o nível AWS gratuito.

As tarifas padrão se aplicam a registros armazenados por outros serviços usando CloudWatch registros (por exemplo, registros de fluxo do Amazon VPC e registros do Lambda).

Para obter mais informações sobre preços, consulte Amazon CloudWatch Pricing.

Para obter mais informações sobre como analisar seus custos e uso do CloudWatch Logs e CloudWatch sobre as melhores práticas sobre como reduzir seus custos, consulte CloudWatch faturamento e custo.

Conceitos do Amazon CloudWatch Logs

A terminologia e os conceitos fundamentais para sua compreensão e uso dos CloudWatch registros estão descritos abaixo.

Classe de registro

CloudWatch O Logs oferece duas classes de grupos de registros. A classe de registro Standard é uma opção completa para registros que exigem monitoramento em tempo real ou registros que você acessa com frequência. A classe de registro de acesso infrequente é uma opção de baixo custo para registros que você acessa com menos frequência. Ele oferece suporte a um subconjunto dos recursos da classe de log padrão.

Eventos de log

Evento de log é um registro de alguma atividade registrada pela aplicação ou recurso que está sendo monitorado. O registro de eventos de log que o CloudWatch Logs compreende contém duas propriedades: a data e hora de quando o evento ocorreu e a mensagem bruta do evento. As mensagens de eventos devem estar codificadas por UTF-8.

Fluxos de log

Uma transmissão de log é uma sequência de eventos de log que compartilham a mesma fonte. Mais especificamente, um stream de log geralmente representa a sequência de eventos que vem

da instância da aplicação 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 delete-log-stream comando aws logs.

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 métricos para extrair observações métricas de eventos ingeridos e transformá-las em pontos de dados em uma CloudWatch métrica. 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 registro são mantidos nos CloudWatch registros. 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.

Faturamento e custos do Amazon CloudWatch Logs

Para obter informações detalhadas sobre como analisar seus custos e uso para o CloudWatch Logs e o CloudWatch e para conhecer as práticas recomendadas sobre como reduzir seus custos, consulte Faturamento e custos do CloudWatch.

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

Ao se cadastrar na AWS, você poderá começar a usar o CloudWatch Logs gratuitamente utilizando 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 da Amazon VPC e os logs do Lambda).

Classes de log

CloudWatch O Logs oferece duas classes de grupos de registros:

- A classe de registro CloudWatch Logs Standard é uma opção completa para registros que exigem monitoramento em tempo real ou registros que você acessa com frequência.
- A classe de registro CloudWatch Logs Infrequent Access é uma nova classe de registro que você pode usar para consolidar seus registros de forma econômica. Essa classe de registro oferece um subconjunto de recursos de CloudWatch registros, incluindo ingestão gerenciada, armazenamento, análise de registros entre contas e criptografia, com um preço de ingestão menor por GB. A classe de registro de acesso infrequente é ideal para consultas ad-hoc e análise after-the-fact forense em registros acessados com pouca frequência.

> **Note**
>
> No que diz respeito às cobranças, as classes de registro de acesso padrão e infrequente diferem somente nos custos de ingestão. As cobranças de armazenamento e CloudWatch as cobranças do Logs Insights são as mesmas em cada classe de registro.

Para obter mais informações sobre CloudWatch os preços do Logs, consulte Amazon CloudWatch Pricing.

> **Important**
>
> Depois que um grupo de registros é criado, sua classe de log não pode ser alterada.

Atributos compatíveis

A tabela a seguir lista os recursos de cada classe de log.

	Padrão	Acesso infrequente	
Ingestão e armazenamento de registros totalmente gerenciados	✓	✓	
Recursos de várias contas	✓	✓	
Criptografia com AWS KMS	✓	✓	
CloudWatch Comandos de consulta do Logs Insights	✓	✓ (A maioria dos comandos—veja Comandos suportados em classes de log.)	
CloudWatch Campos descobertos do Logs Insights	✓		
Assistência de consulta em linguagem natural	✓		
CloudWatch Detecção de anomalias de registros	✓		
Compare com o intervalo de tempo anterior	✓		
Filtros de assinatura	✓		
Exportar para o Amazon S3.	✓		
GetLogEvents e operações FilterLogEvents de API	✓	Sem suporte. Use o CloudWatch Logs Insights para visualizar eventos de registro	

	Padrão	Acesso infrequente	
		armazenados em grupos de registros na classe de registros de acesso infrequente.	
Filtros métricos	✓		
Ingestão de registros do Container Insights	✓		
Ingestão de registros do Lambda Insights	✓		
Proteção de dados confidenciais com mascaramento	✓		
Formato de métricas incorporadas	✓		

Introdução ao CloudWatch Logs

Para coletar registros de suas instâncias do Amazon EC2 e servidores locais em CloudWatch Logs, use o agente unificado. 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 performance.

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

Para obter mais informações, consulte Instalando o CloudWatch agente no Guia CloudWatch do usuário da Amazon.

O agente CloudWatch Logs mais antigo, que oferece suporte somente à coleta de registros de servidores que executam Linux, está obsoleto e não é mais suportado. Para obter informações sobre a migração do agente CloudWatch Logs antigo para o agente unificado, consulte Criar o arquivo de configuração do CloudWatch agente com o assistente.

Conteúdo

- Pré-requisitos
- Use o CloudWatch agente unificado para começar a usar o CloudWatch Logs
- Use o CloudWatch agente anterior para começar a usar o CloudWatch Logs
- Início rápido: use AWS CloudFormation para começar a usar o CloudWatch Logs

Pré-requisitos

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

Inscreva-se para um Conta da AWS

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

Para se inscrever em um Conta da AWS

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

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

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

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando https://aws.amazon.com/ e selecionando Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

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

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

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

Criar um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.

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

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

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

Iniciar sessão como o usuário com acesso administrativo

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

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

Atribuir acesso a usuários adicionais

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

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

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

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

Configurar a interface de linha de comando

Você pode usar o AWS CLI para realizar operações CloudWatch de registros.

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

Use o CloudWatch agente unificado para começar a usar o CloudWatch Logs

Para obter mais informações sobre o uso do CloudWatch agente unificado para começar a usar o CloudWatch Logs, consulte Coletar métricas e registros de instâncias do Amazon EC2 e servidores locais com o CloudWatch agente no Guia do usuário da Amazon CloudWatch . Você conclui as etapas listadas nesta seção para instalar, configurar e iniciar o agente. Se você não estiver usando o agente para também coletar CloudWatch métricas, você pode ignorar qualquer seção que se refira a métricas.

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

Use o CloudWatch agente anterior para começar a usar o CloudWatch Logs

Important

CloudWatch inclui um CloudWatch agente unificado que pode coletar registros e métricas de instâncias do EC2 e servidores locais. O agente somente para logs mais antigo está obsoleto e não é mais compatível.

Para obter informações sobre a migração do antigo agente somente de registros para o agente unificado, consulte Criar o arquivo de configuração do CloudWatch agente com o assistente.

O restante desta seção explica o uso do agente CloudWatch Logs antigo para clientes que ainda o usam.

Usando o agente CloudWatch Logs, você pode publicar dados de log de instâncias do Amazon EC2 executando Linux ou Windows Server e eventos registrados de. AWS CloudTrail Em vez disso, recomendamos usar o agente CloudWatch unificado para publicar seus dados de registro. Para obter mais informações sobre o novo agente, consulte Coletar métricas e registros de instâncias

do Amazon EC2 e servidores locais com o CloudWatch agente no Guia do usuário da Amazon CloudWatch .

Conteúdo

- CloudWatch Pré-requisitos do agente de registros
- Início rápido: instale e configure o agente CloudWatch Logs em uma instância Linux do EC2 em execução
- Início rápido: instale e configure o agente CloudWatch Logs em uma instância do EC2 Linux na inicialização
- Início rápido: habilite suas instâncias do Amazon EC2 executando o Windows Server 2016 para enviar registros para Logs usando o CloudWatch agente CloudWatch Logs
- Início rápido: habilite suas instâncias do Amazon EC2 executando o Windows Server 2012 e o Windows Server 2008 para enviar registros para o Logs CloudWatch
- Início rápido: instale o agente CloudWatch Logs usando o AWS OpsWorks Chef
- Relate o status do agente do CloudWatch Logs
- Inicie o agente CloudWatch Logs
- Pare o agente CloudWatch de registros

CloudWatch Pré-requisitos do agente de registros

O agente CloudWatch Logs exige 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: instale e configure o agente CloudWatch Logs em uma instância Linux do EC2 em execução

Important

O agente de registros mais antigo está obsoleto. CloudWatch inclui um agente unificado que pode coletar registros e métricas de instâncias do EC2 e servidores locais. Para ter mais informações, consulte Introdução ao CloudWatch Logs.
Para obter informações sobre a migração do agente CloudWatch Logs antigo para o agente unificado, consulte Criar o arquivo de configuração do CloudWatch agente com o assistente.
O agente de logs mais antigo é compatível somente com as versões 2.6 a 3.5 do Python. Além disso, o agente CloudWatch Logs mais antigo não é compatível com o Instance Metadata Service Version 2 (IMDSv2). Se seu servidor usa o IMDSv2, você deve usar o agente unificado mais novo em vez do agente de registros mais antigo. CloudWatch
O restante desta seção explica o uso do agente CloudWatch Logs antigo para clientes que ainda o usam.

Tip

CloudWatch inclui um novo agente unificado que pode coletar registros e métricas de instâncias do EC2 e servidores locais. Se você ainda não estiver usando o agente CloudWatch Logs antigo, recomendamos usar o CloudWatch agente unificado mais novo. Para ter mais informações, consulte Introdução ao CloudWatch Logs.
Além disso, o agente mais antigo não é compatível com o Instance Metadata Service versão 2 (IMDSv2). Se seu servidor usa o IMDSv2, você deve usar o agente unificado mais novo em vez do agente de registros mais antigo. CloudWatch
O restante desta seção explica o uso do antigo agente CloudWatch Logs.

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

Você pode usar o instalador do agente CloudWatch Logs em uma instância EC2 existente para instalar e configurar o agente 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 registro usando o AWS CLI SDK do CloudWatch Logs ou a API CloudWatch Logs. O AWS CLI é mais adequado para publicar dados na linha de comando ou por meio de scripts. O SDK de CloudWatch registros é mais adequado para publicar dados de registro diretamente de aplicativos ou criar seu próprio aplicativo de publicação de registros.

Etapa 1: configurar sua função ou usuário do IAM para CloudWatch Logs

O agente CloudWatch Logs é compatível com funções e usuários do IAM. Se 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 CloudWatch Logs

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Perfis.
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 registros em uma instância existente do Amazon EC2

O processo de instalação do agente CloudWatch Logs difere dependendo se sua instância do Amazon EC2 está executando Amazon Linux, Ubuntu, CentOS ou Red Hat. Use as etapas apropriadas para a versão do Linux na sua instância.

Para instalar e configurar CloudWatch registros em uma instância existente do Amazon Linux

A partir do Amazon Linux AMI 2014.09, o agente 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 recebe atualizações e patches regulares do pacote AWS sem precisar reinstalar manualmente o CloudWatch agente do Logs.

> **Warning**
>
> Não atualize o agente 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 impedem que o agente do CloudWatch Logs envie seus registros para CloudWatch o.

1. Conecte-se à sua instância do Amazon Linux. Para obter mais informações, consulte Connect to Your Instance no Guia do usuário do Amazon EC2.

Para obter mais informações sobre problemas de conexão, consulte Solução de problemas de conexão com sua instância no Guia do usuário do Amazon EC2.

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 CloudWatch Referência do agente de registros.
5. Por padrão, o `/etc/awslogs/awscli.conf` aponta para a região us-east-1. Para enviar seus logs a 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 registros e o stream de registros recém-criados no CloudWatch console depois que o agente estiver em execução por alguns instantes.

   Para ter mais informações, consulte Exibir dados de registro enviados para o CloudWatch Logs.

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

Se você estiver usando uma AMI executando o Ubuntu Server, CentOS ou Red Hat, use o procedimento a seguir para instalar manualmente o agente CloudWatch Logs na sua instância.

1. Conecte-se à sua instância do EC2. Para obter mais informações, consulte Connect to Your Instance no Guia do usuário do Amazon EC2.

   Para obter mais informações sobre problemas de conexão, consulte Solução de problemas de conexão com sua instância no Guia do usuário do Amazon EC2.

2. Execute o instalador do agente 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 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 CloudWatch Logs especificando us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-northeast-2, ap-southeast-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, eutheast-2 regiões eu-central-1, eu-west-1 ou 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 CloudWatch Logs exige certas 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	Descrição
AWS ID da chave de acesso	Aperte Enter se estiver usando uma função do IAM. Caso contrário, insira o ID da chave de AWS acesso.
AWS chave de acesso secreta	Aperte Enter se estiver usando uma função do IAM. Caso contrário, insira sua chave de acesso AWS secreta.
Nome da região padrão	Pressione Enter. A região padrão é us-east-2. É possível definir para 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 ou 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 CloudWatch Referência do agente de registros.

> **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 registros e o stream de registros recém-criados no CloudWatch console depois que o agente estiver em execução por alguns instantes.

   Para ter mais informações, consulte Exibir dados de registro enviados para o CloudWatch Logs.

Início rápido: instale e configure o agente CloudWatch Logs em uma instância do EC2 Linux na inicialização

> **Tip**
>
> O antigo agente do CloudWatch Logs discutido nesta seção está prestes a ser descontinuado. É altamente recomendável que você use o novo CloudWatch agente unificado que pode coletar registros e métricas. Além disso, o agente CloudWatch Logs mais antigo exige 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 CloudWatch agente unificado, consulte Instalando o CloudWatch agente.
> O restante desta seção explica o uso do antigo agente CloudWatch Logs.

Instalando o agente CloudWatch Logs antigo em uma instância Linux do EC2 na inicialização

Você pode usar os dados do usuário do Amazon EC2, um recurso do Amazon EC2 que permite que informações paramétricas sejam passadas para a instância na inicialização, para instalar e configurar CloudWatch o agente Logs nessa instância. Para passar as informações de instalação e configuração do agente 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.

Pré-requisito

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 CloudWatch Referência do agente de registros.

Veja a seguir o exemplo de um arquivo de configuração do agente para o Amazon Linux 2

```
[general]
state_file = /var/lib/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.
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. Escolha Create Policy.
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 escolha Next Step (Próxima etapa).
12. Selecione Criar função.

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

Para iniciar uma nova instância e ativar CloudWatch os registros

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

   Para obter mais informações, consulte Lançamento de uma instância no Guia do usuário do Amazon EC2.
3. Na página Etapa 1: escolher uma Imagem de Máquina da Amazon (AMI), selecione o tipo de instância Linux para iniciar e, em seguida, na página Etapa 2: escolher um tipo de instância, selecione Próximo: configurar detalhes da instância.

   Certifique-se de que cloud-init seja incluído na sua Imagem de máquina da Amazon (AMI). As AMIs do Amazon Linux e as AMIs para Ubuntu e RHEL já incluem cloud-init, mas talvez o CentOS e outras AMIs não. AWS Marketplace
4. Na página Etapa 3: Configurar detalhes da instância, em IAM role (Função do IAM), selecione a função do IAM que você criou.
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 registros e o stream de registros recém-criados no CloudWatch console depois que o agente estiver em execução por alguns instantes.

   Para ter mais informações, consulte Exibir dados de registro enviados para o CloudWatch Logs.

Início rápido: habilite suas instâncias do Amazon EC2 executando o Windows Server 2016 para enviar registros para Logs usando o CloudWatch agente CloudWatch Logs

> **Tip**
>
> CloudWatch inclui um novo agente unificado que pode coletar registros e métricas de instâncias do EC2 e servidores locais. Recomendamos que você use o CloudWatch agente unificado mais novo. Para ter mais informações, consulte Introdução ao CloudWatch Logs. O restante desta seção explica o uso do antigo agente CloudWatch Logs.

Habilite suas instâncias do Amazon EC2 executando o Windows Server 2016 para enviar registros para Logs usando o agente de CloudWatch Logs mais antigo CloudWatch

Há vários métodos que você pode usar para permitir que instâncias que executam o Windows Server 2016 enviem registros para o CloudWatch Logs. As etapas desta seção usam o Run Command do Systems Manager. Para obter informações sobre os outros métodos possíveis, consulte Envio de registros, eventos e contadores de desempenho para a Amazon CloudWatch.

Etapas

- Baixe o arquivo de configuração de exemplo
- Configure o arquivo JSON para CloudWatch
- Criar um perfil do IAM para o Systems Manager
- Verifique os pré-requisitos do Systems Manager
- Verificar o acesso à Internet
- Ativar CloudWatch registros usando o comando de execução do Systems Manager

Baixe o arquivo de configuração de exemplo

Baixe o seguinte arquivo de amostra em seu computador: `AWS.EC2.Windows.CloudWatch.json`.

Configure o arquivo JSON para CloudWatch

Você determina para quais registros enviar 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: ativar CloudWatch registros
- Etapa 2: definir as configurações para CloudWatch
- Etapa 3: Configurar os dados a serem enviados
- Etapa 4: Configurar o controle de fluxo
- Etapa 5: Salvar conteúdo JSON

Etapa 1: ativar CloudWatch registros

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

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

Etapa 2: definir as configurações para CloudWatch

Especifique as credenciais, a região, um nome de grupo de logs e um namespace de fluxo de logs. Isso permite que a instância envie dados de registro para o CloudWatch Logs. Para enviar os mesmos dados de registro para locais diferentes, você pode adicionar seções adicionais com IDs exclusivos (por exemplo, "CloudWatchLogs2" e CloudWatchLogs 3") e uma região diferente para cada ID.

Para definir as configurações para enviar dados de registro para o 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. Você configura 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 Grupos de registros no CloudWatch console.
5. Em `LogStream`, digite o fluxo de log de destino. Esse nome aparece na tela Grupos de registros **>** Streams no CloudWatch console.

   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 registros que ainda não existe, o CloudWatch Logs o cria automaticamente para você. 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 a serem enviados

Você pode enviar dados de registro de eventos, dados de rastreamento de eventos para Windows (ETW) e outros dados de registro para CloudWatch o Logs.

Para enviar dados do registro de eventos do aplicativo 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. É possível 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 do registro 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 do registro 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. É possível 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 registro 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. É possível 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 do Event Tracing for Windows para o Logs CloudWatch

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. É possível 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 registros personalizados (qualquer arquivo de registro baseado em texto) para o Logs CloudWatch

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 suportados, consulte o tópico FileSystemWatcherFilter Propriedade 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**
>
> Os valores `div`, `div-MV`, `hu` e `hu-HU` não têm suporte.

7. (Opcional) Em `TimeZoneKind`, digite `Local` ou `UTC`. Você pode definir isso para fornecer informações de fuso horário quando essas informações não estiverem incluídas no time stamp do log. Se esse parâmetro for deixado em branco e se seu carimbo de data/hora não incluir informações de fuso horário, o CloudWatch Logs usará como padrão o fuso horário local. 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 para o 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 suportados, consulte o tópico FileSystemWatcherFilter Propriedade 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**
>
> Os valores `div`, `div-MV`, `hu` e `hu-HU` não têm suporte.

7. (Opcional) Em `TimeZoneKind`, digite `Local` ou `UTC`. Você pode definir isso para fornecer informações de fuso horário quando essas informações não estiverem incluídas no time stamp do log. Se esse parâmetro for deixado em branco e se seu carimbo de data/hora não incluir informações de fuso horário, o CloudWatch Logs usará como padrão o fuso horário local. 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 o controle de fluxo

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

> **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: Salvar 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 perfil do IAM para o Systems Manager

É necessário ter uma função do IAM para credenciais de instância ao usar o Run Command do Systems Manager. Essa função permite que o Systems Manager execute ações na instância. Para obter mais informações, consulte Configurar funções de segurança para o Systems Manager no Manual do usuário do AWS Systems Manager . Para obter informações sobre como anexar uma função do IAM a uma instância existente, consulte Anexar uma função do IAM a uma instância no Guia do usuário do Amazon EC2.

Verifique os pré-requisitos do Systems Manager

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

Verificar o acesso à Internet

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

Ativar CloudWatch registros usando o comando de execução 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 SSM Agent na instância processa o comando e configura a instância conforme especificado.

Para configurar a integração com o CloudWatch Logs usando o comando Executar

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, selecione Executar comando.
4. Escolha Executar um comando.
5. Para o documento de comando, escolha AWS- ConfigureCloudWatch.
6. Para instâncias do Target, escolha as instâncias a serem integradas com o CloudWatch Logs. Se você não vir uma instância nessa lista, ela poderá não estar configurada para Run Command. Para obter mais informações, consulte os pré-requisitos do Systems Manager no Guia do usuário do Amazon EC2.
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: habilite suas instâncias do Amazon EC2 executando o Windows Server 2012 e o Windows Server 2008 para enviar registros para o Logs CloudWatch

> **Tip**
>
> CloudWatch inclui um novo agente unificado que pode coletar registros e métricas de instâncias do EC2 e servidores locais. Recomendamos que você use o CloudWatch agente unificado mais novo. Para ter mais informações, consulte Introdução ao CloudWatch Logs. O restante desta seção explica o uso do antigo agente CloudWatch Logs.

Habilite suas instâncias do Amazon EC2 executando o Windows Server 2012 e o Windows Server 2008 para enviar registros para Logs CloudWatch

Use as etapas a seguir para permitir que suas instâncias que executam o Windows Server 2012 e o Windows Server 2008 enviem CloudWatch registros para o Logs.

Baixe o arquivo de configuração de exemplo

Baixe os seguintes arquivos JSON em seu computador: `AWS.EC2.Windows.CloudWatch.json`. Você o edita nas etapas a seguir.

Configure o arquivo JSON para CloudWatch

Você determina para quais registros enviar 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: ativar CloudWatch registros
- Etapa 2: definir as configurações para CloudWatch
- Etapa 3: Configurar os dados a serem enviados
- Etapa 4: Configurar o controle de fluxo

Etapa 1: ativar CloudWatch registros

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

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

Etapa 2: definir as configurações para CloudWatch

Especifique as credenciais, a região, um nome de grupo de logs e um namespace de fluxo de logs. Isso permite que a instância envie dados de registro para o CloudWatch Logs. Para enviar os mesmos dados de registro para locais diferentes, você pode adicionar seções adicionais com IDs exclusivos (por exemplo, "CloudWatchLogs2" e CloudWatchLogs 3") e uma região diferente para cada ID.

Para definir as configurações para enviar dados de registro para o 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. Você configura 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 Grupos de registros no CloudWatch console.
5. Em `LogStream`, digite o fluxo de log de destino. Esse nome aparece na tela Grupos de registros > Streams no CloudWatch console.

   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 registros que ainda não existe, o CloudWatch Logs o cria automaticamente para você. 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 a serem enviados

Você pode enviar dados de registro de eventos, dados de rastreamento de eventos para Windows (ETW) e outros dados de registro para CloudWatch o Logs.

Para enviar dados do registro de eventos do aplicativo 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. É possível 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 do registro 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 do registro 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. É possível 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 registro 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. É possível 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 do Event Tracing for Windows para o Logs CloudWatch

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. É possível 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 registros personalizados (qualquer arquivo de registro baseado em texto) para o Logs CloudWatch

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 suportados, consulte o tópico FileSystemWatcherFilter Propriedade 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**
>
> Os valores `div`, `div-MV`, `hu` e `hu-HU` não têm suporte.

7. (Opcional) Em `TimeZoneKind`, digite `Local` ou `UTC`. Você pode definir isso para fornecer informações de fuso horário quando essas informações não estiverem incluídas no time stamp do log. Se esse parâmetro for deixado em branco e se seu carimbo de data/hora não incluir informações de fuso horário, o CloudWatch Logs usará como padrão o fuso horário local. 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 para o 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 suportados, consulte o tópico FileSystemWatcherFilter Propriedade 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**
   >
   > Os valores `div`, `div-MV`, `hu` e `hu-HU` não têm suporte.

7. (Opcional) Em `TimeZoneKind`, digite `Local` ou `UTC`. Você pode definir isso para fornecer informações de fuso horário quando essas informações não estiverem incluídas no time stamp do log. Se esse parâmetro for deixado em branco e se seu carimbo de data/hora não incluir informações de fuso horário, o CloudWatch Logs usará como padrão o fuso horário local. 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 o controle de fluxo

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

> **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.

Inicie o agente

Para permitir que uma instância do Amazon EC2 executando o Windows Server 2012 ou o Windows Server 2008 envie CloudWatch registros para o Logs, use o serviço EC2Config (. `EC2Config.exe`) A instância deve ter o EC2Config 4.0 ou posterior, e você pode usar esse procedimento. Para obter mais informações sobre o uso de uma versão anterior do EC2Config, consulte Usar EC2Config 3.x ou anterior para configurar no Guia do usuário do Amazon EC2 CloudWatch

Para configurar 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 SSM (`AmazonSSMAgent.exe`) usando o painel de controle dos Serviços do Windows ou usando o seguinte PowerShell comando:

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

Depois que o agente SSM é reiniciado, ele detecta o arquivo de configuração e configura a instância para integração. CloudWatch Se você alterar os parâmetros e as configurações no arquivo de configuração local, será necessário reiniciar o SSM Agent para que as alterações sejam efetivadas. Para desativar a CloudWatch integração na instância, `IsEnabled false` altere e salve suas alterações no arquivo de configuração.

Início rápido: instale o agente CloudWatch Logs usando o AWS OpsWorks Chef

Você pode instalar o agente CloudWatch Logs e criar fluxos de log usando o AWS OpsWorks Chef, que é uma ferramenta terceirizada de automação de sistemas e infraestrutura em nuvem. 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. AWS OpsWorks suporta Git e 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 Manual do usuário do AWS OpsWorks . Os exemplos a seguir presumem que o livro de receitas seja denominado `logs`. A receita install.rb instala o agente Logs. CloudWatch Você também pode baixar o exemplo do 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 CloudWatch de configuração de registros:

```
#config.rb

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

Baixe e instale o agente 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, substitua *região* por uma das 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 ou 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 CloudWatch Referência do agente de registros.

```
[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 uma AWS OpsWorks pilha

1. Abra o AWS OpsWorks console em https://console.aws.amazon.com/opsworks/.
2. No OpsWorks Painel, escolha Adicionar pilha para criar uma AWS OpsWorks pilha.
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: Estender sua função do IAM

Para usar CloudWatch Logs com suas AWS OpsWorks instâncias, 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.
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. Escolha Create Policy.
7. No painel de navegação, escolha Funções e, em seguida, no painel de conteúdo, em Nome da função, selecione o nome da função da instância usada pela sua AWS OpsWorks pilha. 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 usuários e políticas, consulte Usuários e grupos do IAM e Gerenciar políticas do IAM no Guia do usuário do IAM.

Etapa 4: Adicionar uma camada

1. Abra o AWS OpsWorks console em https://console.aws.amazon.com/opsworks/.
2. No painel de navegação, escolha Camadas.

3. No painel de conteúdo, selecione uma camada e escolha Adicionar camada.
4. Na OpsWorksguia, em Tipo de camada, escolha Personalizado.
5. Nos campos Nome e Nome curto, digite os nomes longo e curto da camada. Em seguida, escolha Adicionar camada.
6. Na guia Receitas, em Receitas personalizadas do Chef, há vários títulos — Configuração, Configuração, Implantação, Desimplantação e Desativação — que correspondem aos eventos do ciclo de vida. AWS OpsWorks AWS OpsWorks aciona esses eventos nesses pontos-chave do 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.

   AWS OpsWorks executa essa receita em cada uma das novas instâncias dessa camada, logo após a inicialização da instância.

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 AWS OpsWorks console 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.

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

Etapa 6: Visualizar seus logs

Você deve ver o grupo de registros e o stream de registros recém-criados no CloudWatch console depois que o agente estiver em execução por alguns instantes.

Para ter mais informações, consulte Exibir dados de registro enviados para o CloudWatch Logs.

Relate o status do agente do CloudWatch Logs

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

Para relatar o status do agente

1. Conecte-se à sua instância do EC2. Para obter mais informações, consulte Connect to Your Instance no Guia do usuário do Amazon EC2.

   Para obter mais informações sobre problemas de conexão, consulte Solução de problemas de conexão com sua instância no Guia do usuário do Amazon EC2
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. Verifique se há erros, avisos ou problemas com o agente CloudWatch Logs no arquivo /var/log/awslogs.log.

Inicie o agente CloudWatch Logs

Se o agente CloudWatch Logs na sua instância do EC2 não foi iniciado automaticamente após a instalação, ou se você interrompeu o agente, você pode 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 Connect to Your Instance no Guia do usuário do Amazon EC2.

Para obter mais informações sobre problemas de conexão, consulte Solução de problemas de conexão com sua instância no Guia do usuário do Amazon EC2.

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
```

Pare o agente CloudWatch de registros

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

Para interromper o agente da

1. Conecte-se à sua instância do EC2. Para obter mais informações, consulte Connect to Your Instance no Guia do usuário do Amazon EC2.

   Para obter mais informações sobre problemas de conexão, consulte Solução de problemas de conexão com sua instância no Guia do usuário do Amazon EC2.
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: use AWS CloudFormation para começar a usar o CloudWatch Logs

AWS CloudFormation permite que você descreva e provisione seus AWS recursos no formato JSON. As vantagens desse método incluem ser capaz de gerenciar uma coleção de AWS recursos como uma única unidade e replicar facilmente seus AWS recursos em todas as regiões.

Ao provisionar AWS usando AWS CloudFormation, você cria modelos que descrevem os AWS recursos 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 muito mais avançadas do CloudWatch Logs usando. AWS CloudFormation Para obter mais informações sobre exemplos de modelos, consulte Amazon CloudWatch Logs Template Snippets no Guia do AWS CloudFormation usuário. Para obter mais informações sobre como começar, consulte Conceitos básicos do AWS CloudFormation no Manual do usuário do AWS CloudFormation .

Usando CloudWatch registros com um AWS SDK

AWS kits de desenvolvimento de software (SDKs) estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que facilitam a criação de aplicações em seu idioma preferido pelos desenvolvedores.

Documentação do SDK	Exemplos de código
AWS SDK for C++	AWS SDK for C++ exemplos de código
AWS CLI	AWS CLI exemplos de código
AWS SDK for Go	AWS SDK for Go exemplos de código
AWS SDK for Java	AWS SDK for Java exemplos de código
AWS SDK for JavaScript	AWS SDK for JavaScript exemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin exemplos de código
AWS SDK for .NET	AWS SDK for .NET exemplos de código
AWS SDK for PHP	AWS SDK for PHP exemplos de código
AWS Tools for PowerShell	Ferramentas para exemplos PowerShell de código
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) exemplos de código
AWS SDK for Ruby	AWS SDK for Ruby exemplos de código
AWS SDK para Rust	AWS SDK para Rust exemplos de código
SDK da AWS para SAP ABAP	SDK da AWS para SAP ABAP exemplos de código
AWS SDK for Swift	AWS SDK for Swift exemplos de código

Para exemplos específicos de CloudWatch registros, consulteExemplos de código para CloudWatch registros usando AWS SDKs.

> **Exemplo de disponibilidade**
>
> Você não consegue encontrar o que precisa? Solicite um código de exemplo no link Fornecer feedback na parte inferior desta página.

Análise de dados de registro com o CloudWatch Logs Insights

Com o CloudWatch Logs Insights, você pode pesquisar e analisar interativamente seus dados de log no Amazon CloudWatch Logs. Realize consultas para ajudar a responder de maneira mais rápida e eficiente a problemas operacionais. Se ocorrer um problema, você pode usar o CloudWatch Logs Insights para identificar possíveis causas e validar as correções implantadas.

CloudWatch O Logs Insights inclui uma linguagem de consulta específica com alguns comandos simples, mas poderosos. CloudWatch O Logs Insights fornece exemplos de consultas, descrições de comandos, preenchimento automático de consultas e descoberta de campos de registro para ajudar você a começar. Exemplos de consultas estão incluídos para diversos tipos de logs de serviço da AWS .

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

Você pode usar o CloudWatch Logs Insights para pesquisar dados de registro enviados ao CloudWatch Logs em 5 de novembro de 2018 ou mais tarde.

> **Important**
>
> CloudWatch O Logs Insights não pode acessar eventos de registro com carimbos de data e hora anteriores à hora de criação do grupo de registros.

Você também pode usar linguagem natural para criar consultas do CloudWatch Logs Insights. Para fazer isso, faça perguntas ou descreva os dados que você está procurando. Esse recurso assistido por IA gera uma consulta com base em sua solicitação e fornece uma line-by-line explicação de como a consulta funciona. Para obter mais informações, consulte Usar linguagem natural para gerar e atualizar consultas do CloudWatch Logs Insights.

Se você estiver conectado a uma conta configurada como uma conta de monitoramento na observabilidade CloudWatch entre contas, poderá executar consultas do CloudWatch Logs Insights em grupos de registros nas contas de origem vinculadas a essa conta de monitoramento. Você pode executar uma consulta que pesquisa vários grupos de logs localizados em contas diferentes. Para obter mais informações, consulte CloudWatch observabilidade entre contas.

Uma única solicitação pode consultar até 50 grupos de logs. O tempo limite das consultas, caso não sejam concluídas, é de 60 minutos. 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.

CloudWatch As consultas do Logs Insights incorrem em cobranças com base na quantidade de dados consultados. Para obter mais informações, consulte Amazon CloudWatch Pricing.

> **Important**
>
> Se sua equipe de segurança de rede não permite o uso de soquetes da web, atualmente você não pode acessar a parte do CloudWatch Logs Insights do CloudWatch console. Você pode usar os recursos de consulta do CloudWatch Logs Insights usando APIs. Para obter mais informações, consulte StartQuerya Referência da API Amazon CloudWatch Logs.

Conteúdo

- Comandos suportados em classes de log
- Primeiros passos: tutoriais de consulta
- Logs compatíveis e campos descobertos
- CloudWatch Sintaxe de consulta do Logs Insights
- Análise de padrões
- Compare (diff) com intervalos de tempo anteriores
- Consultas de exemplo
- Visualize dados de log em grafos
- Salve e execute novamente as consultas do CloudWatch Logs Insights
- Adicionar consulta ao painel ou exportar os resultados da consulta
- Exibir as consultas em execução ou o histórico de consultas
- Criptografe os resultados da consulta com AWS Key Management Service
- Use linguagem natural para gerar e atualizar consultas do CloudWatch Logs Insights

Comandos suportados em classes de log

Todos os comandos de consulta do CloudWatch Logs Insights são compatíveis com grupos de registros na classe de registro Standard. Os grupos de registros na classe de registro de Acesso Infrequente oferecem suporte a todos os comandos de consulta`pattern`, exceto`diff`, e. `unmask`

Primeiros passos: tutoriais de consulta

As seções a seguir incluem exemplos de tutoriais de consulta para ajudar você a começar a usar o CloudWatch Logs Insights.

Tópicos

- Tutorial: executar e modificar um exemplo de consulta
- Tutorial: Executar uma consulta com uma função de agregação
- Tutorial: Executar uma consulta que produz uma visualização agrupada por campos de log
- Tutorial: Executar uma consulta que produz uma visualização de séries temporais

Tutorial: executar e modificar um exemplo de consulta

O tutorial a seguir ajuda você a começar a usar o CloudWatch Logs Insights. Você executa uma consulta de amostra e vê como modificar e reexecutá-la.

Para executar uma consulta, você já deve ter registros armazenados em CloudWatch Registros. Se você já usa o CloudWatch Logs e tem grupos e fluxos de registros configurados, você está pronto para começar. Você também pode já ter registros se usar serviços como AWS CloudTrail o Amazon Route 53 ou o Amazon VPC e tiver configurado os registros desses serviços para CloudWatch acessar o Logs. Para obter mais informações sobre o envio de registros para o CloudWatch Logs, consulteIntrodução ao CloudWatch Logs.

As consultas no CloudWatch Logs Insights retornam um conjunto de campos de eventos de log ou 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 um exemplo de consulta

Para executar uma consulta de amostra do CloudWatch Logs Insights

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

2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Log Insights(Insights de log).

   Na página Logs Insights (Insights de log), o editor de consultas contém uma consulta padrão que retorna os 20 eventos de log mais recentes.

3. No menu suspenso Select log group(s) (Selecione grupo(s) de logs), escolha um ou mais grupos de logs a serem consultados.

   Se for uma conta de monitoramento na observabilidade CloudWatch entre contas, você poderá selecionar grupos de registros nas contas de origem e na conta de monitoramento. Uma única consulta pode consultar logs de diferentes contas ao mesmo tempo.

   Você pode filtrar os grupos de log por nome de grupo de log, ID de conta ou rótulo de conta.

   Quando você seleciona um grupo de registros na classe de registros padrão, o CloudWatch Logs Insights detecta automaticamente os campos de dados no grupo. Para ver esses campos descobertos, selecione o menu Fields (Campos) próximo ao canto superior direito da página.

   > **Note**
   >
   > Os campos descobertos são compatíveis somente com grupos de registros na classe de registros padrão. Para obter mais informações sobre classes de log, consulteClasses de log.

4. (Opcional) Use o seletor de tempo para selecionar o período que você deseja consultar.

   Você pode escolher intervalos de 5 a 30 minutos, intervalos de 1, 3 e 12 horas ou um período personalizado.

5. Selecione Run (Executar) para visualizar os resultados.

   Para este tutorial, os resultados incluem os 20 eventos de log adicionados mais recentemente.

   CloudWatch Os registros exibem um gráfico de barras dos eventos de registro no grupo de registros ao longo do tempo. O gráfico de barras mostra não apenas os eventos na tabela. mas também a distribuição de eventos no grupo de logs correspondente à consulta e ao intervalo de tempo.

6. Para ver todos os campos de um evento de log retornado, escolha o ícone suspenso triangular à esquerda do evento numerado.

Modificar o exemplo de consulta

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**
>
> Alguns exemplos de consultas fornecidos com o CloudWatch Logs Insights usam `tail` comandos `head` ou em vez `delimit`. 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 Salve e execute novamente as consultas do CloudWatch Logs Insights.

Adicionar um comando de filtro ao exemplo de consulta

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 nos quais 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, usamos awsRegion como o campo de filtro, mas você pode usar um campo diferente caso esse campo não esteja 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. Digite **`filter awsRegion="us-east-1"`**.
5. Escolha 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 O Logs Insights inclui poderosos recursos de consulta, incluindo vários comandos e suporte para expressões regulares, operações matemáticas e estatísticas. Para ter mais informações, consulte CloudWatch Sintaxe de consulta do Logs Insights.

Tutorial: Executar uma consulta com uma função de agregação

Você pode usar funções de agregação no comando `stats` e como argumentos de outras funções. Neste tutorial, você executará um comando de consulta que conta o número de eventos de log contendo um campo especificado. O comando de consulta retorna uma contagem total agrupada pelo valor ou valores do campo especificado. Para obter mais informações sobre funções de agregação, consulte Operações e funções suportadas no Guia do usuário do Amazon CloudWatch Logs.

Executar uma consulta com uma função de agregação

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Log Insights(Insights de log).
3. No menu suspenso Select log group(s) (Selecione grupo(s) de logs), escolha um ou mais grupos de logs a serem consultados.

   Se for uma conta de monitoramento na observabilidade CloudWatch entre contas, você poderá selecionar grupos de registros nas contas de origem e na conta de monitoramento. Uma única consulta pode consultar logs de diferentes contas ao mesmo tempo.

   Você pode filtrar os grupos de log por nome de grupo de log, ID de conta ou rótulo de conta.

   Quando você seleciona um grupo de CloudWatch registros, o Logs Insights detecta automaticamente os campos de dados no grupo de registros se for um grupo de registros de classe padrão. Para ver esses campos descobertos, selecione o menu Fields (Campos) próximo ao canto superior direito da página.
4. Exclua a consulta padrão no editor de consultas e insira o seguinte comando:

   ```
   stats count(*) by fieldName
   ```
5. Substitua *fieldName* por um campo descoberto do menu Fields (Campos).

   O menu Campos está localizado no canto superior direito da página e exibe todos os campos descobertos que o CloudWatch Logs Insights detecta em seu grupo de registros.
6. Selecione Run (Executar) para visualizar os resultados da consulta.

   Os resultados da consulta mostram o número de registros no grupo de logs que correspondem ao comando de consulta e a contagem total agrupada pelo valor ou valores do campo especificado.

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, é possível visualizar os resultados como gráfico

de barras, gráfico de pizza, grafo de linhas ou grafo de área empilhada. Isso ajuda a visualizar as tendências em seus logs de forma mais eficiente.

Para executar uma consulta de visualização

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Log Insights(Insights de log).
3. No menu suspenso Select log group(s) (Selecione grupo(s) de logs), escolha um ou mais grupos de logs a serem consultados.

   Se for uma conta de monitoramento na observabilidade CloudWatch entre contas, você poderá selecionar grupos de registros nas contas de origem e na conta de monitoramento. Uma única consulta pode consultar logs de diferentes contas ao mesmo tempo.

   Você pode filtrar os grupos de log por nome de grupo de log, ID de conta ou rótulo de conta.
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 grafo de linhas, um grafo de áreas empilhadas, um gráfico de pizza ou gráfico de barras. Isso ajuda a visualizar as tendências em eventos de log ao longo do tempo de modo mais eficiente.

Para executar uma consulta de visualização

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Log Insights(Insights de log).
3. No menu suspenso Select log group(s) (Selecione grupo(s) de logs), escolha um ou mais grupos de logs a serem consultados.

   Se for uma conta de monitoramento na observabilidade CloudWatch entre contas, você poderá selecionar grupos de registros nas contas de origem e na conta de monitoramento. Uma única consulta pode consultar logs de diferentes contas ao mesmo tempo.

   Você pode filtrar os grupos de log por nome de grupo de log, ID de conta ou rótulo de conta.
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 registro no grupo de registros que foram recebidos pelo CloudWatch Logs em 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, de pizza ou de áreas empilhadas, escolha a seta ao lado de Line (Linha) no canto superior direito do grafo.

Logs compatíveis e campos descobertos

CloudWatch O Logs Insights oferece suporte a diferentes tipos de registros. Para cada registro enviado para um grupo de registros da classe Standard Amazon CloudWatch Logs, o CloudWatch Logs Insights gera automaticamente cinco campos do sistema:

- `@message` contém o evento de log bruto não avaliado. Isso é o equivalente ao `message` campo em InputLogevent.
- `@timestamp` contém o timestamp do evento contido no campo `timestamp` do evento de log. Isso é o equivalente ao `timestamp` campo em InputLogevent.
- `@ingestionTime`contém a hora em que o CloudWatch Logs recebeu o evento de log.

- `@logStream` contém o nome do fluxo de logs ao qual o evento de log foi adicionado. Os fluxos de logs agrupam logs pelo mesmo processo que os gerou.
- `@log` é um identificador de grupo de logs na forma de *`account-id`*`:`*`log-group-name`*. Ao consultar vários grupos de logs, isso pode ser útil para identificar a que grupo de logs um determinado evento pertence.

> **Note**
>
> A descoberta de campo é suportada somente para grupos de registros na classe de registros padrão. Para obter mais informações sobre classes de log, consulteClasses de log.

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

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

Para outros tipos de registros com campos que o CloudWatch Logs Insights não descobre automaticamente, você pode usar o `parse` comando para extrair e criar campos extraídos para uso nessa consulta. Para ter mais informações, consulte CloudWatch Sintaxe de consulta do Logs Insights.

Se o nome de um campo de registro 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
Logs de fluxo do Amazon VPC	`@timestamp` , `@logStream` , `@message`, `accountId` , `endTime`, `interfaceId` , `logStatus` , `startTime` , `version`, `action`, `bytes`, `dstAddr`, `dstPort`, `packets`, `protocol`, `srcAddr`, `srcPort`
Logs do Route 53	`@timestamp` , `@logStream` , `@message`, `edgeLocation` , `ednsClientSubnet` , `hostZoneId` , `protocol`, `queryName` , `queryTimestamp` , `queryType` , `resolverIp` , `responseCode` , `version`

Tipo de log	Campos de log descobertos
Logs do Lambda	`@timestamp` , `@logStream` , `@message`, `@requestId` , `@duration`, `@billedDuration` , `@type`, `@maxMemoryUsed` , `@memorySize` Se uma linha de log do Lambda contiver um ID de rastreamento do X-Ray, ela também incluirá os seguintes campos: `@xrayTraceId` e `@xraySegmentId` . CloudWatch O Logs Insights descobre automaticamente os campos de log nos registros do Lambda, mas somente 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 ter mais informações, consulte Campos em logs JSON.
CloudTrail troncos Logs em formato JSON	Para obter mais informações, consulte Campos em logs JSON.
Outros tipos de log	`@timestamp` , `@ingestionTime` , `@logStream` , `@message`, `@log`.

Campos em logs JSON

Com o CloudWatch Logs Insights, você usa a notação de pontos para representar campos JSON. Esta seção contém um exemplo de evento JSON e trecho de código que mostra como você pode acessar campos JSON usando notação de ponto.

Exemplo: evento JSON

```
{
    "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": "192.0.2.255",
    "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"
                    }
                }
            ]
        }
    }
}
```

O exemplo de evento JSON contém um objeto chamado `userIdentity`. `userIdentity` contém um campo chamado `type`. Para representar o valor de `type` com notação de ponto, você usa `userIdentity.type`.

O exemplo de evento JSON contém matrizes que se nivelam em listas de nomes e valores de campos aninhados. Para representar o valor de `instanceId` para o primeiro item em `requestParameters.instancesSet`, use `requestParameters.instancesSet.items.0.instanceId`. O número `0` que é colocado antes do campo `instanceID` refere-se à posição dos valores para o campo `items`. O exemplo a seguir contém um trecho de código que mostra como você pode acessar campos JSON aninhados em um evento de log JSON.

Exemplo: consulta

```
fields @timestamp, @message
| filter requestParameters.instancesSet.items.0.instanceId="i-abcde123"
| sort @timestamp desc
```

O trecho de código mostra uma consulta que usa notação de ponto com o comando `filter` para acessar o valor do campo JSON aninhado `instanceId`. A consulta filtra as mensagens para as quais o valor de `instanceId` é igual a `"i-abcde123"` e retorna todos os eventos de log que contêm o valor especificado.

> **Note**
>
> CloudWatch O Logs Insights pode extrair no máximo 200 campos de eventos de log de um log JSON. Para os campos adicionais que não são extraídos, você pode usar o comando `parse` para extrair os campos do evento de log não analisado bruto no campo de mensagem. Para obter mais informações sobre o `parse` comando, consulte Sintaxe de consulta no Guia do CloudWatch usuário da Amazon.

CloudWatch Sintaxe de consulta do Logs Insights

Com o CloudWatch Logs Insights, você usa uma linguagem de consulta para consultar seus grupos de registros. A sintaxe de consulta é compatível com diferentes funções e operações que incluem, mas não se limitam a funções gerais, operações aritméticas e de comparação e expressões regulares.

Para criar consultas contendo vários comandos, separe-os com o caractere de barra vertical (|).

Para criar consultas contendo comentários, marque-os com o caractere de grade (#).

> **Note**
>
> CloudWatch O Logs Insights descobre automaticamente campos para diferentes tipos de registro e gera campos que começam com o caractere @. Para obter mais informações sobre esses campos, consulte Registros suportados e campos descobertos no Guia CloudWatch do usuário da Amazon.

A tabela a seguir descreve resumidamente cada comando. Em seguida, há uma descrição mais abrangente de cada comando com exemplos.

> **Note**
>
> Todos os comandos de consulta do CloudWatch Logs Insights são compatíveis com grupos de registros na classe de registro Standard. Os grupos de registros na classe de registro de Acesso Infrequente oferecem suporte a todos os comandos de consulta`pattern`, exceto`diff`, e. `unmask`

`display`	Exibe um ou mais campos específicos em resultados de consultas.
`fields`	Exibe campos específicos em resultados de consultas e oferece suporte a funções e operações que podem ser usadas para modificar valores de campo e criar novos campos para uso em consultas.
`filter`	Filtra a consulta para retornar apenas os eventos de log que correspon dem a uma ou mais condições.
`pattern`	Agrupa automaticamente seus dados de log em padrões. Um padrão é uma estrutura de texto compartilhada que se repete entre seus campos de registro. CloudWatch O Logs Insights fornece maneiras de analisar os padrões encontrados em seus eventos de registro. Para ter mais informações, consulte Análise de padrões.
`diff`	Compara os eventos de registro encontrados no período solicitado com os eventos de registro de um período anterior de igual duração, para que

	você possa procurar tendências e descobrir se determinados eventos de registro são novos.
parse	Extrai dados de um campo de log para criar um campo extraído que pode ser processado na consulta. **parse** é compatível tanto com o modo glob usando curingas quanto com as expressões regulares.
sort	Exibe os eventos de log retornados em ordem crescente (`asc`) ou decrescente (`desc`).
stats	Calcule estatísticas agregadas usando valores nos campos de log.
limit	Especifica um número máximo de eventos de log que você deseja que sua consulta retorne. Útil com **sort** para retornar resultados “20 principais” ou “20 mais recentes”.
dedup	Remove resultados duplicados com base em valores específicos em campos que você especifica.
unmask	Exibe todo o conteúdo de um evento de log que tenha algum conteúdo mascarado devido à política de proteção de dados. Para mais informaçõ es sobre a proteção de dados nos grupos de logs, acesse Ajude a proteger dados de log confidenciais com mascaramento.
Outras operações e funções	CloudWatch O Logs Insights também oferece suporte a muitas funções e operações de comparação, aritmética, data e hora, numéricas, seqüências de caracteres, endereços IP e funções e operações gerais.

As seções a seguir fornecem mais detalhes sobre os comandos de consulta do CloudWatch Logs Insights.

Tópicos

- display
- fields
- filtrar
- pattern
- diferença

- parse
- sort
- stats
- limite
- dedup
- unmask
- Funções booleanas, de comparação, numéricas e de data e hora, entre outras
- Campos contendo caracteres especiais
- Usar aliases e comentários em consultas

display

Use `display` para mostrar um ou mais campos específicos nos resultados da consulta.

O comando `display` mostra apenas os campos especificados. Se sua consulta contiver vários comandos `display`, os resultados da consulta mostram apenas o campo ou os campos que você especificou no comando `display` final.

Exemplo: Exibir um campo

O trecho de código mostra um exemplo de uma consulta que usa o comando parse para extrair dados de `@message` para criar os campos extraídos `loggingType` e `loggingMessage`. A consulta retorna eventos de log em que os valores para `loggingType` são ERROR. `display` mostra somente os valores para `loggingMessage` nos resultados da consulta.

```
fields @message
| parse @message "[*] *" as loggingType, loggingMessage
| filter loggingType = "ERROR"
| display loggingMessage
```

> **Tip**
>
> Use `display` somente uma vez em uma consulta. Se você usar `display` mais de uma vez em uma consulta, os resultados da consulta mostram apenas os campos especificados na última ocorrência em que o comando `display` for usado.

fields

Use `fields` para mostrar campos específicos nos resultados da consulta.

Se sua consulta contiver vários comandos `fields` e não incluir um comando `display`, os resultados exibirão todos os campos especificados nos comandos `fields`.

Exemplo: Exibir campos específicos

O exemplo a seguir mostra uma consulta que retorna 20 eventos de logs e os exibe em ordem decrescente. Os valores para `@timestamp` e `@message` são mostrados nos resultados da consulta.

```
fields @timestamp, @message
| sort @timestamp desc
| limit 20
```

Use `fields` em vez de `display` quando não quiser usar as diferentes funções e operações compatíveis com o `fields` para modificar valores de campo e criar novos campos que podem ser usados em consultas.

É possível usar o comando `fields` com a palavra-chave as para criar campos extraídos que usam os campos e funções de seus eventos de log. Por exemplo, `fields ispresent as isRes` cria um campo extraído chamado `isRes`, e o campo extraído pode ser usado no restante de sua consulta.

filtrar

Use `filter` para obter eventos de log que correspondam a uma ou mais condições.

Exemplo: Filtrar eventos de log usando uma condição

O trecho de código mostra um exemplo de uma consulta que retorna todos os eventos de log em que o valor de `range` é maior do que 3000. A consulta limita os resultados a 20 eventos de log e classifica os eventos de registros por `@timestamp` e em ordem decrescente.

```
fields @timestamp, @message
| filter (range>3000)
| sort @timestamp desc
| limit 20
```

Exemplo: Filtrar eventos de log usando mais de uma condição

Você pode usar as palavras-chave `and` e `or` para combinar mais de uma condição.

O trecho de código mostra um exemplo de consulta que retorna todos os eventos de log em que o valor de `range` é maior do que 3.000 e o valor de `accountId` é igual a 123.456.789.012. A consulta limita os resultados a 20 eventos de log e classifica os eventos de registros por `@timestamp` e em ordem decrescente.

```
fields @timestamp, @message
| filter (range>3000 and accountId=123456789012)
| sort @timestamp desc
| limit 20
```

Correspondências e expressões regulares no comando de filtro

O comando filtrar é compatível com o uso de expressões regulares. Você pode usar os seguintes operadores de comparação (=, `!=`, <, <=, >, >=) e operadores boolianos (`and`, `or` e `not`).

Você pode usar a palavra-chave `in` para testar a associação do conjunto e verificar se há elementos em uma matriz. Para verificar elementos em uma matriz, coloque a matriz após `in`. É possível usar o operador booliano `not` com `in`. Você pode criar consultas que usam `in` para retornar eventos de log nos quais os campos são correspondências de string. Os campos devem ser strings completas. Por exemplo, o trecho de código a seguir mostra uma consulta que usa `in` para retornar eventos de log nos quais o campo `logGroup` é a string `example_group` completa.

```
fields @timestamp, @message
| filter logGroup in ["example_group"]
```

Você pode usar as frases de palavras-chave `like` e `not like` para combinar substrings.
Você pode usar o operador de expressão regular =~ para combinar substrings. Para fazer a correspondência de uma substring com `like` e `not like`, envolva entre aspas duplas ou simples a substring que deseja corresponder. Você pode usar padrões de expressão regular com `like` e `not like`. Para fazer a correspondência de uma substritng com o operador de expressão regular, envolva entre barras a substring a ser correspondida. Os exemplos a seguir contêm trechos de código que mostram como você pode fazer a correspondências de substrings usando o comando `filter`.

Exemplos: correspondência de substrings

Os exemplos a seguir retornam eventos de log quando `f1` contém a palavra Exception. Os três exemplos fazem distinção entre maiúsculas e minúsculas.

O primeiro exemplo faz a correspondência de uma substring com `like`.

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

O segundo exemplo faz a correspondência de uma substring com `like` e um padrão de expressão regular.

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

O terceiro exemplo faz a correspondência de uma substring com uma expressão regular.

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

Exemplo: fazer a correspondência de substrings com curingas

Você pode usar o símbolo de ponto (`.`) como um curinga em expressões regulares para fazer a correspondência com substrings. No exemplo a seguir, a consulta retorna correspondências em que o valor de `f1` começa com a string `ServiceLog`.

```
fields f1, f2, f3
| filter f1 like /ServiceLog./
```

Você pode colocar um símbolo de asterisco depois do ponto (`.*`) para criar um quantificador ganancioso que retorna o maior número possível de correspondências. No exemplo a seguir, a consulta retorna correspondências em que o valor de `f1` não somente começa com a string `ServiceLog`, mas também inclui a string `ServiceLog`.

```
fields f1, f2, f3
| filter f1 like /ServiceLog.*/
```

As correspondências possíveis podem ser formatadas da seguinte forma:

- `ServiceLogSampleApiLogGroup`
- `SampleApiLogGroupServiceLog`

Exemplo: excluir substrings das correspondências

O exemplo a seguir mostra uma consulta que retorna eventos de log quando `f1` não contém a palavra Exception. O exemplo diferencia minúsculas e maiúsculas.

```
fields f1, f2, f3
| filter f1 not like "Exception"
```

Exemplo: fazer a correspondência de subcadeias de caracteres com padrões que não diferenciam maiúsculas e minúsculas

Você pode fazer a correspondência de subcadeias de caracteres que não diferenciam minúsculas e maiúsculas com `like` e expressões regulares. Coloque o seguinte parâmetro (?i) antes da substring a ser correspondida. O exemplo a seguir mostra uma consulta que retorna eventos de log quando `f1` contém a palavra Exception ou exception.

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

pattern

Use `pattern` para agrupar automaticamente seus dados de log em padrões.

Um padrão é uma estrutura de texto compartilhada que se repete entre seus campos de log. Você pode usar `pattern` para revelar tendências emergentes, monitorar erros conhecidos e identificar linhas de registro que ocorrem com frequência ou são de alto custo. CloudWatch O Logs Insights também fornece uma experiência de console que você pode usar para encontrar e analisar melhor os padrões em seus eventos de log. Para ter mais informações, consulte Análise de padrões.

Como o `pattern` comando identifica automaticamente padrões comuns, você pode usá-lo como ponto de partida para pesquisar e analisar seus registros. Você também pode combinar `pattern` com os comandos `filter`, `parse`, ou `sort` para identificar padrões em consultas mais ajustadas.

Entrada do comando pattern

O comando `pattern` espera uma das seguintes entradas: o campo `@message`, um campo extraído criado usando o comando `parse` ou uma string manipulada usando uma ou mais Funções string.

Saída do comando pattern

O comando `pattern` produzirá a seguinte saída:

- `@pattern`: uma estrutura de texto compartilhada que se repete entre seus campos de eventos de log. Os campos que variam dentro de um padrão, como um ID de solicitação ou um carimbo de data/hora, são representados por `<*>`. Por exemplo, `[INFO] Request time: <*> ms` é uma saída potencial para a mensagem de log `[INFO] Request time: 327 ms`.
- `@ratio`: a proporção de eventos de log de um período selecionado e grupos de registros especificados que correspondem a um padrão identificado. Por exemplo, se metade dos eventos de log nos grupos de logs e no período de tempo selecionados corresponderem ao padrão, `@ratio` retornará `0.50`
- `@sampleCount`: o número de eventos de log de um período selecionado e grupos de logs especificados que correspondem a um padrão identificado.
- `@severityLabel`: a severidade ou o nível do log que indica o tipo de informação contida em um log. Por exemplo, `Error`, `Warning`, `Info` ou `Debug`.

Exemplos

O comando a seguir identifica logs com estruturas semelhantes em grupos de logs especificados no intervalo de tempo selecionado, agrupando-os por padrão e contagem

```
pattern @message
```

O comando `pattern` pode ser usado em combinação com o comando `filter`

```
filter @message like /ERROR/
| pattern @message
```

O comando `pattern` pode ser usado com os comandos `parse` e `sort`

```
filter @message like /ERROR/
| parse @message 'Failed to do: *' as cause
| pattern cause
| sort @sampleCount asc
```

diferença

Compara os eventos de registro encontrados no período solicitado com os eventos de registro de um período anterior de igual duração. Dessa forma, você pode procurar tendências e descobrir se eventos de log específicos são novos.

Adicione um modificador ao `diff` comando para especificar o período com o qual você deseja comparar:

- `diff`compara os eventos de log no intervalo de tempo atualmente selecionado com os eventos de log do intervalo de tempo imediatamente anterior.
- `diff previousDay`compara os eventos de log no intervalo de tempo atualmente selecionado com os eventos de log da mesma hora do dia anterior.
- `diff previousWeek`compara os eventos de log no intervalo de tempo atualmente selecionado com os eventos de log do mesmo horário da semana anterior.
- `diff previousMonth`compara os eventos de log no intervalo de tempo atualmente selecionado com os eventos de log da mesma época do mês anterior.

Para ter mais informações, consulte Compare (diff) com intervalos de tempo anteriores.

parse

Use `parse` para extrair dados de um campo de log e criar um campo extraído que pode ser processado na consulta. **`parse`** é compatível tanto com o modo glob usando curingas quanto com as expressões regulares. Para obter informações sobre a sintaxe de expressões regulares, consulteSintaxe de expressões regulares (regex) compatíveis.

Você pode analisar campos JSON aninhados com uma expressão regular.

Exemplo: análise de um campo JSON aninhado

O trecho de código mostra como analisar um evento de logs JSON que foi nivelado durante a ingestão.

```
{'fieldsA': 'logs', 'fieldsB': [{'fA': 'a1'}, {'fA': 'a2'}]}
```

O trecho de código mostra uma consulta com uma expressão regular que extrai os valores de `fieldsA` e `fieldsB` para criar os campos extraídos `fld` e `array`.

```
parse @message "'fieldsA': '*', 'fieldsB': ['*']" as fld, array
```

Grupos de captura nomeados

Quando você usa **`parse`** com uma expressão regular, pode usar grupos de captura nomeados para capturar um padrão para um campo. A sintaxe é `parse @message (?<Name>pattern).`

O exemplo a seguir usa um grupo de captura em um log de fluxo da VPC para extrair o ENI para um campo denominado `NetworkInterface`.

```
parse @message /(?<NetworkInterface>eni-.*?) / display @timestamp, NetworkInterface
```

> **Note**
>
> Os eventos de logs JSON são nivelados durante a ingestão. Atualmente, a análise de campos JSON aninhados com uma expressão global não é suportada. Você só pode analisar eventos de logs JSON que incluam no máximo 200 campos de eventos de logs. Ao analisar campos JSON aninhados, você deve formatar a expressão regular em sua consulta para que corresponda ao formato do seu evento de logs JSON.

Exemplos do comando de análise

Use uma expressão de glob para extrair os campos **`@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 **`@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 **`loggingTime`**, **`loggingType`** e **`loggingMessage`**, aplica o filtro para eventos de logs que contêm strings **`ERROR`** ou **`INFO`** e exibe apenas os campos **`loggingMessage`** e **`loggingType`** para eventos que contêm uma string **`ERROR`**.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

sort

Use `sort` para exibir eventos de log em ordem crescente (`asc`) ou decrescente (`desc`) por um campo especificado. Use-o com o comando `limit` para criar consultas de “N principais” ou “N inferiores”.

O algoritmo de classificação é uma versão atualizada da classificação natural. Se você classificar em ordem crescente, a lógica a seguir será usada.

- Todos os valores não numéricos vêm antes de todos os valores numéricos. Valores numéricos são valores que incluem somente números, não uma mistura de números e outros caracteres.
- Para valores não numéricos, o algoritmo agrupa caracteres numéricos consecutivos e caracteres alfabéticos consecutivos em partes separadas para comparação. Ele ordena porções não numéricas por seus valores Unicode e ordena porções numéricas por seu comprimento primeiro e depois por seu valor numérico.

Para obter mais informações sobre a ordem Unicode, consulte Lista de caracteres Unicode.

Por exemplo, o seguinte é o resultado de uma classificação em ordem crescente.

```
!: >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> sorted by unicode order
#
*%04
0# >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Alphanumeric starting with numbers
5A
111A   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>  Starts with more digits than 5A, so it sorted
 to be later than 5A
2345_
@ >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> 2345 is compared with @ in the unicode order,
@_
A >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Values starting with letters
A9876fghj
a12345hfh
0 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> Number values
01
1
2
3
```

Se você classificar em ordem decrescente, os resultados da classificação serão inversos.

Por exemplo, a consulta a seguir de logs de fluxo da Amazon VPC localiza as 15 principais transferências de pacotes entre hosts.

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

stats

Use `stats` para criar visualizações dos seus dados de log, como gráficos de barras, gráficos de linhas e gráficos de áreas empilhadas. Isso ajuda você a identificar padrões em seus dados de registro com mais eficiência. CloudWatch O Logs Insights gera visualizações para consultas que usam a `stats` função e uma ou mais funções de agregação.

Por exemplo, a consulta a seguir em um grupo de logs do Route 53 retorna visualizações que mostram a distribuição dos registros do Route 53 por hora, por tipo de consulta.

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

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.

As unidades e abreviações de tempo a seguir são compatíveis com a função `bin`. Para todas as unidades e abreviações que incluem mais de um caractere, é permitido adicionar "s" para formar o plural. Assim, ambos `hr` e `hrs` funcionam para especificar horas.

- `millisecond` `ms` `msec`
- `second` `s` `sec`
- `minute` `m` `min`
- `hour` `h` `hr`
- `day` `d`
- `week` `w`
- `month` `mo` `mon`
- `quarter` `q` `qtr`
- `year` `y` `yr`

Tópicos

- Visualizar dados de séries temporais
- Visualizar dados de log agrupados por campos
- Use vários comandos de estatísticas em uma única consulta
- Funções para uso com estatísticas

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.
- A consulta usa a função `bin()` para agrupar os dados por um campo.

Essas consultas podem produzir gráficos de linha, gráficos de áreas empilhadas, 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”.

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”.

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.

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”. 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
```

Use vários comandos de estatísticas em uma única consulta

Você pode usar até dois comandos `stats` em uma única consulta. Isso permite executar uma agregação adicional na saída da primeira agregação.

Exemplo: consulta com dois comandos **`stats`**

Por exemplo, a consulta a seguir localiza primeiro o volume total de tráfego em compartimentos de cinco minutos e, em seguida, calcula o volume de tráfego mais alto, mais baixo e médio entre esses compartimentos de cinco minutos.

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length)/1024/1024 as logs_mb BY bin(5m)
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
        avg(logs_mb) AS avg_ingest_mb
```

Exemplo: combine vários comandos de estatísticas com outras funções, como **`filter`**, **`fields`** e **`bin`**

Você pode combinar dois comandos `stats` com outros comandos, como `filter` e `fields`, em uma única consulta. Por exemplo, a consulta a seguir localiza o número de endereços IP distintos nas sessões e localiza o número de sessões por plataforma do cliente, filtra esses endereços IP e, por fim, localiza a média de solicitações de sessão por plataforma do cliente.

```
STATS count_distinct(client_ip) AS session_ips,
      count(*) AS requests BY session_id, client_platform
| FILTER session_ips > 1
| STATS count(*) AS multiple_ip_sessions,
        sum(requests) / count(*) AS avg_session_requests BY client_platform
```

Você pode usar as funções `bin` e `dateceil` em consultas com vários comandos `stats`. Por exemplo, a consulta a seguir primeiro combina mensagens em blocos de cinco minutos, depois agrega esses blocos de cinco minutos em blocos de dez minutos e calcula os volumes de tráfego mais altos, mais baixos e médios dentro de cada bloco de dez minutos.

```
FIELDS strlen(@message) AS message_length
| STATS sum(message_length) / 1024 / 1024 AS logs_mb BY BIN(5m) as @t
| STATS max(logs_mb) AS peak_ingest_mb,
        min(logs_mb) AS min_ingest_mb,
```

```
        avg(logs_mb) AS avg_ingest_mb BY dateceil(@t, 10m)
```

Observações e limitações

Uma consulta pode ter no máximo dois comandos `stats`. Não é possível alterar esta cota.

Se você usar um comando `sort` ou `limit`, ele deverá aparecer após o segundo comando `stats`. Se estiver antes do segundo comando `stats`, a consulta não é válida.

Quando uma consulta tem dois comandos `stats`, a exibição dos resultados parciais da consulta só começará quando a primeira agregação `stats` for concluída.

No segundo comando `stats` de uma única consulta, você pode se referir somente aos campos definidos no primeiro comando `stats`. Por exemplo, a consulta a seguir não é válida porque o campo `@message` não estará disponível após a primeira agregação `stats`.

```
FIELDS @message
| STATS SUM(Fault) by Operation
# You can only reference `SUM(Fault)` or Operation at this point
| STATS MAX(strlen(@message)) AS MaxMessageSize # Invalid reference to @message
```

Todos os campos que você referenciar após o primeiro comando `stats` devem ser definidos nesse primeiro comando `stats`.

```
STATS sum(x) as sum_x by y, z
| STATS max(sum_x) as max_x by z
# You can only reference `max(sum_x)`, max_x or z at this point
```

Important

A função `bin` sempre usa implicitamente o campo `@timestamp`. Isso significa que você não pode usar `bin` no segundo comando `stats` sem usar o primeiro comando `stats` para propagar o campo `timestamp`. Por exemplo, a consulta a seguir não é válida.

```
FIELDS strlen(@message) AS message_length
 | STATS sum(message_length) AS ingested_bytes BY @logStream
 | STATS avg(ingested_bytes) BY bin(5m) # Invalid reference to @timestamp field
```

Em vez disso, defina o campo `@timestamp` no primeiro comando `stats` e, em seguida, você poderá usá-lo com `dateceil` no segundo comando `stats`, como no exemplo a seguir.

```
FIELDS strlen(@message) AS message_length
 | STATS sum(message_length) AS ingested_bytes, max(@timestamp) as @t BY
 @logStream
 | STATS avg(ingested_bytes) BY dateceil(@t, 5m)
```

Funções para uso com estatísticas

CloudWatch O Logs Insights suporta funções de agregação de estatísticas e funções de não agregação de estatísticas.

Use funções de agregação de estatísticas no comando `stats` e como argumentos de outras funções.

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

`count()`
 	
`count(fieldName: LogField)`
	número	Faz a contagem dos eventos de log. `count()` (ou `count(*)`) conta todos os eventos retornados pela consulta, e o `count(fieldName)` conta todos os registros que incluam o nome do campo especificado.

`count_distinct(fieldName: LogField)`	número	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.
`min(fieldName: LogField)`	LogFieldValue	O mínimo dos valores desse campo de log nos logs consultados.

Função	Tipo de resultado	Descrição
`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)`	número	O desvio padrão dos valores no campo especificado.
`sum(fieldName: NumericLogField)`	número	A soma dos valores no campo especificado.

Funções de estatísticas de não agregação

Use 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.

limite

Use `limit` para especificar o número de eventos de log que você deseja que sua consulta retorne.

Por exemplo, o exemplo a seguir retorna apenas os 25 eventos de log mais recentes

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

dedup

Use `dedup` para remover resultados duplicados com base em valores específicos em campos que você especifica. É possível usar `dedup` com um ou mais campos. Se você especificar um campo com `dedup`, apenas um evento de log será retornado para cada valor exclusivo desse campo. Se especificar vários campos, um evento de log será retornado para cada combinação exclusiva de valores desses campos.

As duplicatas serão descartadas em ordem de classificação, e apenas o primeiro resultado dessa ordem será mantido. Convém classificar os resultados antes de os submeter ao comando `dedup`. Se os resultados não forem classificados antes de serem executados por `dedup`, a ordem de classificação decrescente padrão `@timestamp` será usada.

Valores nulos não são considerados duplicatas para avaliação. Eventos de log com valores nulos para qualquer um dos campos especificados são mantidos. Para eliminar campos com valores nulos, use **`filter`** com a função `isPresent(field)`.

O único comando de consulta que você pode usar em uma consulta após o comando `dedup` é `limit`.

Exemplo: visualizar somente o evento de log mais recente de cada valor exclusivo do campo chamado **`server`**

O exemplo a seguir mostra os campos `timestamp`, `server`, `severity` e `message` somente para o evento mais recente de cada valor exclusivo de `server`.

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server
```

Para ver mais exemplos de consultas do CloudWatch Logs Insights, consulte Consultas gerais.

unmask

Use `unmask` para exibir todo o conteúdo de um evento de log que tenha algum conteúdo mascarado devido à política de proteção de dados. Para usar este comando, você deve ter a permissão `logs:Unmask`.

Para mais informações sobre a proteção de dados nos grupos de logs, acesse Ajude a proteger dados de log confidenciais com mascaramento.

Funções booleanas, de comparação, numéricas e de data e hora, entre outras

CloudWatch O Logs Insights oferece suporte a muitas outras operações e funções em consultas, conforme explicado nas seções a seguir.

Tópicos

- Operadores aritméticos
- Operadores booleanos
- Operadores de comparação
- Operadores numéricos
- Funções de data e hora
- Funções gerais
- Funções de string de endereço IP
- Funções de string

Operadores aritméticos

Os operadores aritméticos aceitam tipos de dados numéricos como argumentos e retornam resultados numéricos. Use operadores aritméticos nos comandos `filter` e `fields` e como argumentos de outras funções.

Operation	Descrição
`a + b`	Adição
`a - b`	Subtração

Operation	Descrição
`a * b`	Multiplicação
`a / b`	Divisão
`a ^ b`	Exponenciação (`2 ^ 3` retorna 8)
`a % b`	Resto ou módulo (`10 % 3` retorna 1)

Operadores booleanos

Use os operadores booleanos **`and`**, **`or`** e **`not`**.

> **Note**
>
> Use operadores booleanos somente em funções que retornam um valor de TRUE ou FALSE.

Operadores de comparação

Os operadores de comparação aceitam todos os tipos de dados como argumentos e retornam um resultado booliano. Use operadores de comparação no comando `filter` e como argumentos de outras funções.

Operador	Descrição
=	Equal
!=	Not equal
<	Menor que
>	Maior que
<=	Menor ou igual a
>=	Maior ou igual a

Operadores numéricos

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

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

Funções de data e hora

Funções de data e hora

Use funções de data e hora nos comandos `fields` e `filter` e como argumentos de outras funções. Use essas funções para criar buckets de tempo para consultas com funções de agregação. Use períodos de tempo que consistam em um número e um dos seguintes:

- `ms`por milissegundos

- spor segundos
- mpor minutos
- hpor horas

Por exemplo, 10m é 10 minutos, e 1h é uma hora.

> **Note**
>
> Use a unidade de tempo mais apropriada para sua função de data e hora. CloudWatch Os registros limitam sua solicitação de acordo com a unidade de tempo que você escolher. Por exemplo, ele limita 60 como o valor máximo para qualquer solicitação que usas. Então, se você especificarbin(300s), o CloudWatch Logs realmente implementa isso como 60 segundos, porque 60 é o número de segundos em um minuto, então o CloudWatch Logs não usará um número maior que 60 coms. Para criar um bucket de 5 minutos, use bin(5m) em vez disso.
> O limite para ms é 1000, os limites para s e m são 60 e o limite para h é 24.

A tabela a seguir contém uma lista das diferentes funções de data e hora que você pode usar nos comandos de consulta. A tabela lista o tipo de resultado de cada função e contém uma descrição de cada função.

> **Tip**
>
> Quando você cria um comando de consulta, é possível usar o seletor de intervalo de tempo para selecionar um período que deseja consultar. Por exemplo, você pode definir um período entre intervalos de 5 e 30 minutos; intervalos de 1, 3 e 12 horas; ou um período personalizado. Você também pode definir períodos entre datas específicas.

Função	Tipo de resultado	Descrição
`bin(period: Period)`	Timestamp	Arredonda o valor de `@timestamp` para o período indicado e trunca. Por exemplo, `bin(5m)` arredonda o valor de `@timestamp` para os 5 minutos mais próximos.

Função	Tipo de resultado	Descrição

		Você pode usar isso para agrupar várias entradas de log em uma consulta. O seguinte exemplo conta a quantidade de exceções por hora:
		
		`filter @message like /Exception/`
		`    | stats count(*) as exceptionCount`
		` by bin(1h)`
		`    | sort exceptionCount desc`
		
		As unidades e abreviações de tempo a seguir são compatíveis com a função `bin`. Para todas as unidades e abreviações que incluem mais de um caractere, é permitido adicionar "s" para formar o plural. Assim, ambos `hr` e `hrs` funcionam para especificar horas.
		
		• `millisecond ms msec`
		• `second s sec`
		• `minute m min`
		• `hour h hr`
		• `day d`
		• `week w`
		• `month mo mon`
		• `quarter q qtr`
		• `year y yr`

`datefloor(timestamp: Timestamp, period: Period)`	Timestamp	Trunca o time stamp para o período indicado. Por exemplo, `datefloor(@timestamp, 1h)` trunca todos os valores de `@timestamp` no final.

Função	Tipo de resultado	Descrição
`dateceil(timestamp : Timestamp, period: Period)`	Timestamp	Arredonda o time stamp para o período indicado e trunca. Por exemplo, `dateceil( @timestamp, 1h)` trunca todos os valores de `@timestamp` no início.
`fromMillis(fieldNa me: number)`	Timestamp	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)`	número	Converte o time stamp encontrado no campo nomeado em um número que representa os milissegundos desde a epoch do Unix. Por exemplo, `toMillis(@timestamp)` converte o carimbo de data/hora `2022-01-1 4T13:18:031.000-08:00` para `1642195111000` .

> **Note**
>
> Atualmente, o CloudWatch Logs Insights não oferece suporte à filtragem de registros com carimbos de data/hora legíveis por humanos.

Funções gerais

Funções gerais

Use funções gerais nos comandos `fields` e `filter` e como argumentos de outras funções.

Função	Tipo de resultado	Descrição
`ispresent(fieldName: LogField)`	Booleano	Retorna `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 de endereço IP

Funções de string de endereço IP

Use 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)`	booleano	Retorna `true` se o campo for um endereço IPv4 ou IPv6 válido.
`isValidIpV4(fieldName: string)`	boolean	Retorna `true` se o campo for um endereço IPv4 válido.
`isValidIpV6(fieldName: string)`	boolean	Retorna `true` se o campo for um endereço IPv6 válido.
`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` , onde `192.0.2.0` ou`2001:db8::` é o início do bloco CIDR.
`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,

Função	Tipo de resultado	Descrição
		como `192.0.2.0/24` , em que `192.0.2.0` é o início do bloco CIDR.
`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` , em que `2001:db8::` é o início do bloco CIDR.

Funções de string

Funções de string

Use funções de string nos comandos `fields` e `filter` e como argumentos de outras funções.

Função	Tipo de resultado	Descrição
`isempty(fieldName: string)`	Número	Retornará 1 se o campo não for encontrado ou for uma string vazia.
`isblank(fieldName: string)`	Número	Retornará 1 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, trimChars: string)`
	string	Se a função não tiver um segundo argumento, ela removerá os caracteres em branco da esquerda da string. Se a função tiver um segundo

Função	Tipo de resultado	Descrição
		argumento de string, ela não removerá os caracteres em branco. Em vez disso, remove os caracteres em `trimChars` à esquerda de `str`. Por exemplo, `ltrim("xy ZxyfooxyZ","xyZ")` retorna `"fooxyZ"`.

`rtrim(str: string)`
 	
`rtrim(str: string, trimChars: string)`
	string	Se a função não tiver um segundo argumento, ela removerá os caracteres em branco da direita da string. Se a função tiver um segundo argumento de string, ela não removerá os caractere s em branco. Em vez disso, remove os caracteres de `trimChars` à direita de `str`. Por exemplo, `rtrim("xy ZfooxyxyZ","xyZ")` retorna `"xyZfoo"`.

Função	Tipo de resultado	Descrição

`trim(str: string)`
 	
`trim(str: string, trimChars: string)`
	string	Se a função não tiver um segundo argumento, ela removerá os caracteres em branco nas duas extremidades da string. Se a função tiver um segundo argumento de string, ela não removerá os caracteres em branco. Em vez disso, remove os caracteres de `trimChars` de ambos os lados de `str`. Por exemplo, `trim("xyZxyfooxyxyZ","xyZ")` retorna `"foo"`.

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

Função	Tipo de resultado	Descrição
`replace(fieldName: string, searchValue: string, replaceValue: string)`	string	Substitui todas as instâncias de `searchValue` em `fieldName: string` por `replaceValue`. Por exemplo, a função `replace(logGroup,"smoke_test","Smoke")` pesquisa eventos de log em que o campo `logGroup` contém o valor da string `smoke_test` e substitui o valor pela string `Smoke`.
`strcontains(str: string, searchValue: string)`	número	Retornará 1 se `str` contiver `searchValue` e 0, do contrário.

Campos contendo caracteres especiais

Se um campo contiver caracteres não alfanuméricos que não sejam o `@` símbolo ou o ponto (`.`), você deverá cercar o campo com caracteres de crase (). ` Por exemplo, o campo de log `foo-bar` deve estar entre acentos graves (`` `foo-bar` ``) porque contém um caractere não alfanumérico, o hífen (`-`).

Usar aliases e comentários em consultas

Crie consultas que contenham aliases. Use aliases para renomear campos de log ou ao extrair valores em campos. Use a palavra-chave `as` para dar um campo de log ou resultar em um alias. Você pode usar mais de um alias em uma consulta. Você pode usar aliases nos comandos a seguir:

- `fields`
- `parse`
- `sort`
- `stats`

Os exemplos a seguir mostram como criar consultas que contenham aliases.

Exemplo

A consulta contém um alias no comando `fields`.

```
fields @timestamp, @message, accountId as ID
| sort @timestamp desc
| limit 20
```

A consulta retorna os valores dos campos `@timestamp`, `@message` e `accountId`. Os resultados são classificados em ordem decrescente e limitados a 20. Os valores para `accountId` estão listados sob o alias `ID`.

Exemplo

A consulta contém um alias nos comandos `sort` e `stats`.

```
stats count(*) by duration as time
| sort time desc
```

A consulta conta o número de vezes que o campo `duration` ocorre no grupo de logs e classifica os resultados em ordem decrescente. Os valores para `duration` estão listados sob o alias `time`.

Usar comentários

CloudWatch O Logs Insights oferece suporte a comentários em consultas. Use o caractere jogo da velha (#) para separar comentários. Você pode usar comentários para ignorar linhas em consultas ou consultas de documentos.

Exemplo: consulta

Quando a consulta a seguir é executada, a segunda linha é ignorada.

```
fields @timestamp, @message, accountId
# | filter accountId not like "7983124201998"
| sort @timestamp desc
| limit 20
```

Análise de padrões

CloudWatch O Logs Insights usa algoritmos de aprendizado de máquina para encontrar padrões quando você consulta seus registros. Um padrão é uma estrutura de texto compartilhada que se repete entre seus campos de registro. Ao visualizar os resultados de uma consulta, você pode escolher a guia Padrões para ver os padrões que o CloudWatch Logs encontrou com base em uma amostra dos seus resultados. Como alternativa, você pode acrescentar o `pattern` comando à sua consulta para analisar os padrões em todo o conjunto de eventos de log correspondentes.

Os padrões são úteis para analisar grandes conjuntos de registros porque um grande número de eventos de registro geralmente pode ser compactado em alguns padrões.

Considere a seguinte amostra de três eventos de log.

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for resource id 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for resource id 324892398123-12345
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for resource id 3ff231242342-12345
```

No exemplo anterior, todos os três eventos de log seguem um padrão:

```
<*> <*> [INFO] Calling DynamoDB to store for resource id <*>
```

Os campos dentro de um padrão são chamados de tokens. Os campos que variam dentro de um padrão, como ID de solicitação ou carimbo de data/hora, são tokens dinâmicos. Cada token dinâmico é representado pelo `<*>` momento em que o CloudWatch Logs o exibe.

Exemplos comuns de tokens dinâmicos incluem códigos de erro, registros de data e hora e IDs de solicitação. Um valor de token representa um valor específico de um token dinâmico. Por exemplo, se um token dinâmico representa um código de erro HTTP, o valor de um token pode ser`501`.

A detecção de padrões também é usada no detector de anomalias CloudWatch Logs e nos recursos de comparação. Para obter mais informações, consulte Detecção de anomalias de log e Compare (diff) com intervalos de tempo anteriores.

Introdução à análise de padrões

A detecção de padrões é realizada automaticamente em qualquer consulta do CloudWatch Logs Insights. As consultas que não incluem o `pattern` comando obtêm eventos e padrões de log nos resultados.

Se você incluir o `pattern` comando em sua consulta, a análise de padrões será executada em todo o conjunto correspondente de eventos de log. Isso fornece resultados de padrões mais precisos, mas os eventos de log brutos não são retornados quando você usa o `pattern` comando. Quando uma consulta não inclui`pattern`, os resultados do padrão são baseados nos primeiros 1000 eventos de log retornados ou no valor limite que você usou na sua consulta. Se você incluir `pattern` na consulta, os resultados exibidos na guia Padrões serão derivados de todos os eventos de registro correspondentes à consulta.

Para começar com a análise de padrões no CloudWatch Logs Insights

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

   Na página Logs Insights (Insights de log), o editor de consultas contém uma consulta padrão que retorna os 20 eventos de log mais recentes.
3. Remova a `| limit 20` linha na caixa de consulta para que a consulta tenha a seguinte aparência:

   ```
   fields @timestamp, @message, @logStream, @log
   | sort @timestamp desc
   ```
4. No menu suspenso Selecionar grupo (s) de log, escolha um ou mais grupos de log para consultar.
5. (Opcional) Use o seletor de tempo para selecionar o período que você deseja consultar.

   Você pode escolher entre intervalos de 5 a 30 minutos; intervalos de 1 hora, 3 horas e 12 horas; ou um período de tempo personalizado.
6. Escolha Executar consulta para iniciar a consulta.

   Quando a consulta termina de ser executada, a guia Registros exibe uma tabela de eventos de registro retornados pela consulta. Acima da tabela, há uma mensagem sobre quantos registros corresponderam à consulta, semelhante a Mostrar 1000 dos 71.101 registros correspondidos.
7. Escolha a guia Padrões.
8. A tabela agora exibe os padrões encontrados na consulta. Como a consulta não incluiu o `pattern` comando, essa guia exibe somente os padrões descobertos entre os 1000 eventos de log que foram mostrados na tabela na guia Registros.

   Para cada padrão, as seguintes informações são exibidas:

- O padrão, com cada token dinâmico exibido como<*>.
- A contagem de eventos, que é o número de vezes que o padrão apareceu nos eventos de registro consultados. Escolha o título da coluna Contagem de eventos para classificar os padrões por frequência.
- A taxa de eventos, que é a porcentagem dos eventos de log consultados que contêm esse padrão.
- O tipo de severidade, que será um dos seguintes:
  - ERRO se o padrão contiver a palavra Erro.
  - AVISE se o padrão contiver a palavra Avisar, mas não contiver Erro.
  - INFORMAÇÕES se o padrão não contiver aviso ou erro.

  Escolha o título da coluna Informações de gravidade para classificar os padrões por severidade.

9. Agora, altere a consulta. Substitua a `| sort @timestamp desc` linha na consulta por`| pattern @message`, para que a consulta completa seja a seguinte:

```
fields @timestamp, @message, @logStream, @log
| pattern @message
```

10. Selecione Executar consulta.

    Quando a consulta é concluída, não há resultados na guia Registros. No entanto, a guia Padrões provavelmente tem um número maior de padrões listados, dependendo do número total de eventos de log que foram consultados.

11. Independentemente de você ter incluído `pattern` na sua consulta, você pode inspecionar ainda mais os padrões que a consulta retorna. Para fazer isso, escolha o ícone na coluna Inspecionar para um dos padrões.

    O painel Inspeção de padrões aparece e exibe o seguinte:

    - O padrão. Selecione um token dentro do padrão para analisar os valores desse token.
    - Um histograma mostrando o número de ocorrências do padrão no intervalo de tempo consultado. Isso pode ajudá-lo a identificar tendências interessantes, como um aumento repentino na ocorrência de um padrão.
    - A guia Amostras de registro exibe alguns dos eventos de registro que correspondem ao padrão selecionado.

- A guia Valores do Token exibe os valores do token dinâmico selecionado, se você tiver selecionado um.

  > **Note**
  >
  > Um máximo de 10 valores de token são capturados para cada token. A contagem de tokens pode não ser precisa. CloudWatch O Logs usa um contador probabilístico para gerar a contagem de tokens, não o valor absoluto.

- A guia Padrões relacionados exibe outros padrões que freqüentemente ocorreram quase ao mesmo tempo que o padrão que você está inspecionando. Por exemplo, se um padrão para uma `ERROR` mensagem geralmente era acompanhado por outro evento de registro marcado como `INFO` com detalhes adicionais, esse padrão é exibido aqui.

Detalhes sobre o comando pattern

Esta seção contém mais detalhes sobre o `pattern` comando e seus usos.

- No tutorial anterior, removemos o `sort` comando quando o adicionamos `pattern` porque uma consulta não é válida se incluir um `pattern` comando após um `sort` comando. É válido ter um `pattern` antes de um`sort`.

  Para obter mais detalhes sobre a `pattern` sintaxe, consultepattern.
- Quando você usa `pattern` em uma consulta, `@message` deve ser um dos campos selecionados no `pattern` comando.
- Você pode incluir o `filter` comando antes de um `pattern` comando para fazer com que somente o conjunto filtrado de eventos de log seja usado como entrada para análise de padrões.
- Para ver os resultados do padrão para um campo específico, como um campo derivado do `parse` comando, use`pattern @fieldname`.
- Consultas com saída que não seja de log, como consultas com o `stats` comando, não retornam resultados padrão.

Compare (diff) com intervalos de tempo anteriores

Você pode usar o CloudWatch Logs Insights para comparar as alterações em seus eventos de registro ao longo do tempo. Você pode comparar os eventos de registro ingeridos durante um intervalo de tempo recente com os registros do período imediatamente anterior. Como alternativa, você pode comparar com períodos anteriores semelhantes. Isso pode ajudá-lo a descobrir se um erro nos seus registros foi introduzido recentemente ou já estava ocorrendo, além de ajudar a encontrar outras tendências.

As consultas de comparação retornam somente padrões nos resultados, não eventos de log brutos. Os padrões retornados ajudarão você a ver rapidamente as tendências e mudanças nos eventos de registro ao longo do tempo. Depois de executar uma consulta de comparação e obter os resultados do padrão, você pode ver exemplos de eventos de log brutos dos padrões nos quais está interessado. Para obter mais informações sobre padrões de log, consulteAnálise de padrões.

Quando você executa uma consulta de comparação, sua consulta é analisada em relação a dois períodos diferentes: o período de consulta original que você seleciona e o período de comparação. O período de comparação é sempre igual ao período de consulta original. Os intervalos de tempo padrão para as comparações são os seguintes.

- Período anterior — compara ao período imediatamente anterior ao período de consulta.
- Dia anterior — compara ao período de um dia antes do período de consulta.
- Semana anterior — compara ao período de uma semana antes do período de consulta.
- Mês anterior — compara ao período de um mês antes do período de consulta.

> **Note**
>
> As consultas que usam comparações geram cobranças semelhantes às da execução de uma única consulta do CloudWatch Logs Insights no intervalo de tempo combinado. Para obter mais informações, consulte Amazon CloudWatch Pricing.

Para executar uma consulta de comparação

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

Uma consulta padrão aparece na caixa de consulta.

3. Mantenha a consulta padrão ou insira uma consulta diferente.
4. No menu suspenso Selecionar grupo (s) de log, escolha um ou mais grupos de log para consultar.
5. (Opcional) Use o seletor de tempo para selecionar o período que você deseja consultar. A consulta padrão é para a hora anterior de dados de registro.
6. No seletor de intervalo de tempo, escolha Comparar. Em seguida, escolha o período anterior com o qual você deseja comparar os registros originais e escolha Aplicar.
7. Selecione Executar consulta.

   Para fazer com que a consulta busque os dados do período de comparação, o `diff` comando é anexado à sua consulta.
8. Escolha a guia Padrões para ver os resultados.

   A tabela exibe as seguintes informações:

   - Cada padrão, com partes variáveis do padrão substituídas pelo símbolo dinâmico do token<*>. Para ter mais informações, consulte Análise de padrões.
   - A contagem de eventos é o número de eventos de log com esse padrão no período original e mais atual.
   - A contagem de eventos de diferença é a diferença entre o número de eventos de log correspondentes no período atual e o período de comparação. Uma diferença positiva significa que há mais eventos desse tipo no período atual.
   - A descrição da diferença resume brevemente a mudança nesse padrão entre o período atual e o período de comparação.
   - O tipo de gravidade é a gravidade provável dos eventos de registros com esse padrão, com base nas palavras encontradas nos eventos de registro`FATAL`, como`ERROR`, e. `WARN`
9. Para inspecionar ainda mais um dos padrões na lista, escolha o ícone na coluna Inspecionar para um dos padrões.

   O painel Inspeção de padrões aparece e exibe o seguinte:

   - O padrão. Selecione um token dentro do padrão para analisar os valores desse token.

- Um histograma mostrando o número de ocorrências do padrão no intervalo de tempo consultado. Isso pode ajudá-lo a identificar tendências interessantes, como um aumento repentino na ocorrência de um padrão.
- A guia Amostras de registro exibe alguns dos eventos de registro que correspondem ao padrão selecionado.
- A guia Valores do Token exibe os valores do token dinâmico selecionado, se você tiver selecionado um.

> **Note**
>
> Um máximo de 10 valores de token são capturados para cada token. A contagem de tokens pode não ser precisa. CloudWatch O Logs usa um contador probabilístico para gerar a contagem de tokens, não o valor absoluto.

- A guia Padrões relacionados exibe outros padrões que freqüentemente ocorreram quase ao mesmo tempo que o padrão que você está inspecionando. Por exemplo, se um padrão para uma `ERROR` mensagem geralmente era acompanhado por outro evento de registro marcado como `INFO` com detalhes adicionais, esse padrão é exibido aqui.

Consultas de exemplo

Esta seção contém uma lista de comandos de consulta gerais e úteis que você pode executar no CloudWatch console. Para obter informações sobre como executar um comando de consulta, consulte Tutorial: Executar e modificar uma consulta de amostra no Guia do usuário do Amazon CloudWatch Logs.

Para obter mais informações sobre a sintaxe de consulta, consulteCloudWatch Sintaxe de consulta do Logs Insights.

Tópicos

- Consultas gerais
- Consultas de logs do Lambda
- Consultas para logs de fluxo da Amazon VPC
- Consultas de logs do Route 53
- Consultas para registros CloudTrail
- Consultas para Amazon API Gateway

- Consultas para gateway NAT
- Consultas para logs do servidor Apache
- Consultas para a Amazon EventBridge
- Exemplos do comando de análise

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/
```

Obtenha o evento de log mais recente para cada valor exclusivo do campo **`server`**.

```
fields @timestamp, server, severity, message
| sort @timestamp asc
| dedup server
```

Obtenha o evento de log mais recente para cada valor exclusivo do campo **`server`** para cada tipo **`severity`**.

```
fields @timestamp, server, severity, message
| sort @timestamp desc
| dedup server, severity
```

Consultas de logs do Lambda

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)
```

Procure invocações de funções lentas e elimine solicitações duplicadas que podem surgir de novas tentativas ou código no lado do cliente. Nessa consulta, **`@duration`** está em milissegundos.

```
fields @timestamp, @requestId, @message, @logStream
| filter @type = "REPORT" and @duration > 1000
| sort @timestamp desc
| dedup @requestId
| limit 20
```

Consultas para logs de fluxo da Amazon 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
```

Encontre um único registro para cada conexão para ajudar a solucionar problemas de conectividade com a rede.

```
fields @timestamp, srcAddr, dstAddr, srcPort, dstPort, protocol, bytes
| filter logStream = 'vpc-flow-logs' and interfaceId = 'eni-0123456789abcdef0'
| sort @timestamp desc
| dedup srcAddr, dstAddr, srcPort, dstPort, protocol
| limit 20
```

Consultas de logs do Route 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 para registros CloudTrail

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

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

Encontre os hosts do Amazon EC2 que foram iniciados ou parados em uma determinada AWS região.

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

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

```
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
```

Encontre entradas de log em que TLS 1.0 ou 1.1 foi usado

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by userIdentity.accountId, recipientAccountId,
 eventSource, eventName, awsRegion, tlsDetails.tlsVersion, tlsDetails.cipherSuite,
 userAgent
| sort eventSource, eventName, awsRegion, tlsDetails.tlsVersion
```

Encontre o número de chamadas por serviço que usaram o TLS versões 1.0 ou 1.1

```
filter tlsDetails.tlsVersion in [ "TLSv1", "TLSv1.1" ]
| stats count(*) as numOutdatedTlsCalls by eventSource
| sort numOutdatedTlsCalls desc
```

Consultas para Amazon API Gateway

Encontre os 10 últimos erros 4XX

```
fields @timestamp, status, ip, path, httpMethod
| filter status>=400 and status<=499
| sort @timestamp desc
| limit 10
```

Identifique as 10 Amazon API Gateway solicitações mais antigas em seu grupo de registros de Amazon API Gateway acesso

```
fields @timestamp, status, ip, path, httpMethod, responseLatency
| sort responseLatency desc
| limit 10
```

Retorne a lista dos caminhos de API mais populares em seu grupo de registros de Amazon API Gateway acesso

```
stats count(*) as requestCount by path
| sort requestCount desc
| limit 10
```

Crie um relatório de latência de integração para seu grupo de registros de Amazon API Gateway acesso

```
filter status=200
| stats avg(integrationLatency), max(integrationLatency),
min(integrationLatency) by bin(1m)
```

Consultas para gateway NAT

Se você notar custos mais altos do que o normal em sua AWS fatura, você pode usar o CloudWatch Logs Insights para encontrar os principais contribuidores. Para obter mais informações sobre os seguintes comandos de consulta, consulte Como posso encontrar os principais contribuidores para o tráfego por meio do gateway NAT na minha VPC? na página de suporte AWS premium.

> **Note**
>
> Nos comandos de consulta a seguir, substitua “x.x.x.x” pelo IP privado do gateway NAT e substitua “y.y” pelos dois primeiros octetos do intervalo CIDR da VPC.

Encontre as instâncias que estão enviando mais tráfego por meio de seu gateway NAT.

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Determine o tráfego de/para as instâncias em seus gateways NAT.

```
filter (dstAddr like 'x.x.x.x' and srcAddr like 'y.y.') or (srcAddr like 'xxx.xx.xx.xx'
 and dstAddr like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Determine os destinos da Internet com os quais as instâncias em sua VPC se comunicam com mais frequência para uploads e downloads.

Para uploads

```
filter (srcAddr like 'x.x.x.x' and dstAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Para downloads

```
filter (dstAddr like 'x.x.x.x' and srcAddr not like 'y.y.')
| stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
| sort bytesTransferred desc
| limit 10
```

Consultas para logs do servidor Apache

Você pode usar o CloudWatch Logs Insights para consultar os registros do servidor Apache. Para obter mais informações sobre as consultas a seguir, consulte Simplificando os registros do servidor Apache com CloudWatch o Logs Insights no blog AWS Cloud Operations & Migrations.

Encontre os campos mais relevantes para que você possa revisar seus logs de acesso e verificar se há tráfego no caminho /admin da aplicação.

```
fields @timestamp, remoteIP, request, status, filename| sort @timestamp desc
| filter filename="/var/www/html/admin"
| limit 20
```

Encontre o número de solicitações GET exclusivas que acessaram sua página principal com o código de status “200” (sucesso).

```
fields @timestamp, remoteIP, method, status
| filter status="200" and referrer= http://34.250.27.141/ and method= "GET"
| stats count_distinct(remoteIP) as UniqueVisits
| limit 10
```

Encontre o número de vezes que o serviço Apache foi reiniciado.

```
fields @timestamp, function, process, message
| filter message like "resuming normal operations"
| sort @timestamp desc
| limit 20
```

Consultas para a Amazon EventBridge

Obtenha o número de EventBridge eventos agrupados por tipo de detalhe do evento

```
fields @timestamp, @message
| stats count(*) as numberOfEvents by `detail-type`
| sort numberOfEvents desc
```

Exemplos do comando de análise

Use uma expressão de glob para extrair os campos **`@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 **`@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 **`loggingTime`**, **`loggingType`** e **`loggingMessage`**, aplica o filtro para eventos de logs que contêm strings **`ERROR`** ou **`INFO`** e exibe apenas os campos **`loggingMessage`** e **`loggingType`** para eventos que contêm uma string **`ERROR`**.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

Visualize dados de log em grafos

Você pode usar visualizações como gráficos de barras, gráficos de linhas e gráficos de áreas empilhadas para identificar com mais eficiência padrões em seus dados de registro. CloudWatch O Logs Insights gera visualizações para consultas que usam a `stats` função e uma ou mais funções de agregação. Para obter mais informações, consulte stats.

Salve e execute novamente as consultas do CloudWatch Logs Insights

Depois de criar uma consulta, você pode salvá-la para que possa ser executada novamente mais tarde. As consultas são salvas em uma estrutura de pastas para que você possa organizá-las. Você pode salvar até 1.000 consultas 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 CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Log Insights(Insights de log).
3. No editor de consultas, crie uma consulta.
4. Escolha Save(Salvar).

   Se você não vê o botão Salvar, você precisa mudar para o novo design do console de CloudWatch registros. 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 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 Save(Salvar).

> **Tip**
>
> Você pode criar uma pasta para consultas salvas usando `PutQueryDefinition`. Para criar uma pasta para as consultas salvas, use uma barra (/) para prefixar o nome da consulta desejada com o nome da pasta desejada: <*`folder-name`*>/<*`query-name`*>. Para obter mais informações sobre essa ação, consulte PutQueryDefinition.

Como executar uma consulta salva

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Log Insights(Insights de log).

3. À direita, escolha Consultas.
4. Selecione sua consulta na lista de Consultas salvas. Ela aparece no editor de consulta.
5. Escolha Executar.

Como salvar uma nova versão de uma consulta salva

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

Adicionar consulta ao painel ou exportar os resultados da consulta

Depois de executar uma consulta, você pode adicionar a consulta a um CloudWatch painel 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 contam para seu limite de 30 consultas simultâneas do CloudWatch Logs Insights.

Para adicionar resultados da consulta a um painel

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Log Insights(Insights de log).
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 Criar novo a fim de 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 CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Log Insights(Insights de log).
3. Escolha um ou mais grupos de logs e execute uma consulta.
4. Escolha Exportar resultados e depois a opção desejada.

Exibir as 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 30 consultas simultâneas do CloudWatch Logs Insights por conta, incluindo consultas adicionadas aos painéis.

Para exibir o histórico de consultas recente

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Log Insights(Insights de log).
3. Escolha Histórico, se você estiver usando o novo design do console de CloudWatch registros. 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, CloudWatch Registros exibe Em andamento para todas as consultas em execução no momento.

Criptografe os resultados da consulta com AWS Key Management Service

Por padrão, o CloudWatch Logs criptografa os resultados armazenados de suas consultas do CloudWatch Logs Insights usando o método padrão de criptografia do lado do servidor do CloudWatch Logs. Em vez disso, você pode optar por usar uma AWS KMS chave para criptografar esses resultados. Se você associar uma AWS KMS chave aos resultados da criptografia, o CloudWatch Logs usará essa chave para criptografar os resultados armazenados de todas as consultas na conta.

Se, posteriormente, você desassociar a chave dos resultados da consulta, o CloudWatch Logs retornará ao método de criptografia padrão para consultas posteriores. Mas as consultas executadas enquanto a chave estava associada ainda são criptografadas com essa chave. CloudWatch Os registros ainda podem retornar esses resultados após a desassociação da chave KMS, porque CloudWatch os registros ainda podem continuar referenciando a chave. No entanto, se a chave for desativada posteriormente, o CloudWatch Logs não conseguirá ler os resultados da consulta que foram criptografados com essa chave.

> **Important**
>
> CloudWatch O Logs suporta somente chaves KMS simétricas. Não use uma chave assimétrica para criptografar os resultados da consulta. Para obter mais informações, consulte Usar chaves simétricas e assimétricas.

Limites

- Para executar as etapas a seguir, é necessário ter as seguintes permissões: `kms:CreateKey`, `kms:GetKeyPolicy` e `kms:PutKeyPolicy`.
- Após uma chave ser associada ou desassociada dos resultados da consulta, até cinco minutos poderão ser necessários para que a operação seja efetivada.
- Se você revogar o acesso do CloudWatch Logs a uma chave associada ou excluir uma chave KMS associada, seus dados criptografados no CloudWatch Logs não poderão mais ser recuperados.
- Você não pode usar o CloudWatch console para associar uma chave, é preciso usar a API AWS CLI ou CloudWatch Logs.

Etapa 1: criar um AWS KMS key

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

```
aws kms create-key
```

A saída contém o ID de chave e o nome do recurso da Amazon (ARN) da chave. 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 chave do KMS

Por padrão, todas as chaves do KMS são privadas. Somente o proprietário do recurso pode usá-la para criptografar e descriptografar dados. No entanto, o proprietário do recurso pode conceder permissões para acessar a chave a outros usuários e recursos. Com essa etapa, você concede permissão principal ao serviço de CloudWatch registros para usar a chave. Esse principal de serviço deve estar na mesma AWS região em que a chave está armazenada.

Como prática recomendada, recomendamos que você restrinja o uso da chave somente às AWS contas que você especificar.

Primeiro, salve a política padrão para sua chave KMS `policy.json` usando o seguinte get-key-policycomando:

```
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 `Condition` seções para aumentar a segurança da AWS KMS chave. Para ter mais informações, consulte AWS KMS chaves e contexto de criptografia.

A `Condition` seção neste exemplo limita o uso da AWS KMS chave para os resultados da consulta do CloudWatch Logs Insights na conta especificada.

```
{
 "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": {
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:logs:region:account_ID:query-result:*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "Your_account_ID"
                }
            }
        }
    ]
}
```

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

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

Etapa 3: associar uma chave do KMS aos resultados da consulta

Para associar a chave do KMS aos resultados da consulta na conta

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

```
aws logs associate-kms-key --resource-identifier "arn:aws:logs:region:account-id:query-result:*" --kms-key-id "key-arn"
```

Etapa 4: desassociar uma chave dos resultados da consulta na conta

Para desassociar a chave KMS associada aos resultados da consulta, use o seguinte disassociate-kms-keycomando:

```
aws logs disassociate-kms-key --resource-identifier "arn:aws:logs:region:account-id:query-result:*"
```

Use linguagem natural para gerar e atualizar consultas do CloudWatch Logs Insights

> **Note**
>
> Esse recurso geralmente está disponível no Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon) e Ásia-Pacífico (Tóquio) para CloudWatch registros.

CloudWatch O Logs oferece suporte a um recurso de consulta em linguagem natural para ajudar você a gerar e atualizar consultas para o CloudWatch Logs Insights e o CloudWatch Metrics Insights.

Com esse recurso, você pode fazer perguntas ou descrever os dados de CloudWatch registros que está procurando em inglês simples. O recurso de linguagem natural gera uma consulta com base em uma solicitação que você insere e fornece uma line-by-line explicação de como a consulta funciona. Você também pode atualizar a consulta para investigar melhor seus dados.

Dependendo do seu ambiente, você pode inserir solicitações como “Quais são os 100 principais endereços IP de origem por bytes transferidos?” e “Encontre as 10 solicitações mais lentas da função Lambda”.

Para gerar uma consulta do CloudWatch Logs Insights com esse recurso, abra o editor de consultas do CloudWatch Logs Insights, selecione o grupo de registros que você deseja consultar e escolha Gerar consulta.

> **Important**
>
> Para usar o recurso de consulta de linguagem natural, você deve usar a ReadOnlyAccesspolítica CloudWatchLogsFullAccessCloudWatchLogsReadOnlyAccess, AdministratorAccess,, ou.
> Você também pode incluir a ação `cloudwatch:GenerateQuery` em uma política nova ou atual gerenciada pelo cliente ou em uma política em linha.

Consultas de exemplo

Os exemplos nesta seção descrevem como gerar e atualizar consultas usando o recurso de linguagem natural.

> **Note**
>
> Para obter mais informações sobre o editor de consultas e a sintaxe do CloudWatch Logs Insights, consulte Sintaxe de consulta do CloudWatch Logs Insights.

Exemplo: gerar uma consulta em linguagem natural

Para gerar uma consulta usando linguagem natural, insira uma solicitação e escolha Gerar nova consulta. Este exemplo mostra uma consulta que executa uma pesquisa básica.

Prompt

Veja a seguir um exemplo de um prompt que direciona a capacidade de pesquisar as 10 invocações mais lentas da função Lambda.

```
Find the 10 slowest requests
```

Consulta

Veja a seguir um exemplo de consulta que o recurso de linguagem natural gera com base na solicitação. Observe como a solicitação aparece em um comentário antes da consulta. Depois da consulta, você pode ler uma explicação que descreve como a consulta funciona.

```
# Find the 10 slowest requests
```

```
fields @timestamp, @message, @duration
| sort @duration desc
| limit 10
# This query retrieves the timestamp, message and duration fields from the logs and
 sorts them in descending order by duration to find the 10 slowest requests.
```

> **Note**
>
> Para desativar o surgimento da solicitação e a explicação de como a consulta funciona, use o ícone de engrenagem no editor.

Exemplo: atualizar uma consulta em linguagem natural

Você pode atualizar uma consulta editando a solicitação inicial e escolhendo Atualizar consulta.

Solicitação atualizada

O exemplo a seguir mostra uma versão atualizada da solicitação anterior. Em vez de um prompt que pesquisa as 10 invocações mais lentas da função Lambda, esse prompt agora direciona a capacidade de pesquisar as 20 invocações mais lentas da função Lambda e incluir outra coluna para eventos de log adicionais.

```
Show top 20 slowest requests instead and display requestId as a column
```

Consulta atualizada

Veja a seguir um exemplo da consulta atualizada. Observe como a solicitação atualizada aparece em um comentário antes da consulta atualizada. Depois da consulta, você pode ler uma explicação que descreve como a consulta original foi atualizada.

```
# Show top 20 slowest requests instead and display requestId as a column
fields @timestamp, @message, @requestId, @duration
| sort @duration desc
| limit 20
# This query modifies the original query by replacing the @message field with the
 @requestId field and changing the limit from 10 to 20 to return the top 20 log events
 by duration instead of the top 10.
```

Optar por não usar seus dados para melhorar o serviço

Os dados das solicitações em linguagem natural que você fornece para treinar o modelo de IA e gerar consultas relevantes são usados exclusivamente para fornecer e manter seu serviço. Esses dados podem ser usados para melhorar a qualidade do CloudWatch Logs Insights. Sua confiança e privacidade, além da segurança do seu conteúdo, são nossas maiores prioridades. Para obter mais informações, consulte Termos de Serviço da AWS e AWS responsible AI policy.

Você pode se recusar a ter seu conteúdo usado para desenvolver ou melhorar a qualidade das consultas em linguagem natural ao criar uma política de rejeição de serviços de IA. Para desativar a coleta de dados de todos os recursos do CloudWatch Logs AI, incluindo o recurso de geração de consultas, você deve criar uma política de exclusão para o Logs. CloudWatch Para obter mais informações, consulte Políticas de exclusão dos serviços de IA no Guia do usuário do AWS Organizations .

Detecção de anomalias de log

Você pode criar um detector de anomalias de log para cada grupo de log. O detector de anomalias examina os eventos de registro ingeridos no grupo de registros e encontra anomalias nos dados de registro. A detecção de anomalias usa aprendizado de máquina e reconhecimento de padrões para estabelecer linhas de base do conteúdo típico de registros.

Depois de criar um detector de anomalias para um grupo de registros, ele treina usando as últimas duas semanas de eventos de registro no grupo de registros para treinamento. O período de treinamento pode levar até 15 minutos. Depois que o treinamento é concluído, ele começa a analisar os registros recebidos para identificar anomalias, e as anomalias são exibidas no console de CloudWatch registros para você examinar.

CloudWatch O reconhecimento de padrões de registros extrai padrões de registros identificando conteúdo estático e dinâmico em seus registros. Os padrões são úteis para analisar grandes conjuntos de registros porque um grande número de eventos de registro geralmente pode ser compactado em alguns padrões.

Por exemplo, veja o exemplo a seguir de três eventos de log.

```
2023-01-01 19:00:01 [INFO] Calling DynamoDB to store for resource id 12342342k124-12345
2023-01-01 19:00:02 [INFO] Calling DynamoDB to store for resource id 324892398123-12345
2023-01-01 19:00:03 [INFO] Calling DynamoDB to store for resource id 3ff231242342-12345
```

No exemplo anterior, todos os três eventos de log seguem um padrão:

```
<*> <*> [INFO] Calling DynamoDB to store for resource id <*>
```

Os campos dentro de um padrão são chamados de tokens. Os campos que variam dentro de um padrão, como ID de solicitação ou carimbo de data/hora, são chamados de tokens dinâmicos. Os tokens dinâmicos são representados por <*> quando o CloudWatch Logs exibe o padrão. Cada valor diferente encontrado para um token dinâmico é chamado de valor de token.

Exemplos comuns de tokens dinâmicos incluem códigos de erro, carimbos de data/hora e IDs de solicitação.

A detecção de anomalias de registros usa esses padrões para encontrar anomalias. Após o período de treinamento do modelo do detector de anomalias, os registros são avaliados em relação às tendências conhecidas. O detector de anomalias sinaliza flutuações significativas como anomalias.

A criação de detectores de anomalias de log não incorre em cobranças.

Gravidade e prioridade de anomalias e padrões

Cada anomalia encontrada por um detector de anomalias logarítmicas recebe uma prioridade. Cada padrão encontrado recebe uma severidade.

- A prioridade é calculada automaticamente e se baseia no nível de severidade do padrão e na quantidade de desvio dos valores esperados. Por exemplo, se um determinado valor de token aumentar repentinamente em 500%, essa anomalia pode ser designada como `HIGH` prioritária, mesmo que sua gravidade seja. `NONE`
- A severidade é baseada apenas em palavras-chave encontradas nos padrões`FATAL`, como`ERROR`, `WARN` e. Se nenhuma dessas palavras-chave for encontrada, a gravidade de um padrão será marcada como`NONE`.

Tempo de visibilidade da anomalia

Ao criar um detector de anomalias, você especifica o período máximo de visibilidade de anomalias para ele. Esse é o número de dias em que a anomalia é exibida no console e é retornada pela operação da ListAnomaliesAPI. Depois de decorrido esse período de tempo para uma anomalia, se ela continuar ocorrendo, ela é automaticamente aceita como um comportamento normal e o modelo do detector de anomalias para de sinalizá-la como uma anomalia.

Se você não ajustar o tempo de visibilidade ao criar um detector de anomalias, 21 dias serão usados como padrão.

Suprimindo uma anomalia

Depois que uma anomalia for encontrada, você pode optar por suprimi-la temporária ou permanentemente. A supressão de uma anomalia faz com que o detector de anomalias pare de sinalizar essa ocorrência como uma anomalia pelo período de tempo especificado. Ao suprimir uma anomalia, você pode optar por suprimir somente aquela anomalia específica ou suprimir todas as anomalias relacionadas ao padrão em que a anomalia foi encontrada.

Você ainda pode ver anomalias suprimidas no console. Você também pode optar por parar de suprimi-los.

Perguntas frequentes

AWS Usa meus dados para treinar algoritmos de aprendizado de máquina para AWS uso ou para outros clientes?

Não. O modelo de detecção de anomalias criado pelo treinamento é baseado nos eventos de registro em um grupo de registros e é usado somente nesse grupo de registros e nessa AWS conta.

Que tipos de eventos de log funcionam bem com a detecção de anomalias?

A detecção de anomalias de log é adequada para: registros de aplicativos e outros tipos de registros em que a maioria das entradas de registro se encaixa nos padrões típicos. Grupos de registros com eventos que contêm palavras-chave de nível de registro ou gravidade, como INFO, ERROR e DEBUG, são especialmente adequados para registrar a detecção de anomalias.

A detecção de anomalias de log não é adequada para: Registrar eventos com estruturas JSON extremamente longas, como CloudTrail Logs. A análise de padrões analisa somente até os primeiros 1500 caracteres de uma linha de registro, portanto, todos os caracteres além desse limite são ignorados.

Registros de auditoria ou acesso, como registros de fluxo de VPC, também terão menos sucesso com a detecção de anomalias. A detecção de anomalias serve para encontrar problemas de aplicativos, portanto, pode não ser adequada para anomalias de rede ou de acesso.

Para ajudá-lo a determinar se um detector de anomalias é adequado para um determinado grupo de CloudWatch registros, use a análise de padrões de registros para encontrar o número de padrões nos eventos de registro no grupo. Se o número de padrões não for maior do que cerca de 300, a detecção de anomalias pode funcionar bem. Para obter mais informações sobre análise de padrões, consulteAnálise de padrões.

O que é sinalizado como uma anomalia?

As seguintes ocorrências podem fazer com que um evento de log seja sinalizado como uma anomalia:

- Um evento de registro com um padrão nunca visto antes no grupo de registros.
- Uma variação significativa de um padrão conhecido.
- Um novo valor para um token dinâmico que tem um conjunto discreto de valores usuais.
- Uma grande mudança no número de ocorrências de um valor para um token dinâmico.

Embora todos os itens anteriores possam estar marcados como anomalias, nem todos significam que o aplicativo está funcionando mal. Por exemplo, higher-than-usual vários valores de `200` sucesso podem ser sinalizados como uma anomalia. Em casos como esse, você pode considerar a supressão dessas anomalias que não indicam problemas.

O que acontece com dados confidenciais que estão sendo mascarados?

Todas as partes dos eventos de registro que são mascaradas como dados confidenciais não são examinadas em busca de anomalias. Para obter mais informações sobre como mascarar dados confidenciais, consulte Ajudar a proteger dados de log confidenciais com mascaramento.

Ativar a detecção de anomalias em um grupo de registros

Use as etapas a seguir para usar o CloudWatch console para criar um detector de anomalias de log que escaneia um grupo de registros em busca de anomalias.

Você também pode criar detectores de anomalias programaticamente. Para obter mais informações, consulte CreateLogAnomalyDetector.

Para criar um detector de anomalias de log

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Escolha Registros, Registre Anomalias.
3. Escolha Criar detector de anomalias.
4. Selecione o grupo de registros para o qual criar esse detector de anomalias.
5. Insira um nome para o detector em Nome do detector de anomalias.
6. (Opcional) Altere a frequência de avaliação do padrão de 5 minutos. Defina esse valor de acordo com a frequência com que o grupo de registros recebe novos registros. Por exemplo, se o grupo de registros receber novos eventos de registro em lotes a cada 10 minutos, talvez seja apropriado definir a frequência de avaliação para 15 minutos.
7. (Opcional) Para configurar o detector de anomalias para procurar anomalias somente em eventos de log que contenham determinadas palavras ou cadeias de caracteres, escolha Filtrar padrões.

   Em seguida, insira um padrão em Padrão do filtro de detecção de anomalias. Para obter mais informações sobre a sintaxe de padrões,Sintaxe de padrões de filtros para filtros de métricas, filtros de assinatura, filtros de eventos de log e Live Tail..

(Opcional) Para testar seu padrão de filtro, insira algumas mensagens de registro em Mensagens de eventos de registro e escolha Padrão de teste.

8. (Opcional) Para alterar o período de visibilidade da anomalia do padrão ou para associar uma AWS KMS chave a esse detector de anomalias, escolha Configuração avançada.

   a. Para alterar o período de visibilidade da anomalia do padrão, insira um novo valor em Período máximo de visibilidade da anomalia (dias).

   b. Para associar uma AWS KMS chave a esse detector de anomalias, insira o ARN no ARN da chave KMS. Se você atribuir uma chave, as informações de anomalia encontradas por esse detector serão criptografadas em repouso com a chave. Os usuários devem ter permissões para essa chave e para que o detector de anomalias recupere informações sobre as anomalias encontradas.

      Você também deve garantir que o responsável pelo serviço de CloudWatch registros tenha permissão para usar a chave. Para ter mais informações, consulte Criptografe um detector de anomalias e seus resultados com AWS KMS.

9. Escolha Ativar detecção de anomalias.

   O detector de anomalias é criado e começa a treinar seu modelo, com base nos eventos de registro que o grupo de registros está ingerindo. Após cerca de 15 minutos, a detecção de anomalias está ativa e começa a encontrar anomalias à superfície.

Exibir anomalias que foram encontradas

Depois de criar um ou mais detectores de anomalias de log, você pode usar o CloudWatch console para ver as anomalias que eles encontraram.

Você pode visualizar anomalias programaticamente. Para obter mais informações, consulte ListAnomalies.

Para visualizar as anomalias encontradas por todos os seus detectores de anomalias de log

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Escolha Registros, Registre Anomalias.

A tabela de anomalias de registros é exibida. O número na parte superior ao lado de Registro de anomalias mostra quantas anomalias de registro estão listadas na tabela. Cada linha na tabela exibe as seguintes informações:

- A coluna Anomalia exibe um breve resumo da anomalia. Esses resumos são gerados pelo CloudWatch Logs.
- A prioridade da anomalia. A prioridade é calculada automaticamente com base na quantidade de alterações nos eventos de registro, em palavras-chave como `Exception` ocorrência em um evento de registro e muito mais.
- O padrão de log no qual a anomalia se baseia. Para obter mais informações sobre padrões, consulteDetecção de anomalias de log.
- A tendência do registro de anomalias exibe um histograma que descreve o volume de registros que correspondem ao padrão.
- A hora da última detecção exibe a hora mais recente em que essa anomalia foi encontrada.
- A primeira hora de detecção mostra a primeira vez em que essa anomalia foi encontrada.
- O detector de anomalias exibe o nome do grupo de registros contendo os eventos de registro relacionados a essa anomalia. Você pode escolher esse nome para ver a página de detalhes do grupo de registros.

3. Para inspecionar ainda mais uma anomalia, escolha o botão de rádio em sua fileira.

   O painel Inspeção de padrões aparece e exibe o seguinte:

   - O padrão no qual essa anomalia se baseia. Selecione um token dentro do padrão para analisar os valores desse token.
   - Um histograma mostrando o número de ocorrências da anomalia no intervalo de tempo consultado.
   - A guia Amostras de registro exibe alguns dos eventos de registro que fazem parte da anomalia.
   - A guia Valores do Token exibe os valores do token dinâmico selecionado, se você tiver selecionado um.

> **Note**
>
> Um máximo de 10 valores de token são capturados para cada token. A contagem de tokens pode não ser precisa. CloudWatch O Logs usa um contador probabilístico para gerar a contagem de tokens, não o valor absoluto.

4. Para suprimir uma anomalia, escolha o botão de rádio em sua linha e faça o seguinte:

   a. Escolha Ações, Suprimir anomalia.

   b. Em seguida, especifique por quanto tempo você deseja que a anomalia seja suprimida.

   c. Para suprimir todas as anomalias relacionadas a esse padrão, selecione Suprimir padrão.

   d. Escolha Suprimir anomalia.

Para visualizar as anomalias encontradas em um único grupo de registros

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. Escolha Logs, Grupos de logs.
3. Escolha o nome de um grupo de registros e, em seguida, escolha a guia Detecção de anomalias.

   A tabela de detecção de anomalias é exibida. O número na parte superior ao lado de Registro de anomalias mostra quantas anomalias de registro estão listadas na tabela. Cada linha na tabela exibe as seguintes informações:

   - A coluna Anomalia exibe um breve resumo da anomalia. Esses resumos são gerados pelo CloudWatch Logs.
   - A prioridade da anomalia. A prioridade é calculada automaticamente com base na quantidade de alterações nos eventos de registro, em palavras-chave como `Exception` ocorrência em um evento de registro e muito mais.
   - O padrão de log no qual a anomalia se baseia. Para obter mais informações sobre padrões, consulteDetecção de anomalias de log.
   - A tendência do registro de anomalias exibe um histograma que descreve o volume de registros que correspondem ao padrão.
   - A hora da última detecção exibe a hora mais recente em que essa anomalia foi encontrada.
   - A primeira hora de detecção mostra a primeira vez em que essa anomalia foi encontrada.

4. Para inspecionar ainda mais uma anomalia, escolha o botão de rádio em sua fileira.

   O painel Inspeção de padrões aparece e exibe o seguinte:

   - O padrão no qual essa anomalia se baseia. Selecione um token dentro do padrão para analisar os valores desse token.
   - Um histograma mostrando o número de ocorrências da anomalia no intervalo de tempo consultado.
   - A guia Amostras de registro exibe alguns dos eventos de registro que fazem parte da anomalia.
   - A guia Valores do Token exibe os valores do token dinâmico selecionado, se você tiver selecionado um.

   > **Note**
   >
   > Um máximo de 10 valores de token são capturados para cada token. A contagem de tokens pode não ser precisa. CloudWatch O Logs usa um contador probabilístico para gerar a contagem de tokens, não o valor absoluto.

5. Para suprimir uma anomalia, escolha o botão de rádio em sua linha e faça o seguinte:

   a. Escolha Ações, Suprimir anomalia.

   b. Em seguida, especifique por quanto tempo você deseja que a anomalia seja suprimida.

   c. Para suprimir todas as anomalias relacionadas a esse padrão, selecione Suprimir padrão.

   d. Escolha Suprimir anomalia.

Crie alarmes em detectores de anomalias de log

Você pode criar um alarme para um detector de anomalias de log em um grupo de log. Você pode especificar que o alarme entre em `ALARM` estado quando um número específico de anomalias for encontrado no grupo de registros durante um período de tempo especificado. Você também pode usar filtros para que somente anomalias de prioridades especificadas sejam contabilizadas pelo alarme.

Para criar um alarme para um detector de anomalias de log

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

2. No painel de navegação, escolha Logs, Log Anomalias.

   A tabela de detectores de anomalias de log é exibida.

3. Escolha o botão de rádio do detector de anomalias para o qual você deseja configurar o alarme e escolha Criar alarme.

   O assistente CloudWatch de criação de alarmes é exibido. O LogAnomalyDetectorcampo exibe o nome do detector de anomalias que você escolheu. O campo Nome da métrica é exibido AnomalyCount.

4. (Opcional) Para filtrar esse alarme pela prioridade de anomalia, faça o seguinte:

   - Para que o alarme conte somente anomalias de alta prioridade, digite para. **`HIGH`** LogAnomalyPriority
   - Para que o alarme conte somente anomalias de alta e média prioridade, digite para. **`MEDIUM`** LogAnomalyPriority

   Para obter mais informações sobre os níveis de prioridade, consulteGravidade e prioridade de anomalias e padrões.

5. Escolha usar um limite estático ou métrico de detecção de anomalias para o alarme. Essa seleção determina como o limite do alarme é definido. Um limite estático significa que o limite de alarme é um número estático e constante que você escolhe. Um limite de detecção de anomalias significa que CloudWatch determina uma faixa de valores usuais, e o alarme é acionado se a contagem real ultrapassar o limite dessa banda. Você não precisa escolher a detecção de anomalias para um alarme de detecção de anomalias de log. Para obter mais informações sobre a detecção métrica de anomalias, consulte Usando a detecção de CloudWatch anomalias.

6. Para Whenever ***`your-metric-name`***is... , escolha Maior, Maior/Igual, Menor/Igual ou Menor. Em seguida, para então …, especifique um número para o valor limite. O alarme entrará em **`ALARM`** estado se o detector de anomalias encontrar mais do que esse número de alarmes durante um tempo especificado por Período.

7. Escolha Additional configuration (Configuração adicional). Em Datapoints to alarm (Pontos de dados para alarme), especifique quantos períodos de avaliação (pontos de dados) devem estar no estado `ALARM` para disparar o alarme. Se os dois valores forem correspondentes, você criará um alarme que passa para o estado `ALARM` se esses períodos consecutivos estiverem violando.

   Para criar um alarme M de um alarme N, especifique um número para o primeiro valor que seja menor do que o segundo valor. Para obter mais informações, consulte Avaliação de um alarme.

8. Em Missing data treatment (Tratamento de dados ausentes), escolha como deseja que o alarme se comporte quando alguns pontos de dados estiverem ausentes. Para obter mais informações, consulte Configurando como os CloudWatch alarmes tratam os dados perdidos.
9. Escolha Próximo.
10. Em Notificação, escolha Adicionar notificação e, em seguida, especifique um tópico do Amazon SNS para notificar quando seu alarme passar para o estado `ALARM`, `OK`, ou. `INSUFFICIENT_DATA`

    a. (Opcional) Para enviar várias notificações para o mesmo estado de alarme ou para diferentes estados de alarme, escolha Add notification (Adicionar notificação).

    > **Note**
    >
    > Recomendamos configurar o alarme para executar ações quando entrar no estado Dados insuficientes, além de para quando entrar no estado Alarme. Isso ocorre porque muitos problemas com a função do Lambda que se conecta à fonte de dados podem fazer com que o alarme transite para Dados insuficientes.

    b. (Opcional) Para não enviar notificações do Amazon SNS, escolha Remover.
11. (Opcional) Se você quiser que seu alarme execute ações para Amazon EC2 Auto Scaling, Amazon EC2, tickets AWS Systems Manager ou, escolha o botão apropriado e especifique o estado e a ação do alarme.

    > **Note**
    >
    > O alarme pode executar ações do Systems Manager somente ao entrar no estado `ALARM`. Para obter informações sobre as ações do Systems Manager, consulte Configurando CloudWatch para criar OpsItems e Criação de incidentes.
12. Escolha Próximo.
13. Em Name and description (Nome e descrição), insira um nome e uma descrição para o alarme e selecione Next (Próximo). O nome deve conter somente caracteres UTF-8, e não poderá conter caracteres de controle ASCII. A descrição pode incluir a formatação markdown, que é exibida somente na guia Detalhes do alarme no CloudWatch console. O markdown pode ser útil para adicionar links para runbooks ou outros recursos internos.

> **Tip**
>
> O nome do alarme deve conter somente caracteres UTF-8. Ele não pode conter caracteres de controle ASCII.

14. Em Preview and create (Previsualizar e criar), confirme se as informações e condições do seu alarme estão corretas e escolha Create alarm (Criar alarme).

Métricas publicadas por detectores de anomalias de log

CloudWatch O Logs publica a AnomalyCountmétrica em CloudWatch métricas. Essa métrica é publicada no `AWS/Logs` namespace.

A AnomalyCountmétrica é publicada com as seguintes dimensões:

- LogAnomalyDetector— O nome do detector de anomalias
- LogAnomalyPriority— O nível de prioridade da anomalia

Criptografe um detector de anomalias e seus resultados com AWS KMS

Os dados do detector de anomalias são sempre criptografados nos CloudWatch registros. Por padrão, o CloudWatch Logs usa criptografia do lado do servidor para os dados em repouso. Como alternativa, você pode usar o AWS Key Management Service para essa criptografia. Se você fizer isso, a criptografia será feita usando uma AWS KMS chave. AWS KMS O uso da criptografia é ativado no nível do detector de anomalias, associando uma chave KMS a um detector de anomalias.

> **Important**
>
> CloudWatch O Logs suporta somente chaves KMS simétricas. Não use uma chave assimétrica para criptografar os dados em seus grupos de logs. Para obter mais informações, consulte Usar chaves simétricas e assimétricas.

Limites

- 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 chave de um detector de anomalias, pode levar até cinco minutos para que a operação entre em vigor.
- Se você revogar o acesso do CloudWatch Logs a uma chave associada ou excluir uma chave KMS associada, seus dados criptografados no CloudWatch Logs não poderão mais ser recuperados.

Etapa 1: criar uma AWS KMS chave

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

```
aws kms create-key
```

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

```
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "key-default-1",
        "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/key-default-1",
        "AWSAccountId": "123456789012",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

Etapa 2: definir permissões na chave do KMS

Por padrão, todas AWS KMS as chaves são privadas. Somente o proprietário do recurso pode usá-la para criptografar e descriptografar dados. No entanto, o proprietário do recurso pode conceder permissões para acessar a chave do KMS a outros usuários e recursos. Com essa etapa, você concede permissão principal ao serviço de CloudWatch registros para usar a chave. Esse principal de serviço deve estar na mesma AWS região em que a chave KMS está armazenada.

Como prática recomendada, recomendamos que você restrinja o uso da chave KMS somente às AWS contas ou detectores de anomalias que você especificar.

Primeiro, salve a política padrão para sua chave KMS `policy.json` usando o seguinte get-key-policycomando:

```
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 `Condition` seções para aumentar a segurança da AWS KMS chave. Para ter mais informações, consulte AWS KMS chaves e contexto de criptografia.

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

```
{
  "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:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*",
      "Condition": {
        "ArnLike": {
          "kms:EncryptionContext:aws:logs:arn":
 "arn:aws:logs:REGION:Your_account_ID:anomaly-detector:*"
        }
      }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.REGION.amazonaws.com"
        },
        "Action": [
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:DescribeKey"
        ],
        "Resource": "*",
        "Condition": {
            "ArnLike": {
                "kms:EncryptionContext:aws-crypto-ec:aws:logs:arn":
 "arn:aws:logs:REGION:Your_account_ID:anomaly-detector:*"
            }
        }
    }
  ]
}
```

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

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

Etapa 3: associar uma chave KMS a um detector de anomalias

Você pode associar uma chave KMS a um detector de anomalias ao criá-la no console ou usando as APIs AWS CLI ou.

Etapa 4: Desassociar a chave de um detector de anomalias

Depois que uma chave é associada a um detector de anomalias, você não pode atualizar a chave. A única maneira de remover a chave é excluir o detector de anomalias e, em seguida, recriá-lo.

Trabalhar com grupos de logs e fluxos de logs

Uma transmissão de log é uma sequência de eventos de log que compartilham a mesma fonte. Cada fonte separada de registros no CloudWatch Logs compõe um fluxo de registros 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.

Crie um grupo de registros em CloudWatch Registros

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

Para criar um grupo de logs

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

> **Tip**
>
> Você pode dar preferência a grupos de log, bem como a painéis e alarmes, no menu Favorites and recents (Favoritos e recentes) no painel de navegação. Na coluna Recently visited (Visitados recentemente), passe o mouse sobre o grupo de logs ao qual deseja dar preferência e escolha o símbolo da estrela ao lado dele.

Enviar logs a um grupo de logs

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

- CloudWatch agente — O CloudWatch agente unificado pode enviar métricas e registros para o CloudWatch Logs. Para obter informações sobre como instalar e usar o CloudWatch agente, consulte Coletando métricas e registros de instâncias do Amazon EC2 e servidores locais com o CloudWatch agente no Guia do usuário da Amazon CloudWatch .
- AWS CLI—O put-log-eventsupload de lotes de eventos de log para CloudWatch o Logs.
- Programaticamente — A PutLogEventsAPI permite que você faça upload programático de lotes de eventos de log para o Logs. CloudWatch

Exibir dados de registro enviados para o CloudWatch Logs

Você pode visualizar e percorrer os dados de registro de stream-by-stream acordo com os enviados ao 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 CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha 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 de eventos de log usando filtros.
   - 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.

Usar o Live Tail para visualizar logs quase em tempo real

CloudWatch O Logs Live Tail ajuda você a solucionar incidentes rapidamente, visualizando uma lista de streaming de novos eventos de log à medida que são ingeridos. É possível visualizar, filtrar e realçar logs consumidos quase em tempo real, o que ajuda a detectar e resolver problemas rapidamente. Os logs podem ser filtrados com base em termos especificados, e você também pode realçar logs que contêm termos específicos para ajudar a encontrar rapidamente o que está procurando.

As sessões Live Tail acumulam custos por tempo de uso da sessão, por minuto. Para obter mais informações sobre preços, consulte a guia Logs em Amazon CloudWatch Pricing.

> **Note**
>
> O Live Tail é suportado somente para grupos de registros na classe de registros Standard. Para obter mais informações sobre classes de log, consulteClasses de log.

As seções a seguir explicam como usar o Live Tail no console. Você também pode iniciar uma sessão do Live Tail programaticamente. Para obter mais informações, consulte StartLiveTail. Para exemplos de SDK, consulte Iniciar uma sessão do Live Tail usando um AWS SDK.

Iniciar uma sessão Live Tail

Você usa o CloudWatch console para iniciar uma sessão do Live Tail. O procedimento a seguir explica como iniciar uma sessão Live Tail usando a opção Live Tail no painel de navegação esquerdo. Você também pode iniciar sessões do Live Tail na página Grupos de registros ou na página CloudWatch Logs Insights.

> **Note**
>
> Se você estiver usando políticas de proteção de dados para mascarar dados confidenciais em um grupo de logs que está visualizando com o Live Tail, esses dados sempre aparecerão

mascarados na sessão Live Tail. Para obter mais informações sobre como mascarar dados confidenciais em grupos de logs, consulte Ajude a proteger dados de log confidenciais com mascaramento.

Para iniciar uma sessão Live Tail

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Logs, Live tail.
3. Em Selecionar grupos de logs, selecione os grupos de logs cujos eventos você deseja visualizar na sessão Live Tail. Você pode selecionar até 10 grupos de logs.
4. (Opcional) Se você tiver selecionado somente um grupo de logs, poderá filtrar ainda mais sua sessão Live Tail selecionando um ou mais fluxos de logs dos quais visualizar eventos de log. Para fazer isso, em Selecionar fluxos de logs, selecione os nomes dos fluxos de logs na lista suspensa. Como alternativa, você pode usar a segunda caixa em Selecionar fluxos de logs para inserir um prefixo de nome de fluxo de logs. Todos os fluxos de log com nomes que corresponderem ao prefixo serão selecionados.
5. (Opcional) Para exibir somente eventos de log que contenham certas palavras ou outras strings, insira a palavra ou string em `Add filter patterns`.

   Por exemplo, para exibir somente eventos de log que incluam a palavra `Warning`, insira **`Warning`**. Os filtros diferenciam maiúsculas de minúsculas. É possível incluir vários termos e operadores de padrão nesse campo:

   - **`error 404`** exibe somente eventos de log que incluem tanto `error` quanto `404`
   - **`?Error ?error`** exibe eventos de log que incluem `Error` ou `error`
   - **`-INFO`** exibe todos os eventos de log que não incluem `INFO`
   - **`{ $.eventType = "UpdateTrail" }`** exibe todos os eventos de log JSON em que o valor do campo de tipo de evento é `UpdateTrail`

   Você também pode usar a expressão regular (regex) para filtrar:

   - **`%ERROR%`** usa regex para exibir todos os eventos de log que consistem na palavra-chave ERRO
   - **`{ $.names = %Steve% }`** usa regex para exibir eventos de log JSON em que Steve está na propriedade `"name"`

- **`[ w1 = %abc%, w2 ]`** usa regex para exibir eventos de log delimitados por espaço em que a primeira palavra é abc

Para obter mais informações sobre a sintaxe do padrão, consulte Sintaxe de padrões de filtros.

6. (Opcional) Para realçar alguns dos eventos de log exibidos, insira um termo para procurar e realçar em Live Tail. Insira um por vez os termos de realce. Se você adicionar vários termos para realçar, uma cor diferente será atribuída para representar cada um. Um indicador de realce é exibido à esquerda de qualquer evento de log que contenha o termo especificado e também abaixo do próprio termo quando você expande o evento de log na janela principal para visualizar o evento de log completo.

   Você pode usar a filtragem e o realce para solucionar problemas rapidamente. Por exemplo, você pode filtrar os eventos para exibir somente aqueles que contêm `Error` e, em seguida, realçar os eventos que contêm `404`.

7. Para iniciar a sessão, escolha Aplicar filtros.

   Os eventos de log correspondentes começam a aparecer na janela. Também são exibidas as seguintes informações:

   - O timer mostra por quanto tempo a sessão Live Tail está ativa.
   - eventos/s mostra quantos eventos de log consumidos por segundo correspondem aos filtros que você definiu.
   - Para evitar que a sessão seja rolada muito rápido porque muitos eventos correspondem aos filtros, o CloudWatch Logs pode exibir somente alguns eventos correspondentes. Se isso acontecer, a porcentagem de eventos correspondentes exibidos na tela é mostrada em % de exibidos.

8. Para pausar o fluxo de eventos a fim de investigar o que está sendo exibido atualmente, clique em qualquer parte da janela de eventos.

9. Durante a sessão, é possível usar o seguinte para ver mais detalhes sobre cada evento de log.

   - Para exibir o texto inteiro de um evento de log na janela principal, escolha a seta ao lado desse evento de log.
   - Para exibir o texto inteiro de um evento de log em uma janela lateral, escolha a lupa + ao lado desse evento de log. O fluxo de eventos é interrompido, e a janela lateral é exibida.

A exibição do texto de um evento de log na janela lateral pode ser útil para comparar seu texto com outros eventos na janela principal.

10. Para interromper a sessão Live Tail, escolha Parar.
11. Para reiniciar a sessão, uma opção é usar o painel Filtro para modificar os critérios de filtragem e escolher Aplicar filtros. Depois, selecione Start (Iniciar).

Pesquisar dados de log usando padrões de filtro

Você pode pesquisar seus dados de log usando Sintaxe de padrões de filtros para filtros de métricas, filtros de assinatura, filtros de eventos de log e Live Tail.. Você pode pesquisar todos os fluxos de log em um grupo de registros ou, usando o, AWS CLI você também pode pesquisar fluxos de log específicos. 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.

Se você estiver conectado a uma conta configurada como uma conta de monitoramento na observabilidade CloudWatch entre contas, poderá pesquisar e filtrar eventos de log das contas de origem vinculadas a essa conta de monitoramento. Para obter mais informações, consulte CloudWatch observabilidade entre contas.

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 CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.

3. Em Grupos de logs, escolha o nome do grupo de logs que contém o fluxo de log a ser pesquisado.
4. Em Fluxos de log, escolha o nome do fluxo de log para pesquisa.
5. Em Eventos de log, insira a sintaxe do filtro a ser usada.

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

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. Em Grupos de logs, escolha o nome do grupo de logs que contém o fluxo de log a ser pesquisado.
4. Escolha Pesquisar grupo de logs.
5. Para Eventos de log, selecione o intervalo de data e hora e insira a sintaxe do filtro.

Pesquisar entradas de registro usando o AWS CLI

Você pode pesquisar entradas de registro que atendam a um critério específico usando AWS CLI o.

Para pesquisar entradas de registro usando o AWS CLI

Em um prompt de comando, execute o filter-log-eventscomando a seguir. Use `--filter-pattern` para limitar os resultados para o padrão de filtro especificado e `--log-stream-names` para limitar os resultados para os fluxos 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 registro em um determinado intervalo de tempo usando o AWS CLI

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

```
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 CloudWatch console 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 CloudWatch console 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.

Solução de problemas

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 estiver usando o AWS CLI, você pode limitar a pesquisa apenas aos fluxos de log nos quais está interessado. Por exemplo, se seu grupo de registros tiver 1.000 fluxos de registros, mas você quiser ver apenas três fluxos de registros que você sabe que são relevantes, você pode usar o AWS CLI para limitar sua pesquisa somente aos três fluxos de registros dentro do grupo de registros.
- 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 de dados de CloudWatch registro em registros

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

> **Note**
>
> CloudWatch Logs não exclui imediatamente os eventos de registro quando eles atingem a configuração de retenção. Normalmente, demora até 72 horas depois disso para que os eventos de log sejam excluídos, mas em raras situações pode demorar mais.
> Isso significa que, se você alterar um grupo de logs para ter uma configuração de retenção mais longa quando ele contiver eventos de log que já passaram da data de expiração, mas que não foram realmente excluídos, esses eventos de log levarão até 72 horas para serem excluídos depois que a nova data de retenção for atingida. Para garantir que os dados de log sejam excluídos permanentemente, mantenha um grupo de logs em sua configuração de retenção inferior a 72 horas após o término do período de retenção anterior ou você ter confirmado que os eventos de log mais antigos foram excluídos.
> Quando os eventos de log atingem sua configuração de retenção, eles são marcados para exclusão. Depois de marcados para exclusão, eles não aumentam mais seus custos de armazenamento de arquivamento, mesmo que não sejam realmente excluídos no momento. Esses eventos de logs marcados para exclusão também não são incluídos quando você usa uma API para recuperar o valor `storedBytes` para ver quantos bytes um grupo de logs está armazenando.

Para alterar a configuração de retenção de logs

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Logs, Log groups (Grupos de log).
3. Localize o grupo de logs a ser atualizado.
4. Na coluna Retenção desse grupo de registros, escolha a configuração de retenção atual, como Nunca expirar.
5. Em Configuração de retenção, em Expirar eventos depois, escolha um valor de retenção de log e escolha Salvar.

Marque grupos de registros no Amazon CloudWatch Logs

Você pode atribuir seus próprios metadados aos grupos de registros que você cria no Amazon CloudWatch Logs na forma de tags. Marca é um par de chave-valor que você define para um grupo de logs. Usar tags é uma maneira simples, porém poderosa, de gerenciar AWS recursos e organizar dados, incluindo dados de faturamento.

> **Note**
>
> Você pode usar tags para controlar o acesso aos recursos do CloudWatch Logs, incluindo grupos e destinos de registros. O acesso aos fluxos de log é controlado no nível do grupo de log, devido à relação hierárquica entre grupos de log e fluxos de log. Para obter mais informações sobre como usar etiquetas para controlar o acesso, consulte Controlar acesso a recursos da Amazon Web Services usando etiquetas de recursos.

Conteúdo

- Conceitos Básicos de Tags
- Monitorar custos usando a marcação
- Restrições de tags
- Marcando grupos de registros usando o AWS CLI
- Como marcar grupos de registros usando a API CloudWatch Logs

Conceitos Básicos de Tags

Você usa AWS CloudFormation a API AWS CLI, ou 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 aplicação associada. Aqui estão alguns exemplos de marcas:

- Projeto: nome do projeto
- Proprietário: nome
- Objetivo: testes de carga
- Aplicação: nome da aplicação
- Ambiente: produção

Monitorar custos usando a marcação

Você pode usar tags para categorizar e monitorar seus AWS custos. Quando você aplica tags aos seus AWS recursos, incluindo grupos de registros, seu relatório de alocação de AWS custos inclui o uso e os custos agregados por tags. É possível aplicar tags que representem categorias de negócios (como centros de custos, nomes de aplicações ou proprietários) para organizar seus custos de vários serviços. Para obter mais informações, consulte Usar etiquetas de alocação de custos para relatórios de faturamento personalizados no Manual do usuário do AWS Billing .

Restrições de tags

As restrições a seguir se aplicam a marcas.

Restrições básicas

- O número máximo de marcas por grupo de logs é 50.
- As chaves e valores das 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 tag com `aws:` porque esse prefixo é reservado para uso por AWS. AWS cria tags 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: `_ . / = + - @`.

Marcando grupos de registros usando o 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-resource

Atribui uma ou mais tags (pares de valores-chave) ao recurso de registros especificado CloudWatch .

list-tags-for-resource

Exibe as tags associadas a um recurso de CloudWatch registros.

untag-resource

Remove uma ou mais tags do recurso de CloudWatch registros especificado.

Como marcar grupos de registros usando a API CloudWatch Logs

Você pode adicionar, listar e remover tags usando a API 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.

TagResource

Atribui uma ou mais tags (pares de valores-chave) ao recurso de registros especificado CloudWatch .

ListTagsForResource

Exibe as tags associadas a um recurso de CloudWatch registros.

UntagResource

Remove uma ou mais tags do recurso de CloudWatch registros especificado.

Criptografe dados de registro no CloudWatch Logs usando AWS Key Management Service

Os dados do grupo de registros são sempre criptografados nos CloudWatch registros. Por padrão, o CloudWatch Logs usa criptografia do lado do servidor para os dados de registro em repouso. Como alternativa, você pode usar o AWS Key Management Service para essa criptografia. Se você fizer isso, a criptografia será feita usando uma AWS KMS chave. O uso da criptografia AWS KMS é habilitado no nível do grupo de registros, associando uma chave KMS a um grupo de registros, seja quando você cria o grupo de registros ou depois que ele existe.

Important

CloudWatch Os registros agora oferecem suporte ao contexto de criptografia, usando `kms:EncryptionContext:aws:logs:arn` como chave e o ARN do grupo de registros como o valor dessa chave. Se você tiver grupos de logs que já criptografou com uma chave do KMS e quiser restringir a chave de modo que ela seja usada com uma única conta e grupo de logs, atribua uma nova chave do KMS que contenha uma condição na política do IAM. Para ter mais informações, consulte AWS KMS chaves e contexto de criptografia.

Depois que você associar uma chave do KMS a um grupo de logs, todos os novos dados ingeridos para o grupo de logs serão criptografados usando essa chave. Esses dados são armazenados em formato criptografado durante todo o período de retenção. CloudWatch O Logs descriptografa esses dados sempre que solicitados. CloudWatch Os registros devem ter permissões para a chave KMS sempre que dados criptografados forem solicitados.

Se, posteriormente, você desassociar uma chave KMS de um grupo de CloudWatch registros, o Logs criptografará os dados recém-ingeridos usando o método de criptografia padrão do CloudWatch Logs. Todos os dados ingeridos anteriormente que foram criptografados com a chave KMS permanecem criptografados com a chave KMS. CloudWatch Os registros ainda podem retornar esses dados após a desassociação da chave KMS, porque CloudWatch os registros ainda podem continuar referenciando a chave. No entanto, se a chave for desativada posteriormente, o CloudWatch Logs não conseguirá ler os registros que foram criptografados com essa chave.

> **Important**
>
> CloudWatch O Logs suporta somente chaves KMS simétricas. Não use uma chave assimétrica para criptografar os dados em seus grupos de logs. Para obter mais informações, consulte Usar chaves simétricas e assimétricas.

Limites

- 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 chave de um grupo de logs, pode levar até cinco minutos para que a operação seja efetivada.
- Se você revogar o acesso do CloudWatch Logs a uma chave associada ou excluir uma chave KMS associada, seus dados criptografados no CloudWatch Logs não poderão mais ser recuperados.
- Você não pode associar uma chave KMS a um grupo de registros usando o CloudWatch console.

Etapa 1: criar uma AWS KMS chave

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

```
aws kms create-key
```

A saída contém o ID de chave e o nome do recurso da Amazon (ARN) da chave. 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 chave do KMS

Por padrão, todas AWS KMS as chaves são privadas. Somente o proprietário do recurso pode usá-la para criptografar e descriptografar dados. No entanto, o proprietário do recurso pode conceder permissões para acessar a chave do KMS a outros usuários e recursos. Com essa etapa, você concede permissão principal ao serviço de CloudWatch registros para usar a chave. Esse principal de serviço deve estar na mesma AWS região em que a chave KMS está armazenada.

Como prática recomendada, recomendamos que você restrinja o uso da chave KMS somente às AWS contas ou grupos de registros que você especificar.

Primeiro, salve a política padrão para sua chave KMS `policy.json` usando o seguinte get-key-policycomando:

```
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 `Condition` seções para aumentar a segurança da AWS KMS chave. Para obter mais informações, consulte AWS KMS chaves e contexto de criptografia.

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` deste exemplo limita o uso da chave do AWS KMS à conta especificada, mas ele pode ser usado 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 put-key-policycomando:

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

Etapa 3: associar uma chave do KMS a um grupo de logs

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

Para descobrir se um grupo de registros já tem uma chave KMS associada, use o seguinte describe-log-groupscomando:

```
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 chave do KMS 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 chave do KMS 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 uma chave de um grupo de logs

Para desassociar a chave KMS associada a um grupo de registros, use o seguinte comando: disassociate-kms-key

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

AWS KMS chaves e contexto de criptografia

Para aumentar a segurança de suas AWS Key Management Service chaves e de seus grupos de registros criptografados, o CloudWatch Logs agora coloca os ARNs do grupo de registros como parte do contexto de criptografia usado para criptografar seus dados de registro. 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 as condições da política do IAM para limitar o acesso à sua AWS KMS chave por AWS conta e grupo de registros. Para obter mais informações, consulte Contexto de criptografia e Elementos de política JSON do IAM: condição.

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

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

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

1. Insira o comando a seguir para localizar o ARN da chave 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 chave do KMS:

   ```
   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. Insira o comando a seguir para adicionar a política atualizada à nova chave do KMS:

```
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 Agora, o Logs criptografa todos os novos dados usando a nova chave.

7. Depois, revogue todas as permissões, exceto `Decrypt` da chave 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::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt*"
            ],
            "Resource": "*"
        }
    ]
}
```

9. Insira o comando a seguir para adicionar a política atualizada à antiga chave:

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

Ajude a proteger dados de log confidenciais com mascaramento

Você pode ajudar a proteger dados confidenciais que são ingeridos pelo CloudWatch Logs usando políticas de proteção de dados de grupos de registros. Essas políticas permitem auditar e mascarar dados confidenciais que aparecem nos eventos de log consumidos pelos grupos de logs na sua conta.

Quando você cria uma política de proteção de dados, por padrão, os dados confidenciais que correspondem aos identificadores de dados selecionados são mascarados em todos os pontos de saída, incluindo CloudWatch Logs Insights, filtros métricos e filtros de assinatura. Somente usuários com a permissão do IAM `logs:Unmask` podem visualizar dados não mascarados.

Você pode criar uma política de proteção de dados para todos os grupos de logs da sua conta, além de criar políticas de proteção de dados para grupos de logs individuais. Ao criar uma política para toda a conta, ela se aplica tanto aos grupos de logs existentes quanto aos grupos de logs que forem criados no futuro.

Se você criar uma política de proteção de dados para toda a conta e também criar uma política para um único grupo de logs, as duas políticas serão aplicáveis a esse grupo de logs. Todos os identificadores de dados gerenciados especificados em qualquer política são auditados e mascarados nesse grupo de logs.

> **Note**
>
> O mascaramento de dados confidenciais é suportado somente para grupos de registros na classe de registros Standard. Se você criar uma política de proteção de dados para todos os grupos de registros em sua conta, ela se aplicará somente aos grupos de registros na classe de registro Padrão. Para obter mais informações sobre classes de log, consulteClasses de log.

Cada grupo de logs pode ter somente uma política de proteção de dados em nível de grupo de logs, mas essa política pode especificar vários identificadores de dados gerenciados para auditoria e mascaramento. O limite de uma política de proteção de dados é de 30.720 caracteres.

> **Important**
>
> Os dados confidenciais são detectados e mascarados quando são ingeridos no grupo de logs. Quando você define uma política de proteção de dados, os eventos de log ingeridos no grupo de logs antes dessa hora não são mascarados.

CloudWatch O Logs oferece suporte a vários identificadores de dados gerenciados, que oferecem tipos de dados pré-configurados que você pode selecionar para proteger dados financeiros, informações pessoais de saúde (PHI) e informações de identificação pessoal (PII). CloudWatch A proteção de dados de registros permite que você aproveite a correspondência de padrões e os modelos de aprendizado de máquina para detectar dados confidenciais. Para alguns tipos de identificadores de dados gerenciados, a detecção depende também da localização de determinadas palavras-chave próximas aos dados confidenciais. Você também pode usar identificadores de dados personalizados para criar identificadores de dados personalizados para seu caso de uso específico.

CloudWatch Quando dados confidenciais são detectados, é emitida uma métrica que corresponde aos identificadores de dados selecionados. Essa é a LogEventsWithFindingsmétrica e é emitida no namespace AWS/Logs. Você pode usar essa métrica para criar CloudWatch alarmes e visualizá-la em gráficos e painéis. As métricas emitidas pela proteção de dados são métricas vendidas e gratuitas. Para obter mais informações sobre as métricas para as CloudWatch quais o Logs envia CloudWatch, consulteMonitoramento com CloudWatch métricas.

Cada identificador de dados gerenciados foi projetado para detectar um tipo específico de dados confidenciais, como números de cartão de crédito, chaves de acesso AWS secretas ou números de passaportes de um determinado país ou região. Ao criar uma política de proteção de dados, você pode configurá-la para usar esses identificadores para analisar logs ingeridos pelo grupo de logs e executar ações quando forem detectados.

CloudWatch A proteção de dados de registros pode detectar as seguintes categorias de dados confidenciais usando identificadores de dados gerenciados:

- Credenciais, como chaves privadas ou chaves de acesso AWS secretas
- Informações financeiras, como números de cartão de crédito
- Informações de identificação pessoal (PII), como carteiras de motorista ou números de previdência social
- Informações de saúde protegidas (PHI), como seguro de saúde ou números de identificação médica
- Identificadores de dispositivos, como endereços IP ou endereços MAC

Para obter detalhes sobre os tipos de dados que você pode proteger, consulte Tipos de dados que você pode proteger.

Sumário

- Noções básicas sobre políticas de proteção de dados
  - O que são políticas de proteção de dados?
  - Como a política de proteção de dados é estruturada?
    - Propriedades JSON para a política de proteção de dados
    - Propriedades JSON para uma declaração de política
    - Propriedades JSON para uma operação de declaração de política
- Permissões de IAM necessárias para criar ou trabalhar com uma política de proteção de dados

- Permissões necessárias para políticas de proteção de dados no nível da conta
- Permissões necessárias para políticas de proteção de dados para um único grupo de logs
- Amostra da política de proteção de dados
- Criar uma política de proteção de dados para toda a conta
  - Console
  - AWS CLI
    - Sintaxe da política de proteção de dados para AWS CLI nossas operações de API
- Criar uma política de proteção de dados para um único grupo de logs
  - Console
  - AWS CLI
    - Sintaxe da política de proteção de dados para AWS CLI nossas operações de API
- Exibir dados não mascarados
- Relatórios de descobertas de auditoria
  - Política-chave necessária para enviar os resultados da auditoria para um bucket protegido por AWS KMS
- Tipos de dados que você pode proteger
  - CloudWatch Registra identificadores de dados gerenciados para tipos de dados confidenciais
    - Credenciais
      - ARNs de identificadores de dados para tipos de dados de credencial
    - Identificadores de dispositivo
      - ARNs de identificadores de dados para tipos de dados de dispositivos
    - Informações financeiras
      - ARNs de identificadores de dados para tipos de dados financeiros
    - Informações de saúde protegidas (PHI)
      - ARNs identificadores de dados para tipos de dados de informações de saúde protegidas (PHI)
    - Informações de identificação pessoal (PII)
      - Palavras-chave para números de identificação da carteira de habilitação
      - Palavras-chave para números de identificação nacional
      - Palavras-chave para números de passaporte
      - Palavras-chave para identificação do contribuinte e números de referência

    - ARNs do identificador de dados de informações de identificação pessoal (PII)
- Identificadores de dados personalizados
    - O que são identificadores de dados personalizados?
    - Restrições de identificadores de dados personalizados
    - Usando identificadores de dados personalizados no console
    - Usar identificadores de dados personalizados na política de proteção de dados

Noções básicas sobre políticas de proteção de dados

Tópicos

- O que são políticas de proteção de dados?
- Como a política de proteção de dados é estruturada?

O que são políticas de proteção de dados?

CloudWatch O Logs usa políticas de proteção de dados para selecionar os dados confidenciais que você deseja verificar e as ações que você deseja tomar para proteger esses dados. Para selecionar os dados confidenciais de interesse, você usa identificadores de dados. CloudWatch A proteção de dados de registros e, em seguida, detecta os dados confidenciais usando aprendizado de máquina e correspondência de padrões. Para agir sobre identificadores de dados encontrados, você pode definir operações de auditoria e desidentificação. Essas operações permitem registrar os dados confidenciais encontrados (ou não encontrados) e mascarar os dados sensíveis quando os eventos de log são exibidos.

Como a política de proteção de dados é estruturada?

Como mostrado na figura abaixo, um documento de política de proteção de dados inclui os seguintes elementos:

- Informações opcionais da política na parte superior do documento
- Uma declaração que define as ações de auditoria e desidentificação

Somente uma política de proteção de dados pode ser definida por grupo de CloudWatch registros de registros. A política de proteção de dados pode ter uma ou mais instruções de negação ou desidentificação, mas somente uma instrução de auditoria.

Propriedades JSON para a política de proteção de dados

Uma política de proteção de dados requer as seguintes informações básicas de política para identificação:

- Nome: o nome da política.
- Descrição (Opcional): a descrição da política.
- Versão: a versão do idioma das políticas. A versão atual é 2021-06-01.
- Declaração: uma lista de declarações que especificam as ações da política de proteção de dados.

```
{
  "Name": "CloudWatchLogs-PersonalInformation-Protection",
  "Description": "Protect basic types of sensitive data",
  "Version": "2021-06-01",
  "Statement": [
        ...
  ]
}
```

Propriedades JSON para uma declaração de política

Uma declaração de política define o contexto de detecção para a operação de proteção de dados.

- Sid (Opcional): o identificador da declaração.
- DataIdentifier— Os dados confidenciais que os CloudWatch registros devem verificar. Por exemplo, nome, endereço ou número de telefone.
- Operação — As ações subsequentes, seja Auditoria ou Desidentificação. CloudWatch O Logs executa essas ações quando encontra dados confidenciais.

```
{
  ...
  "Statement": [
    {
      "Sid": "audit-policy",
      "DataIdentifier": [
        "arn:aws:dataprotection::aws:data-identifier/Address"
      ],
      "Operation": {
```

```
        "Audit": {
          "FindingsDestination": {}
        }
      }
    },
```

Propriedades JSON para uma operação de declaração de política

Uma declaração de política define uma das operações de proteção de dados a seguir.

- Auditoria — emite relatórios de métricas e descobertas sem interromper os logs. As sequências de caracteres correspondentes incrementam a LogEventsWithFindingsmétrica que o CloudWatch Logs publica no namespace AWS/Logs. CloudWatch É possível usar essas métricas para criar alarmes.

  Para obter um exemplo de um relatório de descobertas, consulte Relatórios de descobertas de auditoria.

  Para obter mais informações sobre as métricas para as CloudWatch quais o Logs envia CloudWatch, consulteMonitoramento com CloudWatch métricas.
- Desidentificar — mascare os dados confidenciais sem interromper os logs.

Permissões de IAM necessárias para criar ou trabalhar com uma política de proteção de dados

Para poder trabalhar com políticas de proteção de dados para grupos de logs, você deve ter certas permissões, conforme mostrado nas tabelas a seguir. As permissões são diferentes para políticas de proteção de dados em toda a conta e para políticas de proteção de dados aplicáveis a um único grupo de logs.

Permissões necessárias para políticas de proteção de dados no nível da conta

> **Note**
>
> Se você estiver executando qualquer uma dessas operações dentro de uma função do Lambda, a função de execução do Lambda e o limite de permissões também devem incluir as seguintes permissões.

Operation	Permissão necessárias do IAM	Recurso
Criar uma política de proteção de dados sem destinos de auditoria	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
Crie uma política de proteção de dados com o CloudWatch Logs como destino de auditoria	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`logs:PutResourcePolicy`	*
	`logs:DescribeResourcePolicies`	*
	`logs:DescribeLogGroups`	*
Crie uma política de proteção de dados com o Firehose como destino de auditoria	`logs:PutAccountPolicy`	*
	`logs:PutDataProtectionPolicy`	*
	`logs:CreateLogDelivery`	*
	`firehose:TagDeliveryStream`	`arn:aws:logs:::deliverystre`

Operation	Permissão necessárias do IAM	Recurso
		am/ *YOUR_DELIVERY_STREAM*
Criar uma política de proteção de dados com o Amazon S3 como destino de auditoria	logs:PutAccountPolicy	*
	logs:PutDataProtectionPolicy	*
	logs:CreateLogDelivery	*
	s3:GetBucketPolicy	arn:aws:s3::: *YOUR_BUCKET*
	s3:PutBucketPolicy	arn:aws:s3::: *YOUR_BUCKET*
Desmascarar eventos de log mascarados em um grupo de logs especificado	logs:Unmask	arn:aws:logs:::log-group:*
Ver uma política de proteção de dados existente	logs:GetDataProtectionPolicy	*
Excluir uma política de proteção de dados	logs:DeleteAccountPolicy	*
	logs:DeleteDataProtectionPolicy	*

Se algum log de auditoria de proteção de dados já estiver sendo enviado para um destino, outras políticas que enviam logs para o mesmo destino precisarão apenas das permissões `logs:PutDataProtectionPolicy` e `logs:CreateLogDelivery`.

Permissões necessárias para políticas de proteção de dados para um único grupo de logs

> **Note**
>
> Se você estiver executando qualquer uma dessas operações dentro de uma função do Lambda, a função de execução do Lambda e o limite de permissões também devem incluir as seguintes permissões.

Operation	Permissão necessárias do IAM	Recurso
Criar uma política de proteção de dados sem destinos de auditoria	`logs:PutDataProtectionPolicy`	`arn:aws:logs:::log-group:` *`YOUR_LOG_GROUP`* `:*`

Crie uma política de proteção de dados com o CloudWatch Logs como destino de auditoria	`logs:PutDataProtectionPolicy`
	`logs:CreateLogDelivery`
	`logs:PutResourcePolicy`
	`logs:DescribeResourcePolicies`
	`logs:DescribeLogGroups`
		`arn:aws:logs:::log-group:` *`YOUR_LOG_GROUP`* `:*`
		`*`
		`*`
		`*`
		`*`

Crie uma política de proteção de dados com o Firehose como destino de auditoria	`logs:PutDataProtectionPolicy` `logs:CreateLogDelivery`	`arn:aws:logs:::log-group:` *`YOUR_LOG_GROUP`* `:*` `*`

Operation	Permissão necessárias do IAM	Recurso
	`firehose:TagDeliveryStream`	`arn:aws:logs:::deliverystream/` *YOUR_DELIVERY_STREAM*

Criar uma política de proteção de dados com o Amazon S3 como destino de auditoria	`logs:PutDataProtectionPolicy`
	
	`logs:CreateLogDelivery`
	
	`s3:GetBucketPolicy`
	
	`s3:PutBucketPolicy`
		`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
		
		`*`
		
		`arn:aws:s3:::` *YOUR_BUCKET*
		
		`arn:aws:s3:::` *YOUR_BUCKET*

Desmascarar eventos de logs	`logs:Unmask`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
Ver uma política de proteção de dados existente	`logs:GetDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`
Excluir uma política de proteção de dados	`logs:DeleteDataProtectionPolicy`	`arn:aws:logs:::log-group:` *YOUR_LOG_GROUP* `:*`

Se algum log de auditoria de proteção de dados já estiver sendo enviado para um destino, outras políticas que enviam logs para o mesmo destino precisarão apenas das permissões `logs:PutDataProtectionPolicy` e `logs:CreateLogDelivery`.

Amostra da política de proteção de dados

O exemplo de política a seguir permite que um usuário crie, visualize e exclua políticas de proteção de dados que podem enviar descobertas de auditoria para todos os três tipos de destinos de auditoria. Ele não permite que o usuário visualize dados não mascarados.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "YOUR_SID_1",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:DescribeLogGroups",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Sid": "YOUR_SID_2",
            "Effect": "Allow",
            "Action": [
                "logs:GetDataProtectionPolicy",
                "logs:DeleteDataProtectionPolicy",
                "logs:PutDataProtectionPolicy",
                "s3:PutBucketPolicy",
                "firehose:TagDeliveryStream",
                "s3:GetBucketPolicy"
            ],
            "Resource": [
                "arn:aws:firehose:::deliverystream/YOUR_DELIVERY_STREAM",
                "arn:aws:s3:::YOUR_BUCKET",
                "arn:aws:logs:::log-group:YOUR_LOG_GROUP:*"
            ]
        }
    ]
}
```

Criar uma política de proteção de dados para toda a conta

Você pode usar o console de CloudWatch registros ou AWS CLI os comandos para criar uma política de proteção de dados para mascarar dados confidenciais de todos os grupos de registros em sua conta. Isso afeta os grupos de logs atuais e os grupos de logs que você criar no futuro.

> **Important**
>
> Os dados confidenciais são detectados e mascarados quando são ingeridos no grupo de logs. Quando você define uma política de proteção de dados, os eventos de log ingeridos no grupo de logs antes dessa hora não são mascarados.

Tópicos

- Console
- AWS CLI

Console

Para usar o console a fim de criar uma política de proteção de dados para toda a conta

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Configurações. Ele está localizado no final da lista.
3. Escolha a guia Logs.
4. Selecione Configurar.
5. Em Identificadores de dados gerenciados, selecione os tipos de dados que você deseja auditar e mascarar para todos os seus grupos de registros. Você pode digitar na caixa de seleção para encontrar os identificadores que deseja.

   Recomendamos que você selecione apenas os identificadores de dados relevantes para seus dados de log e sua empresa. A escolha de muitos tipos de dados pode levar a falsos positivos.

   Para obter detalhes sobre quais tipos de dados que você pode proteger, consulte Tipos de dados que você pode proteger.
6. (Opcional) Se você quiser auditar e mascarar outros tipos de dados usando identificadores de dados personalizados, escolha Adicionar identificador de dados personalizado. Em seguida,

insira um nome para o tipo de dados e a expressão regular a serem usados para pesquisar esse tipo de dados nos eventos de log. Para ter mais informações, consulte Identificadores de dados personalizados.

Uma única política de proteção de dados pode incluir até 10 identificadores de dados personalizados. Cada expressão regular que define um identificador de dados personalizado deve ter 200 caracteres ou menos.

7. (Opcional) Escolha um ou mais serviços para enviar as descobertas da auditoria. Mesmo se você optar por não enviar descobertas de auditoria a nenhum desses serviços, os tipos de dados confidenciais selecionados ainda serão mascarados.
8. Escolha Activate data protection (Ativar proteção de dados).

AWS CLI

Para usar o AWS CLI para criar uma política de proteção de dados

1. Use um editor de texto para criar um arquivo de política chamado `DataProtectionPolicy.json`. Para obter informações sobre a sintaxe da política, consulte a seção a seguir.
2. Digite o comando :

```
aws logs put-account-policy \
--policy-name TEST_POLICY --policy-type "DATA_PROTECTION_POLICY" \
--policy-document file://policy.json \
--scope "ALL" \
--region us-west-2
```

Sintaxe da política de proteção de dados para AWS CLI nossas operações de API

Quando você cria uma política de proteção de dados JSON para usar em um AWS CLI comando ou operação de API, a política deve incluir dois blocos JSON:

- O primeiro bloco deve incluir uma matriz `DataIdentifer` e uma propriedade `0peration` com uma ação `Audit`. A matriz `DataIdentifer` faz uma lista com os tipos de dados sigilosos que você deseja mascarar. Para obter mais informações sobre as opções disponíveis, consulte Tipos de dados que você pode proteger.

A propriedade `Operation` com uma ação `Audit` é necessária para encontrar os termos de dados confidenciais. Essa ação `Audit` deve conter um objeto `FindingsDestination`. Opcionalmente, você pode usar esse objeto `FindingsDestination` para listar um ou mais destinos para enviar relatórios de descobertas de auditoria. Se você especificar destinos como grupos de logs, streams do Amazon Data Firehose e buckets S3, eles já devem existir. Para obter um exemplo de um relatório de constatações de auditoria, consulte Relatórios de descobertas de auditoria.

- O segundo bloco deve incluir uma matriz `DataIdentifer` e uma propriedade `Operation` com uma ação `Deidentify`. A matriz `DataIdentifer` deve corresponder exatamente à matriz `DataIdentifer` no primeiro bloco da política.

  A propriedade `Operation` com a ação `Deidentify` é o que realmente mascara os dados e deve conter o objeto `"MaskConfig": {}`. O objeto `"MaskConfig": {}` deve estar vazio.

Veja a seguir um exemplo de uma política de proteção de dados usando somente identificadores de dados gerenciados. Essa política mascara endereços de e-mail e carteiras de motorista dos Estados Unidos.

Para obter informações sobre políticas que especificam identificadores de dados personalizados, consulteUsar identificadores de dados personalizados na política de proteção de dados.

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
            "Sid": "audit-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
```

```
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

Criar uma política de proteção de dados para um único grupo de logs

Você pode usar o console de CloudWatch registros ou AWS CLI os comandos para criar uma política de proteção de dados para mascarar dados confidenciais.

Você pode atribuir uma política de proteção de dados a cada grupo de logs. Cada política de proteção de dados pode auditar vários tipos de informações. Cada política de proteção de dados pode incluir uma declaração de auditoria.

Tópicos

- Console
- AWS CLI

Console

Para usar o console para criar uma política de proteção de dados

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

2. No painel de navegação, escolha Logs, Log groups (Grupos de log).
3. Escolha o nome do grupo de logs.
4. Escolha Actions (Ações), Create data protection policy (Criar política de proteção de dados).
5. Em Identificadores de dados gerenciados, selecione os tipos de dados que você deseja auditar e mascarar nesse grupo de registros. Você pode digitar na caixa de seleção para encontrar os identificadores que deseja.

   Recomendamos que você selecione apenas os identificadores de dados relevantes para seus dados de log e sua empresa. A escolha de muitos tipos de dados pode levar a falsos positivos.

   Para obter detalhes sobre quais tipos de dados você pode proteger usando identificadores de dados gerenciados, consulteTipos de dados que você pode proteger.
6. (Opcional) Se você quiser auditar e mascarar outros tipos de dados usando identificadores de dados personalizados, escolha Adicionar identificador de dados personalizado. Em seguida, insira um nome para o tipo de dados e a expressão regular a serem usados para pesquisar esse tipo de dados nos eventos de log. Para ter mais informações, consulte Identificadores de dados personalizados.

   Uma única política de proteção de dados pode incluir até 10 identificadores de dados personalizados. Cada expressão regular que define um identificador de dados personalizado deve ter 200 caracteres ou menos.
7. (Opcional) Escolha um ou mais serviços para enviar as descobertas da auditoria. Mesmo se você optar por não enviar descobertas de auditoria a nenhum desses serviços, os tipos de dados confidenciais selecionados ainda serão mascarados.
8. Escolha Activate data protection (Ativar proteção de dados).

AWS CLI

Para usar o AWS CLI para criar uma política de proteção de dados

1. Use um editor de texto para criar um arquivo de política chamado `DataProtectionPolicy.json`. Para obter informações sobre a sintaxe da política, consulte a seção a seguir.
2. Digite o comando :

```
aws logs put-data-protection-policy --log-group-identifier "my-log-group" --policy-
document file:///Path/DataProtectionPolicy.json --region us-west-2
```

Sintaxe da política de proteção de dados para AWS CLI nossas operações de API

Quando você cria uma política de proteção de dados JSON para usar em um AWS CLI comando ou operação de API, a política deve incluir dois blocos JSON:

- O primeiro bloco deve incluir uma matriz `DataIdentifer` e uma propriedade `Operation` com uma ação `Audit`. A matriz `DataIdentifer` faz uma lista com os tipos de dados sigilosos que você deseja mascarar. Para obter mais informações sobre as opções disponíveis, consulte Tipos de dados que você pode proteger.

  A propriedade `Operation` com uma ação `Audit` é necessária para encontrar os termos de dados confidenciais. Essa ação `Audit` deve conter um objeto `FindingsDestination`. Opcionalmente, você pode usar esse objeto `FindingsDestination` para listar um ou mais destinos para enviar relatórios de descobertas de auditoria. Se você especificar destinos como grupos de logs, streams do Amazon Data Firehose e buckets S3, eles já devem existir. Para obter um exemplo de um relatório de constatações de auditoria, consulte Relatórios de descobertas de auditoria.
- O segundo bloco deve incluir uma matriz `DataIdentifer` e uma propriedade `Operation` com uma ação `Deidentify`. A matriz `DataIdentifer` deve corresponder exatamente à matriz `DataIdentifer` no primeiro bloco da política.

  A propriedade `Operation` com a ação `Deidentify` é o que realmente mascara os dados e deve conter o objeto `"MaskConfig": {}`. O objeto `"MaskConfig": {}` deve estar vazio.

Veja a seguir um exemplo de política de proteção de dados que mascara endereços de e-mail e carteiras de habilitação dos Estados Unidos.

```
{
    "Name": "data-protection-policy",
    "Description": "test description",
    "Version": "2021-06-01",
    "Statement": [{
            "Sid": "audit-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
```

```
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "CloudWatchLogs": {
                            "LogGroup": "EXISTING_LOG_GROUP_IN_YOUR_ACCOUNT,"
                        },
                        "Firehose": {
                            "DeliveryStream": "EXISTING_STREAM_IN_YOUR_ACCOUNT"
                        },
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
                "arn:aws:dataprotection::aws:data-identifier/EmailAddress",
                "arn:aws:dataprotection::aws:data-identifier/DriversLicense-US"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {}
                }
            }
        }
    ]
}
```

Exibir dados não mascarados

Para visualizar dados não mascarados, o usuário deve ter a permissão `logs:Unmask`. Os usuários com essa permissão podem ver os dados não mascarados das seguintes maneiras:

- Ao visualizar os eventos em um fluxo de logs, escolha Display (Exibir), Unmask (Desmascarar).
- Use uma consulta do CloudWatch Logs Insights que inclua o comando unmask (@message). O exemplo de consulta a seguir exibe os 20 eventos de logs mais recentes no stream, desmascarados:

```
fields @timestamp, @message, unmask(@message)
| sort @timestamp desc
| limit 20
```

Para obter mais informações sobre CloudWatch os comandos do Logs Insights, consulteCloudWatch Sintaxe de consulta do Logs Insights.

- Use uma FilterLogEventsoperação GetLogEventsou com o `unmask` parâmetro.

A CloudWatchLogsFullAccesspolítica inclui a `logs:Unmask` permissão. Para conceder `logs:Unmask` a um usuário que não tem CloudWatchLogsFullAccess, você pode anexar uma política personalizada do IAM a esse usuário. Para obter mais informações, consulte Incluindo permissões para um usuário (console).

Relatórios de descobertas de auditoria

Se você configurar as políticas de auditoria de proteção de dados do CloudWatch Logs para escrever relatórios de auditoria no CloudWatch Logs, no Amazon S3 ou no Firehose, esses relatórios de descobertas serão semelhantes ao exemplo a seguir. CloudWatch O Logs grava um relatório de descobertas para cada evento de registro que contém dados confidenciais.

```
{
    "auditTimestamp": "2023-01-23T21:11:20Z",
    "resourceArn": "arn:aws:logs:us-west-2:111122223333:log-group:/aws/lambda/
MyLogGroup:*",
    "dataIdentifiers": [
        {
            "name": "EmailAddress",
            "count": 2,
            "detections": [
                {
                    "start": 13,
                    "end": 26
                },
{
                    "start": 30,
                    "end": 43
                }
            ]
        }
```

```
    ]
}
```

Os campos do relatório são os seguintes:

- O campo `resourceArn` exibe o grupo de logs onde os dados confidenciais foram encontrados.
- O objeto `dataIdentifiers` exibe informações sobre as descobertas de um tipo de dados confidenciais que você está auditando.
- O campo `name` identifica o tipo de dados confidenciais sobre os quais esta seção está relatando.
- O campo `count` exibe o número de vezes que esse tipo de dado confidencial aparece no evento de logs.
- Os campos `end` `estart` mostram onde no evento de logs, por contagem de caracteres, cada ocorrência dos dados confidenciais aparece.

O exemplo anterior mostra um relatório de localização de dois endereços de e-mail em um evento de logs. O primeiro endereço de e-mail começa no 13º caractere do evento de logs e termina no 26º caractere. O segundo endereço de e-mail vai do 30º caractere ao 43º caractere. Mesmo que esse evento de log tenha dois endereços de e-mail, o valor da métrica `LogEventsWithFindings` é incrementado apenas em um, porque essa métrica conta o número de eventos de log que contêm dados confidenciais, não o número de ocorrências de dados confidenciais.

Política-chave necessária para enviar os resultados da auditoria para um bucket protegido por AWS KMS

É possível proteger os dados em um bucket do Amazon S3 habilitando a criptografia no lado do servidor com as chaves gerenciadas pelo Amazon S3 (SSE-S3) ou a criptografia no lado do servidor com as chaves do KMS (SSE-KMS). Para obter mais informações, consulte Proteger dados usando criptografia do lado do servidor no Guia do usuário do Amazon S3.

Se você enviar as descobertas de auditoria para um bucket protegido por SSE-S3, nenhuma configuração adicional será necessária. O Amazon S3 lida com a chave de criptografia.

Se você enviar as descobertas de auditoria para um bucket protegido por SSE-KMS, deverá atualizar a política de chaves para a chave do KMS para que a conta de entrega de logs possa gravar no bucket do S3. Para obter mais informações sobre a política de chaves necessária para uso com o SSE-KMS, consulte Amazon S3 o Guia do usuário do Amazon CloudWatch Logs.

Tipos de dados que você pode proteger

Esta seção contém informações sobre os tipos de dados que você pode proteger em uma política de proteção de dados do CloudWatch Logs. CloudWatch Os identificadores de dados gerenciados de registros oferecem tipos de dados pré-configurados para proteger dados financeiros, informações pessoais de saúde (PHI) e informações de identificação pessoal (PII). Você também pode usar identificadores de dados personalizados para criar identificadores de dados personalizados para seu caso de uso específico.

Sumário

- CloudWatch Registra identificadores de dados gerenciados para tipos de dados confidenciais
  - Credenciais
    - ARNs de identificadores de dados para tipos de dados de credencial
  - Identificadores de dispositivo
    - ARNs de identificadores de dados para tipos de dados de dispositivos
  - Informações financeiras
    - ARNs de identificadores de dados para tipos de dados financeiros
  - Informações de saúde protegidas (PHI)
    - ARNs identificadores de dados para tipos de dados de informações de saúde protegidas (PHI)
  - Informações de identificação pessoal (PII)
    - Palavras-chave para números de identificação da carteira de habilitação
    - Palavras-chave para números de identificação nacional
    - Palavras-chave para números de passaporte
    - Palavras-chave para identificação do contribuinte e números de referência
    - ARNs do identificador de dados de informações de identificação pessoal (PII)
- Identificadores de dados personalizados
  - O que são identificadores de dados personalizados?
  - Restrições de identificadores de dados personalizados
  - Usando identificadores de dados personalizados no console
  - Usar identificadores de dados personalizados na política de proteção de dados

CloudWatch Registra identificadores de dados gerenciados para tipos de dados confidenciais

Esta seção contém informações sobre os tipos de dados que você pode proteger usando identificadores de dados gerenciados e quais países e regiões são relevantes para cada um desses tipos de dados.

Para alguns tipos de dados confidenciais, a proteção de dados do CloudWatch Logs verifica as palavras-chave nas proximidades dos dados e encontra uma correspondência somente se encontrar essa palavra-chave. Se uma palavra-chave precisar estar próxima de um tipo específico de dados, a palavra-chave normalmente precisará estar dentro de 30 caracteres (inclusive) dos dados.

Se uma palavra-chave contiver um espaço, a proteção de dados do CloudWatch Logs corresponderá automaticamente às variações de palavras-chave que não têm espaço ou que contêm um sublinhado (_) ou hífen (-) em vez do espaço. Em alguns casos, o CloudWatch Logs também expande ou abrevia uma palavra-chave para abordar variações comuns da palavra-chave.

As tabelas a seguir listam os tipos de credenciais, dispositivos, informações financeiras, médicas e de saúde protegidas (PHI) que o CloudWatch Logs pode detectar usando identificadores de dados gerenciados. Eles são uma adição a determinados tipos de dados que também podem ser qualificados como informações de identificação pessoal (PII).

Identificadores suportados independentes de idioma e região

Identificador	Categoria
`Address`	Pessoal
`AwsSecretKey`	Credenciais
`CreditCardExpiration`	Financeiro
`CreditCardNumber`	Financeiro
`CreditCardSecurityCode`	Financeiro
`EmailAddress`	Pessoal
`IpAddress`	Pessoal

Identificador	Categoria
`LatLong`	Pessoal
`Name`	Pessoal
`OpenSshPrivateKey`	Credenciais
`PgpPrivateKey`	Credenciais
`PkcsPrivateKey`	Credenciais
`PuttyPrivateKey`	Credenciais
`VehicleIdentificationNumber`	Pessoal

Os identificadores de dados dependentes da região devem incluir o nome do identificador, um hífen e os códigos de duas letras (ISO 3166-1 alfa-2). Por exemplo, `DriversLicense-US`.

Identificadores compatíveis que devem incluir um código de país ou região de duas letras

Identificador	Categoria	Países e idiomas
BankAccountNumber	Financeiro	DE, ES, FR, GB, IT
CepCode	Pessoal	BR
Cnpj	Pessoal	BR
CpfCode	Pessoal	BR
DriversLicense	Pessoal	AT, AU, BE, BG, CA, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IT, LT, LU, LV, MT, NL, PL, PT, RO, SE, SI, SK, US
DrugEnforcementAgencyNumber	Integridade	EUA

Identificador	Categoria	Países e idiomas
ElectoralRollNumber	Pessoal	GB
HealthInsuranceCardNumber	Integridade	UE
HealthInsuranceClaimNumber	Integridade	EUA
HealthInsuranceNumber	Integridade	FR
HealthcareProcedureCode	Integridade	EUA
IndividualTaxIdentification Number	Pessoal	EUA
InseeCode	Pessoal	FR
MedicareBeneficiaryNumber	Integridade	EUA
NationalDrugCode	Integridade	EUA
NationalIdentificationNumber	Pessoal	DE, ES, IT
NationalInsuranceNumber	Pessoal	GB
NationalProviderId	Integridade	EUA
NhsNumber	Integridade	GB
NieNumber	Pessoal	ES
NifNumber	Pessoal	ES
PassportNumber	Pessoal	CA, DE, ES, FR, GB, IT, US
PermanentResidenceNumber	Pessoal	CA
PersonalHealthNumber	Integridade	CA
PhoneNumber	Pessoal	BR, DE, ES, FR, GB, IT, US
PostalCode	Pessoal	CA

Identificador	Categoria	Países e idiomas
RgNumber	Pessoal	BR
SocialInsuranceNumber	Pessoal	CA
Ssn	Pessoal	ES, US
TaxId	Pessoal	DE, ES, FR, GB
ZipCode	Pessoal	EUA

Credenciais

CloudWatch A proteção de dados de registros pode encontrar os seguintes tipos de credenciais.

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões
AWS chave de acesso secreta	`AwsSecretKey`	`aws_secret_access_key` , `credentials` , `secret access key`, `secret key`, `set-awscredential`	Todos
Chave privada OpenSSH	`OpenSSHPrivateKey`	Nenhum	Todos
Chave privada PGP	`PgpPrivateKey`	Nenhum	Todos
Chave privada do Pkcs	`PkcsPrivateKey`	Nenhum	Todos
Chave privada PuTTY	`PuttyPrivateKey`	Nenhum	Todos

ARNs de identificadores de dados para tipos de dados de credencial

Veja a seguir os nomes de recurso da Amazon (ARNs) para os identificadores de dados que você pode adicionar às suas políticas de proteção de dados.

ARNs de identificadores de dados de credenciais
`arn:aws:dataprotection::aws:data-identifier/AwsSecretKey`
`arn:aws:dataprotection::aws:data-identifier/OpenSshPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PgpPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PkcsPrivateKey`
`arn:aws:dataprotection::aws:data-identifier/PuttyPrivateKey`

Identificadores de dispositivo

CloudWatch A proteção de dados de registros pode encontrar os seguintes tipos de identificadores de dispositivo.

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões
Endereço IP	`IpAddress`	Nenhum	Todos

ARNs de identificadores de dados para tipos de dados de dispositivos

Veja a seguir os nomes de recurso da Amazon (ARNs) para os identificadores de dados que você pode adicionar às suas políticas de proteção de dados.

ARN de identificador de dados de dispositivos
`arn:aws:dataprotection::aws:data-identifier/IpAddress`

Informações financeiras

CloudWatch A proteção de dados de registros pode encontrar os seguintes tipos de informações financeiras.

Se você definir uma política de proteção de dados, o CloudWatch Logs verificará os identificadores de dados que você especificar, independentemente da geolocalização em que o grupo de registros esteja localizado. As informações na coluna Países e regiões nesta tabela designam se os códigos de país de duas letras devem ser anexados ao identificador de dados para detectar as palavras-chave apropriadas para esses países e regiões.

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Número de conta bancária	`BankAccountNumber`	Sim. Palavras-chave diferentes se aplicam a diferentes países. Para obter detalhes, consulte a tabela Palavras-chave para números de contas bancárias mais adiante nesta seção.	França, Alemanh Itália, Espanha Reino Unido	Inclui números de contas bancárias internacionais (IBANs) que consistem em até 34 caracteres alfanuméricos, incluindo elementos como códigos de país.

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Data de validade do cartão de crédito	`CreditCardExpiration`	`exp d`, `exp m`, `exp y`, `expiration` , `expiry`	Todos	

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Números de cartão de crédito	`CreditCardNumber`	`account number`, `american express`, `amex`, `bank card`, `card`, `card number`, `card num`, `cc #`, `ccn`, `check card`, `credit`, `credit card#`, `dankort`, `debit`, `debit card`, `diners club`, `discover`, `electron`, `japanese card bureau`, `jcb`, `mastercard `, `mc`, `pan`, `payment account number`, `payment card number`, `pcn`, `union pay`, `visa`	Todos	A detecção exige que os dados sejam uma sequência de 13 a 19 dígitos que siga a fórmula de cheque de Luhn e use um prefixo de número de cartão padrão para qualquer um dos seguintes tipos

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
				de cartão de crédito: American Express, Dankort, Diner's Club, Discover, Electron, Japanese Card Bureau (JCB), Mastercard e Visa. UnionPay
Código de verificação do cartão de crédito	`CreditCardSecurityCode`	`card id`, `card identification code`, `card identification number` , `card security code`, `card validation code` , `card validation number` , `card verification data` , `card verification value`, `cvc`, `cvc2`, `cvv`, `cvv2`, `elo verification code`	Todos	

Palavras-chave para números de conta bancária

Use as palavras-chave a seguir para detectar números de conta bancária internacional (IBANs) compostos de até 34 caracteres alfanuméricos, inclusive elementos como códigos do país.

País	Palavras-chave
França	`account code`, `account number`, `accountno#` , `accountnumber#` , `bban`, `code bancaire`, `compte bancaire`, `customer account id`, `customer account number`, `customer bank account id`, `iban`, `numéro de compte`
Alemanha	`account code`, `account number`, `accountno#` , `accountnumber#` , `bankleitzahl` , `bban`, `customer account id`, `customer account number`, `customer bank account id`, `geheimzahl` , `iban`, `kartennummer` , `kontonummer` , `kreditkartennummer` , `sepa`
Itália	`account code`, `account number`, `accountno#` , `accountnumber#` , `bban`, `codice bancario`, `conto bancario`, `customer account id`, `customer account number`, `customer bank account id`, `iban`, `numero di conto`
Espanha	`account code`, `account number`, `accountno#` , `accountnumber#` , `bban`, `código cuenta`, `código cuenta bancaria`, `cuenta cliente id`, `customer account ID`, `customer account number`, `customer bank account id`, `iban`, `número cuenta bancaria cliente`, `número cuenta cliente`
Reino Unido	`account code`, `account number`, `accountno#` , `accountnumber#` , `bban`, `customer account ID`, `customer account number`, `customer bank account id`, `iban`, `sepa`
Estados Unidos	`bank account`, `bank acct`, `checking account`, `checking acct`, `deposit account`, `deposit acct`, `savings account`, `savings acct`, `chequing account`, `chequing acct`

CloudWatch Os registros não relatam as ocorrências das seguintes sequências, que os emissores de cartão de crédito reservaram para testes públicos.

```
122000000000003, 2222405343248877, 2222990905257051, 2223007648726984,
 2223577120017656,
30569309025904, 34343434343434, 3528000700000000, 3530111333300000, 3566002020360505,
 36148900647913,
36700102000000, 371449635398431, 378282246310005, 378734493671000, 38520000023237,
 4012888888881881,
4111111111111111, 4222222222222, 4444333322221111, 4462030000000000, 4484070000000000,
 4911830000000,
4917300800000000, 4917610000000000, 491761000000000003, 5019717010103742,
 5105105105105100,
5111010030175156, 5185540810000019, 5200828282828210, 5204230080000017,
 5204740009900014, 5420923878724339,
5454545454545454, 5455330760000018, 5506900490000436, 5506900490000444,
 5506900510000234, 5506920809243667,
5506922400634930, 5506927427317625, 5553042241984105, 5555553753048194,
 5555555555554444, 5610591081018250,
6011000990139424, 6011000400000000, 6011111111111117, 630490017740292441,
 630495060000000000,
6331101999990016, 6759649826438453, 6799990100000000019, and 76009244561.
```

ARNs de identificadores de dados para tipos de dados financeiros

Veja a seguir os nomes de recurso da Amazon (ARNs) para os identificadores de dados que você pode adicionar às suas políticas de proteção de dados.

ARNs de identificadores de dados financeiros
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-DE
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-ES
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-FR
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-GB
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-IT
arn:aws:dataprotection::aws:data-identifier/BankAccountNumber-US
arn:aws:dataprotection::aws:data-identifier/CreditCardExpiration
arn:aws:dataprotection::aws:data-identifier/CreditCardNumber

ARNs de identificadores de dados financeiros
`arn:aws:dataprotection::aws:data-identifier/CreditCardSecurityCode`

Informações de saúde protegidas (PHI)

CloudWatch A proteção de dados de registros pode encontrar os seguintes tipos de informações de saúde protegidas (PHI).

Se você definir uma política de proteção de dados, o CloudWatch Logs verificará os identificadores de dados que você especificar, independentemente da geolocalização em que o grupo de registros esteja localizado. As informações na coluna Países e regiões nesta tabela designam se os códigos de país de duas letras devem ser anexados ao identificador de dados para detectar as palavras-chave apropriadas para esses países e regiões.

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões
Número de registro da Agência Antidrogas (Drug Enforcement Agency, DEA	`DrugEnforcementAgencyNumber`	`dea number`, `dea registration`	Estados Unidos
Número do cartão de seguro de saúde (EHIC)	`HealthInsuranceCardNumber`	`assicurazione sanitaria numero`, `carta assicurazione numero`, `carte d’assurance maladie` , `carte européenne d'assurance maladie` , `ceam`, `ehic`, `ehic#`, `finlandehicnumber#` , `gesundheitskarte` , `hälsokort` , `health card`, `health card number`, `health insurance card`, `health`	União Europeia

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões
		insurance number, insurance card number, krankenversicherungskarte , krankenversicherungsnummer , medical account number, numero conto medico, numéro d’assurance maladie , numéro de carte d’assurance , numéro de compte medical, número de cuenta médica, número de seguro de salud, número de tarjeta de seguro, sairaanhoitokortin , sairausvakuutuskortti , sairausvakuutusnumero , sjukförsäkringnummer, sjukförsäkringskort , suomi ehic-numero , tarjeta de salud, terveyskortti , tessera sanitaria assicurazione numero , versicherungsnummer	

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões
Health Insurance Claim Number (HICN)	`HealthInsuranceClaimNumber`	`health insurance claim number`, `hic no`, `hic no.`, `hic number`, `hic#`, `hicn`, `hicn#`, `hicno#`	Estados Unidos
Número de identificação médica ou do seguro de saúde	`HealthInsuranceNumber`	`carte d'assuré social`, `carte vitale`, `insurance card`	França
Código do Healthcare Common Procedure Coding System (HCPCS)	`HealthcareProcedureCode`	`current procedural terminology` , `hcpcs`, `healthcare common procedure coding system`	Estados Unidos
Número do beneficiário do Medicare (MBN)	`MedicareBeneficiaryNumber`	`mbi`, `medicare beneficiary`	Estados Unidos
National Drug Code (NDC)	`NationalDrugCode`	`national drug code`, `ndc`	Estados Unidos
National Provider Identifier (NPI)	`NationalProviderId`	`hipaa`, `n.p.i.`, `national provider`, `npi`	Estados Unidos
Número do Serviço Nacional de Saúde (NHS)	`NhsNumber`	`national health service`, `NHS`	Grã-Bretanha
Número de saúde pessoal	`PersonalHealthNumber`	`canada healthcare number`, `msp number`, `care number`, `phn`, `soins de santé`	Canadá

ARNs identificadores de dados para tipos de dados de informações de saúde protegidas (PHI)

A lista a seguir lista os nomes de recursos da Amazon (ARNs) do identificador de dados que podem ser usados em políticas de proteção de dados de informações de saúde protegidas (PHI).

ARNs de identificadores de dados de PHI
`arn:aws:dataprotection::aws:data-identifier/DrugEnforcementAgencyNumber-US`
`arn:aws:dataprotection::aws:data-identifier/HealthcareProcedureCode-US`
`arn:aws:dataprotection::aws:data-identifier/HealthInsuranceCardNumber-EU`
`arn:aws:dataprotection::aws:data-identifier/HealthInsuranceClaimNumber-US`
`arn:aws:dataprotection::aws:data-identifier/HealthInsuranceNumber-FR`
`arn:aws:dataprotection::aws:data-identifier/MedicareBeneficiaryNumber-US`
`arn:aws:dataprotection::aws:data-identifier/NationalDrugCode-US`
`arn:aws:dataprotection::aws:data-identifier/NationalInsuranceNumber-GB`
`arn:aws:dataprotection::aws:data-identifier/NationalProviderId-US`
`arn:aws:dataprotection::aws:data-identifier/NhsNumber-GB`
`arn:aws:dataprotection::aws:data-identifier/PersonalHealthNumber-CA`

Informações de identificação pessoal (PII)

CloudWatch A proteção de dados de registros pode encontrar os seguintes tipos de informações de identificação pessoal (PII).

Se você definir uma política de proteção de dados, o CloudWatch Logs verificará os identificadores de dados que você especificar, independentemente da geolocalização em que o grupo de registros esteja localizado. As informações na coluna Países e regiões nesta tabela designam se os códigos de país de duas letras devem ser anexados ao identificador de dados para detectar as palavras-chave apropriadas para esses países e regiões.

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Datas de nascimento	`DateOfBirth`	`dob`, `date of birth`, `birthdate` , `birth date`, `birthday`, `b-day`, `bday`	Any	O suporte inclui a maioria dos formatos de data, como todos os dígitos e combinações de dígitos e nomes de meses. Os componentes de

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
				data podem ser separados por espaços, barras (/) ou hífens (-).
Código de endereçamento postal (CEP)	`CepCode`	`cep`, `código de endereçamento postal`, `codigo de endereçamento postal`	Brasil	
Cadastro Nacional da Pessoa Jurídica (CNPJ)	`Cnpj`	`cadastro nacional da pessoa jurídica`, `cadastro nacional da pessoa juridica`, `cnpj`	Brasil	
Cadastro de Pessoa Física (CPF)	`CpfCode`	`Cadastro de pessoas fisicas`, `cadastro de pessoas físicas`, `cadastro de pessoa física`, `cadastro de pessoa fisica`, `cpf`	Brasil	

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Número de identificação da carteira de habilitação	`DriversLicense`	Sim. Palavras-chave diferentes se aplicam a diferentes países. Para obter detalhes, consulte a tabela de números de identificação da carteira de habilitação posteriormente nesta seção.	Muitos países. Para obter detalhes, consulte a tabela de números de identificação da carteira.	
Número de registro eleitoral	`ElectoralRollNumber`	`electoral #`, `electoral number`, `electoral roll #`, `electoral roll no.`, `electoral roll number`, `electoral rollno`	Reino Unido	
Identificação do contribuinte individual	`IndividualTaxIdentificationNumber`	Sim. Palavras-chave diferentes se aplicam a diferentes países. Para obter detalhes, consulte a tabela de números de identificação da carteira nesta seção.	Brasil, França, Alemanh Espanha Reino Unido	

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Instituto Nacional de Estatística e Estudos Econômicos (INSEE)	`InseeCode`	Sim. Palavras-chave diferentes se aplicam a diferentes países. Para obter detalhes, consulte a tabela de números de identificação da carteira nesta seção.	França	

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Número de identificação nacional	`NationalIdentificationNumber`	Sim. Para obter detalhes, consulte a tabela de números de identificação da carteira nesta seção.	Alemanha Itália, Espanha	Isso inclui identificadores do Documento Nacional de Identidade (DNI) (Espanha), códigos fiscais (Itália) e números de carteira de identidade nacional (Alemanha).

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Número do Seguro Nacional (NINO)	`NationalInsuranceNumber`	`insurance no.`, `insurance number`, `insurance#` , `national insurance number`, `nationalinsurance#` , `nationalinsurancenumber` , `nin`, `nino`	Reino Unido	–
Número de identidad de extranjero (NIE)	`NieNumber`	Sim. Palavras-chave diferentes se aplicam a diferentes países. Para obter detalhes, consulte a tabela de números de identificação da carteira nesta seção.	Espanha	
Número de Identificación Fiscal (NIF)	`NifNumber`	Sim. Palavras-chave diferentes se aplicam a diferentes países. Para obter detalhes, consulte a tabela de números de identificação da carteira nesta seção.	Espanha	
Número de passaporte	`PassportNumber`	Sim. Palavras-chave diferentes se aplicam a diferentes países. Para obter detalhes, consulte a tabela Palavras-chave para números de passaporte mais adiante nesta seção.	Canadá, França, Alemanh Itália, Espanha Reino Unido, Estados	

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Número de residência permanente	`Permanent Residence Number`	`carte résident permanent , numéro carte résident permanent , numéro résident permanent , permanent resident card, permanent resident card number, permanent resident no, permanent resident no., permanent resident number, pr no, pr no., pr non, pr number, résident permanent no., résident permanent non`	Canadá	

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Número de telefone	PhoneNumber	Brasil: as palavras-chave também incluem: `cel`, `celular`, `fone`, `móvel`, `número residencial`, `numero residencial`, `telefone` Outros: `cell`, `contact`, `fax`, `fax number`, `mobile`, phone, `phone number`, `tel`, `telephone`, `telephone number`	Brasil, Canadá, França, Alemanh Itália, Espanha Reino Unido, Estados Unidos	Isso inclui números gratuitos nos Estados Unidos e números de fax. Se uma palavra-chave estiver próxima dos dados, o número não precisará incluir o código do país. Se uma palavra-chave estiver

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
				próximados dados, o número não precisará incluir o código do país.
CEP	`PostalCode`	Nenhum	Canadá	
Registro Geral (RG)	`RgNumber`	Sim. Palavras-chave diferentes se aplicam a diferentes países. Para obter detalhes, consulte a tabela de números de identificação da carteira nesta seção.	Brasil	
Número do Seguro Social (SIN)	`SocialInsuranceNumber`	`canadian id`, `numéro d'assurance sociale`, `social insurance number`, `sin`	Canadá	

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Número da Previdência Social (SSN)	`Ssn`	Espanha — `número de la seguridad social`, `social security no.`, `social security no.` `número de la seguridad social`, `social security number`, `socialsecurityno#` , `ssn`, `ssn#` Estados Unidos – `social security`, `ss#`, `ssn`	Espanha Estados Unidos	
Identificação do contribuinte ou número de referência	`TaxId`	Sim. Palavras-chave diferentes se aplicam a diferentes países. Para obter detalhes, consulte a tabela de números de identificação da carteira nesta seção. .	França, Alemanh Espanha Reino Unido	Isso inclui TIN (França); Steueridentifikationsnummer (Alemanha); CIF (Espanha) e TRN, UTR (Reino Unido).
Código postal	`ZipCode`	`zip code`, `zip+4`	Estados Unidos	Código postal dos Estados Unidos.

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Endereço postal	`Address`	Nenhum	Austrália, Canadá, França, Alemanh Itália, Espanha Reino Unido, Estados Unidos	Embora uma palavra-chave não seja necessária, a detecção exige que o endereço inclua o nome de uma cidade ou local e um CEP.
Endereço de correio eletrônico	`EmailAddress`	Nenhum	Any	

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Coordenadas do sistema de posicionamento global (GPS)	`LatLong`	`coordinate` , `coordinates` , `latlong`, `latitudelongitude` , `location`, `position`	Any	CloudWatch Os registros podem detectar coordenadas GPS se as coordenadas de latitude e longitude estiverem armazenadas como um par e estiverem no formato de graus decimais (DD), por exemplo, 41,948614, -87,65531

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
				1. O suporte não inclui coordenadas no formato graus, decimais, minutos (DDM), por exemplo, 41°56.9168'N 87°39.3187'W, ou no formato graus, minutos, segundos (DMS), por exemplo, 41°56'55.0104"N 87°39'19.1196"W.

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Nome completo	`Name`	Nenhum	Any	CloudWatch Os registros podem detectar somente nomes completos. O suporte é limitado aos conjuntos de caracteres latinos.

Tipo de dados	ID do identificador de dados	Palavra-chave obrigatória	Países e regiões	Observações
Número de identificação de veículo (VIN)	`VehicleIdentificationNumber`	`Fahrgestellnummer` , `niv`, `numarul de identificare` , `numarul seriei de sasiu`, `serie sasiu`, `numer VIN`, `Número de Identificação do Veículo`, `Número de Identificación de Automóviles` , `numéro d'identification du véhicule`, `vehicle identification number`, `vin`, `VIN numeris`	Any	CloudWatch Os registros podem detectar VINs que consistem em uma sequência de 17 caracteres e seguem os padrões ISO 3779 e 3780. Esses padrões foram projetados para uso em todo o mundo.

Palavras-chave para números de identificação da carteira de habilitação

Para detectar vários tipos de números de identificação da carteira de motorista, o CloudWatch Logs exige que uma palavra-chave esteja próxima dos números. A tabela a seguir lista as palavras-chave que o CloudWatch Logs reconhece para países e regiões específicos.

País ou região	Palavras-chave
Austrália	dl# dl:, dl :, dlno# driver licence, driver license, driver permit, drivers lic., drivers licence, driver's licence, drivers license, driver's license, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
Áustria	führerschein, fuhrerschein, führerschein republik österreich, fuhrerschein republik osterreich
Bélgica	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer, numéro permis conduire, permis de conduire, rijbewijs, rijbewijsnummer
Bulgária	превозно средство, свидетелство за управление на моторно, свидетелство за управление на мпс, сумпс, шофьорска книжка
Canadá	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit,

País ou região	Palavras-chave
	drivers permit number, driving licence, driving license, driving permit, permis de conduire
Croácia	vozačka dozvola
Chipre	άδεια οδήγησης
República Tcheca	číslo licence, číslo licence řidiče, číslo řidičskéh o průkazu, ovladače lic., povolení k jízdě, povolení řidiče, řidiči povolení, řidičský prúkaz, řidičský průkaz
Dinamarca	kørekort, kørekortnummer
Estônia	juhi litsentsi number, juhiloa number, juhiluba, juhiluba number
Finlândia	ajokortin numero, ajokortti, förare lic., körkort, körkort nummer, kuljettaja lic., permis de conduire
França	permis de conduire
Alemanha	fuehrerschein, fuehrerschein- nr, fuehrersc heinnummer, fuhrerschein, führerschein, fuhrerschein- nr, führerschein- nr, fuhrersch einnummer, führerscheinnummer
Grécia	δεια οδήγησης, adeia odigisis
Hungria	illesztőprogramok lic, jogosítvány, jogsi, licencszám, vezető engedély, vezetői engedély
Irlanda	ceadúnas tiomána
Itália	patente di guida, patente di guida numero, patente guida, patente guida numero

País ou região	Palavras-chave
Letônia	autovadītāja apliecība, licences numurs, vadītāja apliecība, vadītāja apliecības numurs, vadītāja atļauja, vadītāja licences numurs, vadītāji lic.
Lituânia	vairuotojo pažymėjimas
Luxemburgo	fahrerlaubnis, führerschäin
Malta	liċenzja tas-sewqan
Holanda	permis de conduire, rijbewijs, rijbewijsnummer
Polônia	numer licencyjny, prawo jazdy, zezwolenie na prowadzenie
Portugal	carta de condução, carteira de habilitação, carteira de motorist, carteira habilitação, carteira motorist, licença condução, licença de condução, número de licença, número licença, permissão condução, permissão de condução
Romênia	numărul permisului de conducere, permis de conducere
Eslováquia	číslo licencie, číslo vodičského preukazu, ovládače lic., povolenia vodičov, povolenie jazdu, povolenie na jazdu, povolenie vodiča, vodičský preukaz
Eslovênia	vozniško dovoljenje

País ou região	Palavras-chave
Espanha	carnet conducer, el carnet de conducer, licencia conducer, licencia de manejo, número carnet conducer, número de carnet de conducer, número de permiso conducer, número de permiso de conducer, número licencia conducer, número permiso conducer, permiso conducción, permiso conducer, permiso de conducción
Suécia	ajokortin numero, dlno# ajokortti, drivere lic., förare lic., körkort, körkort nummer, körkortsn ummer, kuljettajat lic.
Reino Unido	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit
Estados Unidos	dl#, dl:, dlno#, driver licence, driver licences, driver license, driver licenses, driver permit, drivers lic., drivers licence, driver's licence, drivers licences, driver's licences, drivers license, driver's license, drivers licenses, driver's licenses, drivers permit, driver's permit, drivers permit number, driving licence, driving license, driving permit

Palavras-chave para números de identificação nacional

Para detectar vários tipos de números de identificação nacional, o CloudWatch Logs exige que uma palavra-chave esteja próxima dos números. Isso inclui identificadores do Documento Nacional

de Identidad (DNI) (Espanha), códigos do Instituto Nacional de Estatística e Estudos Econômicos (INSEE) da França, números da carteira de identidade nacional alemã e números do Registro Geral (RG) (Brasil).

A tabela a seguir lista as palavras-chave que o CloudWatch Logs reconhece para países e regiões específicos.

País ou região	Palavras-chave
Brasil	registro geral, rg
França	assurance sociale, carte nationale d’identit é, cni, code sécurité sociale, French social security number, fssn#, insee, insurance number, national id number, nationalid#, numéro d'assurance, sécurité sociale, sécurité sociale non., sécurité sociale numéro, social, social security, social security number, socialsecuritynumber, ss#, ssn, ssn#
Alemanha	ausweisnummer, id number, identification number, identity number, insurance number, personal id, personalausweis
Itália	codice fiscal, dati anagrafici, ehic, health card, health insurance card, p. iva, partita i.v.a., personal data, tax code, tessera sanitaria
Espanha	dni, dni#, dninúmero#, documento nacional de identidad, identidad único, identidadúnico#, insurance number, national identification number, national identity, nationalid#, nationali dno#, número nacional identidad, personal identification number, personal identity no, unique identity number, uniqueid#

Palavras-chave para números de passaporte

Para detectar vários tipos de números de passaporte, o CloudWatch Logs exige que uma palavra-chave esteja próxima dos números. A tabela a seguir lista as palavras-chave que o CloudWatch Logs reconhece para países e regiões específicos.

País ou região	Palavras-chave
Canadá	passeport, passeport#, passport, passport#, passportno, passportno#
França	numéro de passeport, passeport, passeport #, passeport #, passeportn °, passeport n °, passeportNon, passeport non
Alemanha	ausstellungsdatum, ausstellungsort, geburtsdatum, passport, passports, reisepass, reisepass–nr, reisepassnummer
Itália	italian passport number, numéro passeport, numéro passeport italien, passaporto, passaporto italiana, passaporto numero, passport number, repubblica italiana passaporto
Espanha	españa pasaporte, libreta pasaporte, número pasaporte, pasaporte, passport, passport book, passport no, passport number, spain passport
Reino Unido	passeport #, passeport n °, passeportNon, passeport non, passeportn °, passport #, passport no, passport number, passport#, passportid
Estados Unidos	passport, travel document

Palavras-chave para identificação do contribuinte e números de referência

Para detectar vários tipos de identificação de contribuintes e números de referência, o CloudWatch Logs exige que uma palavra-chave esteja próxima dos números. A tabela a seguir lista as palavras-chave que o CloudWatch Logs reconhece para países e regiões específicos.

País ou região	Palavras-chave
Brasil	cadastro de pessoa física, cadastro de pessoa fisica, cadastro de pessoas físicas, cadastro de pessoas fisicas, cadastro nacional da pessoa jurídica, cadastro nacional da pessoa juridica, cnpj, cpf
França	numéro d'identification fiscale, tax id, tax identification number, tax number, tin, tin#
Alemanha	identifikationsnummer, steuer id, steueride ntifikationsnummer, steuernummer, tax id, tax identification number, tax number
Espanha	cif, cif número, cifnúmero#, nie, nif, número de contribuyente, número de identidad de extranjero, número de identificación fiscal, número de impuesto corporativo, personal tax number, tax id, tax identification number, tax number, tin, tin#
Reino Unido	paye, tax id, tax id no., tax id number, tax identification, tax identification#, tax no., tax number, tax reference, tax#, taxid#, temporary reference number, tin, trn, unique tax reference, unique taxpayer reference, utr
Estados Unidos	número de identificação de contribuinte individual, itin, i.t.i.n.

ARNs do identificador de dados de informações de identificação pessoal (PII)

A tabela a seguir lista os nomes de recursos da Amazon (ARNs) para os identificadores de dados de informações de identificação pessoal (PII) que você pode adicionar às suas políticas de proteção de dados.

ARNs de identificadores de dados de PII
`arn:aws:dataprotection::aws:data-identifier/Address`
`arn:aws:dataprotection::aws:data-identifier/CepCode-BR`
`arn:aws:dataprotection::aws:data-identifier/Cnpj-BR`
`arn:aws:dataprotection::aws:data-identifier/CpfCode-BR`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-AT`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-AU`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-BE`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-BG`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-CA`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-CY`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-CZ`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-DE`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-DK`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-EE`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-ES`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-FI`
`arn:aws:dataprotection::aws:data-identifier/DriversLicense-FR`

ARNs de identificadores de dados de PII
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GB
arn:aws:dataprotection::aws:data-identifier/DriversLicense-GR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HR
arn:aws:dataprotection::aws:data-identifier/DriversLicense-HU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-IT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LU
arn:aws:dataprotection::aws:data-identifier/DriversLicense-LV
arn:aws:dataprotection::aws:data-identifier/DriversLicense-MT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-NL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PL
arn:aws:dataprotection::aws:data-identifier/DriversLicense-PT
arn:aws:dataprotection::aws:data-identifier/DriversLicense-RO
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SE
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SI
arn:aws:dataprotection::aws:data-identifier/DriversLicense-SK
arn:aws:dataprotection::aws:data-identifier/DriversLicense-US
arn:aws:dataprotection::aws:data-identifier/ElectoralRollNumber-GB
arn:aws:dataprotection::aws:data-identifier/EmailAddress

ARNs de identificadores de dados de PII
arn:aws:dataprotection::aws:data-identifier/IndividualTaxIdentificationNumber-US
arn:aws:dataprotection::aws:data-identifier/InseeCode-FR
arn:aws:dataprotection::aws:data-identifier/LatLong
arn:aws:dataprotection::aws:data-identifier/Name
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-DE
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-ES
arn:aws:dataprotection::aws:data-identifier/NationalIdentificationNumber-IT
arn:aws:dataprotection::aws:data-identifier/NieNumber-ES
arn:aws:dataprotection::aws:data-identifier/NifNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-CA
arn:aws:dataprotection::aws:data-identifier/PassportNumber-DE
arn:aws:dataprotection::aws:data-identifier/PassportNumber-ES
arn:aws:dataprotection::aws:data-identifier/PassportNumber-FR
arn:aws:dataprotection::aws:data-identifier/PassportNumber-GB
arn:aws:dataprotection::aws:data-identifier/PassportNumber-IT
arn:aws:dataprotection::aws:data-identifier/PassportNumber-US
arn:aws:dataprotection::aws:data-identifier/PermanentResidenceNumber-CA

ARNs de identificadores de dados de PII
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-BR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-DE
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-ES
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-FR
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-GB
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-IT
arn:aws:dataprotection::aws:data-identifier/PhoneNumber-US
arn:aws:dataprotection::aws:data-identifier/PostalCode-CA
arn:aws:dataprotection::aws:data-identifier/RgNumber-BR
arn:aws:dataprotection::aws:data-identifier/SocialInsuranceNumber-CA
arn:aws:dataprotection::aws:data-identifier/Ssn-ES
arn:aws:dataprotection::aws:data-identifier/Ssn-US
arn:aws:dataprotection::aws:data-identifier/TaxId-DE
arn:aws:dataprotection::aws:data-identifier/TaxId-ES
arn:aws:dataprotection::aws:data-identifier/TaxId-FR
arn:aws:dataprotection::aws:data-identifier/TaxId-GB
arn:aws:dataprotection::aws:data-identifier/VehicleIdentificationNumber
arn:aws:dataprotection::aws:data-identifier/ZipCode-US

Identificadores de dados personalizados

Tópicos

- O que são identificadores de dados personalizados?
- Restrições de identificadores de dados personalizados
- Usando identificadores de dados personalizados no console
- Usar identificadores de dados personalizados na política de proteção de dados

O que são identificadores de dados personalizados?

Os identificadores de dados personalizados (CDIs) permitem que você defina suas próprias expressões regulares personalizadas que podem ser usadas na política de proteção de dados. Usando identificadores de dados personalizados, é possível segmentar casos de uso de informações de identificação pessoal (PII) específicos da empresa que os identificadores de dados gerenciados não podem fornecer. Por exemplo, você pode usar um identificador de dados personalizado para procurar IDs de funcionários específicos da empresa. Identificadores de dados personalizados podem ser usados em conjunto com identificadores de dados gerenciados.

Restrições de identificadores de dados personalizados

CloudWatch Os identificadores de dados personalizados de registros têm as seguintes limitações:

- As políticas de proteção de dados são compatíveis com dez identificadores de dados personalizados, no máximo.
- O tamanho máximo dos nomes de identificadores de dados personalizados é 128 caracteres. Os seguintes caracteres são aceitos:
  - Alfanuméricos: (a – z; A – Z; 0 – 9)
  - Símbolos: (“_” | “-”)
- O RegEx tem um tamanho máximo de 200 caracteres. Os seguintes caracteres são aceitos:
  - Alfanuméricos: (a – z; A – Z; 0 – 9)
  - Símbolos: (“_” | “#” | “=” | “@” | “/” | “;” | “,” | “-” | “ ”)
  - Caracteres reservados RegEx: (“^” | “$” | “?” | “[“ | “]” | “{“ | “}” | “|” | “\\” | “*” | “+” | “.”)
- Os identificadores de dados personalizados não podem compartilhar o mesmo nome de um identificador de dados gerenciados.

- Identificadores de dados personalizados podem ser especificados em uma política de proteção de dados em nível de conta ou em políticas de proteção de dados em nível de grupo de registros. Semelhantes aos identificadores de dados gerenciados, os identificadores de dados personalizados definidos em uma política em nível de conta funcionam em combinação com identificadores de dados personalizados definidos em uma política em nível de grupo de registros.

Usando identificadores de dados personalizados no console

Ao usar o CloudWatch console para criar ou editar uma política de proteção de dados, para especificar um identificador de dados personalizado, basta inserir um nome e uma expressão regular para o identificador de dados. Por exemplo, você pode inserir **`Employee_ID`** como nome e **`EmployeeID-\d{9}`** como expressão regular. Essa expressão regular detectará e mascarará eventos de log com nove números depois`EmployeeID-`. Por exemplo, `EmployeeID-123456789`.

Usar identificadores de dados personalizados na política de proteção de dados

Se você estiver usando a AWS API AWS CLI ou para especificar um identificador de dados personalizado, precisará incluir o nome do identificador de dados e a expressão regular na política JSON usada para definir a política de proteção de dados. A política de proteção de dados a seguir detecta e mascara eventos de registro que contêm IDs de funcionários específicos da empresa.

1. Crie um bloco `Configuration` na política de proteção de dados.
2. Insira um `Name` para o identificador de dados personalizado. Por exemplo, **`EmployeeId`**.
3. Insira um `Regex` para o identificador de dados personalizado. Por exemplo, **`EmployeeID-\d{9}`**. Essa expressão regular corresponderá a eventos de log contendo `EmployeeID-` nove dígitos depois`EmployeeID-`. Por exemplo, `EmployeeID-123456789`.
4. Consulte o identificador de dados personalizado a seguir em uma declaração de política.

```
{
    "Name": "example_data_protection_policy",
    "Description": "Example data protection policy with custom data identifiers",
    "Version": "2021-06-01",
    "Configuration": {
      "CustomDataIdentifier": [
        {"Name": "EmployeeId", "Regex": "EmployeeId-\\d{9}"}
      ]
    },
    "Statement": [
        {
```

```
            "Sid": "audit-policy",
            "DataIdentifier": [
                "EmployeeId"
            ],
            "Operation": {
                "Audit": {
                    "FindingsDestination": {
                        "S3": {
                            "Bucket": "EXISTING_BUCKET"
                        }
                    }
                }
            }
        },
        {
            "Sid": "redact-policy",
            "DataIdentifier": [
            "EmployeeId"
            ],
            "Operation": {
                "Deidentify": {
                    "MaskConfig": {
                    }
                }
            }
        }
    ]
}
```

5. (Opcional) Continue adicionando identificadores de dados personalizados ao bloco `Configuration`, conforme necessário. No momento, as políticas de proteção de dados são compatíveis com dez identificadores de dados personalizados, no máximo.

Criar métricas de eventos de log usando filtros

Você pode pesquisar e filtrar os dados de registro que chegam ao CloudWatch Logs criando um ou mais filtros de métrica. Os filtros de métricas definem os termos e padrões a serem procurados nos dados de registro à medida que são enviados para o CloudWatch Logs. CloudWatch O Logs usa esses filtros métricos para transformar dados de registro em CloudWatch métricas numéricas que você pode representar graficamente ou ativar um alarme.

Ao criar uma métrica de um filtro de log, você também pode atribuir dimensões e uma unidade à métrica. Se você especificar uma unidade, certifique-se de especificar a correta ao criar o filtro. Alterar a unidade do filtro posteriormente não terá efeito.

> **Note**
>
> Os filtros métricos são compatíveis somente com grupos de registros na classe de registros Standard. Para obter mais informações sobre classes de log, consulteClasses de log.

Você pode usar qualquer tipo de CloudWatch estatística, incluindo estatísticas percentuais, ao visualizar essas métricas ou definir alarmes.

> **Note**
>
> As estatísticas de percentual 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 percentual não estarão disponíveis para essa métrica quando ela tiver números negativos como valores. Para obter mais informações, consulte Percentuais.

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.

Conteúdo

- Conceitos

- Sintaxe de padrões de filtros para filtros de métricas
- Criar filtros de métrica
- Listagem de filtros de métrica
- Excluir um filtro de métrica

Conceitos

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

valor padrão

O valor indicado para o filtro de métrica durante um período em que logs são ingeridos, mas não se encontram logs correspondentes. Ao definir esse valor como 0, você garante que os dados sejam relatados durante cada período, impedindo métricas irregulares em períodos sem dados correspondentes. Se nenhum log for ingerido durante um período de um minuto, nenhum valor será relatado.

Se você atribuir dimensões a uma métrica criada por um filtro de métrica, não será possível atribuir um valor padrão a essa métrica.

dimensões

Dimensões são os pares chave-valor que definem melhor uma métrica. Também é possível atribuir dimensões à métrica criada de um filtro de métrica. Como as dimensões fazem parte do identificador exclusivo de uma métrica, sempre que um par de nome/valor exclusivo for extraído de uma de suas métricas, você criará uma nova variação daquela métrica.

padrão de filtro

Uma descrição simbólica de como CloudWatch os registros devem interpretar os dados em cada evento de registro. 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.

metric name (nome da métrica)

O nome da CloudWatch métrica na qual as informações de registro monitoradas devem ser publicadas. Por exemplo, você pode publicar em uma métrica chamada ErrorCount.

namespace de métrica

O namespace de destino da nova CloudWatch métrica.

valor da métrica

O valor numérico para publicar a métrica cada vez que um log correspondente for 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.

Sintaxe de padrões de filtros para filtros de métricas

> **Note**
>
> Como os filtros de métricas diferem nas consultas do CloudWatch Logs Insights
> Os filtros de métrica diferem das consultas do CloudWatch Logs Insights porque um valor numérico especificado é adicionado a um filtro de métrica sempre que um registro correspondente é encontrado. Para ter mais informações, consulte Configurando valores métricos para um filtro de métrica.
> Para obter informações sobre como consultar seus grupos de CloudWatch logs com a linguagem de consulta Amazon Logs Insights, consulteCloudWatch Sintaxe de consulta do Logs Insights.
> Exemplos genéricos de padrões de filtros
> Para obter mais informações sobre a sintaxe de padrões de filtros genéricos aplicável a filtros de métricas, bem como filtros de assinatura e filtros de eventos de log, consulte Sintaxe de padrões de filtros para filtros de métricas, filtros de assinatura e filtros de eventos de log, que inclui os seguintes exemplos:
>
> - Sintaxe de expressões regulares (regex) compatíveis
> - Como fazer a correspondência de termos em eventos de log não estruturados
> - Correspondência de termos em eventos de log JSON
> - Como fazer a correspondência de termos em eventos de log delimitados por espaços

Os filtros de métricas permitem pesquisar e filtrar dados de registro que chegam aos CloudWatch registros, extrair observações métricas dos dados de registro filtrados e transformar os pontos de dados em uma métrica de CloudWatch registros. Você define os termos e padrões a serem procurados nos dados de registro à medida que eles são enviados para o CloudWatch Logs. 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.

Quando um filtro de métrica corresponde a um termo, ele incrementa a contagem da métrica em um valor numérico especificado. Por exemplo, você pode criar um filtro de métrica que conte a ocorrência da palavra ERRO em seus eventos de log.

É possível atribuir unidades de medida e dimensões a métricas. Por exemplo, se você criar um filtro de métrica que conte as vezes que a palavra ERROR ocorre em seus eventos de log, poderá especificar uma dimensão chamada `ErrorCode` para mostrar o número total de eventos de log que contenham a palavra ERROR (ERRO) e filtrar os dados por códigos de erro relatados.

> **Tip**
>
> Ao atribuir uma unidade de medida a uma métrica, certifique-se de especificar a correta. Se você alterar a unidade depois, sua alteração pode não entrar em vigor. Para obter a lista completa das unidades que oferecem CloudWatch suporte, consulte MetricDatuma Amazon CloudWatch API Reference.

Tópicos

- Configurando valores métricos para um filtro de métrica
- Publicar dimensões com métricas de valores em eventos de log JSON ou delimitados por espaços
- Usando valores em eventos de log para incrementar o valor de uma métrica

Configurando valores métricos para um filtro de métrica

Ao criar um filtro de métrica, você define seu padrão de filtro e especifica o valor da métrica e o valor padrão. Você pode definir valores de métrica para números, identificadores nomeados ou identificadores numéricos. Se você não especificar um valor padrão, CloudWatch não reportará dados quando seu filtro de métricas não encontrar uma correspondência. Recomendamos que você especifique um valor padrão, mesmo que o valor seja 0. Definir um valor padrão ajuda a CloudWatch relatar dados com mais precisão e CloudWatch evita a agregação de métricas irregulares. CloudWatch agrega e relata valores métricos a cada minuto.

Quando o filtro de métrica encontra uma correspondência em seus eventos de log, ele incrementa a contagem da métrica de acordo com o valor da métrica. Se seu filtro de métricas não encontrar

uma correspondência, CloudWatch informa o valor padrão da métrica. Por exemplo, seu grupo de logs publica dois registros a cada minuto, o valor de métrica é 1 e o valor padrão é 0. Se o filtro de métrica encontrar correspondências nos dois registros de log no primeiro minuto, o valor de métrica para aquele minuto será 2. Se o filtro de métrica não encontrar correspondências em nenhum dos dois registros durante o segundo minuto, o valor padrão para aquele minuto será 0. Se você atribuir dimensões a métricas geradas por filtros de métrica, não será possível especificar valores padrão para essas métricas.

Você também pode configurar um filtro de métrica para incrementar uma métrica com um valor extraído de um evento de log, em vez de um valor estático. Para obter mais informações, consulte Usando valores em eventos de log para incrementar o valor de uma métrica.

Publicar dimensões com métricas de valores em eventos de log JSON ou delimitados por espaços

Você pode usar o CloudWatch console ou a AWS CLI para criar filtros métricos que publicam dimensões com métricas geradas por JSON e eventos de log delimitados por espaço. As dimensões são pares de valor nome/valor e só estão disponíveis para padrões de filtro JSON e delimitados por espaço. Você pode criar filtros de métrica JSON e delimitados por espaço com até três dimensões. Para obter mais informações sobre dimensões e informações sobre como atribuir dimensões a métricas, consulte as seguintes seções:

- Dimensões no guia do CloudWatch usuário da Amazon
- Exemplo: extraia campos de um log do Apache e atribua dimensões no Guia do usuário do Amazon CloudWatch Logs

> **Important**
>
> As dimensões contêm valores que coletam cobranças iguais às métricas personalizadas. Para evitar cobranças inesperadas, não especifique campos de alta cardinalidade, por exemplo, `IPAddress` ou `requestID`, como dimensões.
> Se você extrair métricas de eventos de log, você será cobrado por métricas personalizadas. Para evitar cobranças elevadas acidentais, a Amazon pode desabilitar seu filtro de métrica se este gerar 1.000 pares de nome/valor diferentes para as dimensões especificadas em um período de tempo específico.

Você pode criar alarmes de faturamento que o notificam sobre suas cobranças estimadas. Para obter mais informações, consulte Criação de um alarme de cobrança para monitorar suas AWS cobranças estimadas.

Publicar dimensões com métricas de eventos de log JSON

Os exemplos a seguir contêm trechos de código que descrevem como especificar dimensões em um filtro de métrica JSON.

Example: JSON log event

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

}
```

Note

Se você testar o filtro de métrica de exemplo com o exemplo de evento de log JSON, deverá inserir o exemplo de log JSON em uma única linha.

Example: Metric filter

O filtro de métrica incrementa a métrica sempre que um evento de log JSON contiver as propriedades `eventType` e `"sourceIPAddress"`.

```
{ $.eventType = "*" && $.sourceIPAddress != 123.123.* }
```

Ao criar um filtro de métrica JSON, é possível especificar qualquer uma das propriedades no filtro de métrica como dimensão. Por exemplo, para definir `eventType` como uma dimensão, use o seguinte:

```
"eventType" : $.eventType
```

A métrica de exemplo contém uma dimensão nomeada `"eventType"`, e o valor da dimensão no evento de log de exemplo é `"UpdateTrail"`.

Publicar dimensões com métricas de eventos de log delimitados por espaços

Os exemplos a seguir contêm trechos de código que descrevem como especificar dimensões em um filtro de métrica delimitado por espaço.

Example: Space-delimited log event

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

Example: Metric filter

```
[ip, server, username, timestamp, request, status_code, bytes > 1000]
```

O filtro de métrica incrementa a métrica quando um evento de log delimitado por espaço inclui qualquer um dos campos especificados no filtro. Por exemplo, o filtro de métrica encontra os seguintes campos e valores no exemplo de evento de log delimitado por espaço.

```
{
   "$bytes": "1534",
   "$status_code": "404",

   "$request": "GET /index.html HTTP/1.0",
   "$timestamp": "10/Oct/2000:13:25:15 -0700",
   "$username": "frank",
   "$server": "Prod",
   "$ip": "127.0.0.1"
}
```

Ao criar um filtro de métrica delimitado por espaço, é possível especificar qualquer um dos campos no filtro de métrica como uma dimensão. Por exemplo, para definir `server` como uma dimensão, use o seguinte:

```
"server" : $server
```

O filtro de métrica de exemplo contém uma dimensão nomeada `server`, e o valor da dimensão no evento de log de exemplo é `"Prod"`.

Example: Match terms with AND (&&) and OR (||)

Você pode usar os operadores lógicos AND ("&&") e OR ("||") para criar filtros de métrica delimitados por espaço que contenham condições. O filtro de métrica a seguir retorna eventos de log em que a primeira palavra nos eventos é ERROR ou qualquer string que contenha WARN.

```
[w1=ERROR || w1=%WARN%, w2]
```

Usando valores em eventos de log para incrementar o valor de uma métrica

Você pode criar filtros de métrica que publicam valores numéricos encontrados em eventos de log. O procedimento nesta seção usa o seguinte exemplo de filtro de métrica para mostrar como você pode publicar um valor numérico em um evento de log JSON em uma métrica.

```
{ $.latency = * } metricValue: $.latency
```

Criar um filtro de métrica que publique um valor em um evento de log

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Logs e, em seguida, escolha Grupos de log.
3. Selecione ou crie um grupo de logs.

   Para obter informações sobre como criar um grupo de registros, consulte Criar um grupo de CloudWatch registros em Logs no Guia do usuário do Amazon CloudWatch Logs.
4. Escolha Actions (Ações) e Create metric filter (Criar filtro de métrica).
5. Em Padrão de filtro, digite **`{ $.latency = * }`** e escolha Próximo.
6. Em Nome da métrica, insira myMetric.
7. Em Valor da métrica, insira **`$.latency`**.
8. (Opcional) Em Valor padrão, insira 0 e escolha Próximo.

   Recomendamos que você especifique um valor padrão, mesmo que o valor seja 0. Definir um valor padrão ajuda a CloudWatch relatar dados com mais precisão e CloudWatch evita a agregação de métricas irregulares. CloudWatch agrega e relata valores métricos a cada minuto.
9. Escolha Criar filtro de métrica.

O filtro de métrica de exemplo corresponde ao termo `"latency"` no exemplo de evento de log JSON e publica um valor numérico de 50 na métrica MyMetric.

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

Criar filtros de métrica

Os procedimentos e exemplos a seguir mostram como criar filtros de métrica.

Exemplos

- Criar um filtro de métrica para um grupo de logs
- Exemplo: contar eventos de log
- Exemplo: contar as ocorrências de um termo

- Exemplo: contar códigos HTTP 404
- Exemplo: contar códigos HTTP 4xx
- Exemplo: Extrair campos de um log Apache e atribuir dimensões

Criar um filtro de métrica para um grupo de logs

Para criar um filtro de métrica para um grupo de logs, siga estas etapas. A métrica não será visível até que haja alguns pontos de dados para ela.

Para criar um filtro métrico usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Logs e, em seguida, escolha Grupos de log.
3. Escolha o nome do grupo de logs.
4. Escolha `Actions` e Create metric filter (Criar filtro de métrica).
5. Em Filter pattern (Padrão de filtro), insira o padrão de filtro. Para ter mais informações, consulte Sintaxe de padrões de filtros para filtros de métricas, filtros de assinatura, filtros de eventos de log e Live Tail..
6. (Opcional) Para testar seu padrão de filtro, em Test Pattern (Testar padrão), insira um ou mais eventos de logs para testar o padrão. Cada evento de logs deve ser formatado em uma linha. Quebras de linha são usadas para separar eventos de logs na caixa Log event messages (Mensagens de log de eventos).
7. Escolha Next (Próximo) e digite um nome para o filtro de métrica.
8. Em Detalhes da métrica, em Namespace métrica, insira um nome para o CloudWatch namespace em que a métrica será publicada. Se esse namespace ainda não existir, certifique-se de que a opção Create new (Criar novo) esteja selecionada.
9. Em Metric name (Nome da métrica), insira um nome para a nova métrica.
10. Em Metric value (Valor da métrica), se o filtro de métrica estiver contando ocorrências das palavras-chave no filtro, digite 1. Isso incrementa a métrica em 1 para cada evento de log que contém uma das palavras-chave.

    Se preferir, insira um token, como **`$size`**. Isso incrementa a métrica pelo valor do número no campo `size` para cada evento de log que contém um campo `size`.
11. (Opcional) Em Unit (Unidade), selecione uma unidade para atribuir à métrica. Se você não especificar uma unidade, a unidade será definida como `None`.

12. (Opcional) Insira os nomes e tokens de até três dimensões para a métrica. Se você atribuir dimensões a métricas criadas por filtros de métrica, não poderá atribuir valores padrões para essas métricas.

> **Note**
>
> As dimensões são compatíveis apenas com filtros de métrica JSON ou delimitados por espaço.

13. Escolha Criar filtro de métrica. Você pode encontrar o filtro de métrica que criou no painel de navegação. Escolha Logs e depois escolha Log groups (Grupo de logs). Escolha o nome do grupo de logs para o qual você criou o filtro de métrica e, em seguida, selecione a guia Metric filters (Filtros de métrica).

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 registros MyApp /access.log para criar a métrica EventCount no CloudWatch namespace. MyNamespace O filtro é configurado para fazer a correspondência de qualquer conteúdo de eventos de log e incrementar a métrica por "1".

Para criar um filtro métrico usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. Escolha o nome de um grupo de logs.
4. Escolha a `Actions` Criar filtro de métrica.
5. Deixe Padrão de filtro e Selecionar dados de log para testar 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 métrico usando o 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 os pontos de dados publicados na métrica MyAppAccessEventCount.

Para publicar dados do evento usando o 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 registros MyApp/message.log. CloudWatch O Logs publica um ponto de dados ErrorCount na métrica CloudWatch personalizada no namespace MyApp/message.log com o valor “1" para cada evento que contém Error. Se não houver um evento com a palavra Erro, será publicado um valor de 0. Ao representar graficamente esses dados no CloudWatch console, certifique-se de usar a estatística da soma.

Depois de criar um filtro de métrica, você pode ver a métrica no CloudWatch console. 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 métrico usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha 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 Filtrar padrão fazem distinção de maiúsculas e minúsculas.

6. (Opcional) Para testar seu padrão de filtro, em Test Pattern (Testar padrão), insira um ou mais eventos de log a serem usados para testar o padrão. Cada evento de log deve estar dentro de uma linha, porque as quebras de linha são usadas para separar eventos de log na caixa Log event messages (Mensagens do evento de log).
7. Escolha Próximo e, na página Atribuir métrica, em Nome do filtro, digite **`MyAppErrorCount`**.
8. Em Metric Details, em Metric Namespace, digite. MyNameSpace
9. Para Metric Name (Nome da métrica), digite ErrorCount.
10. Confirme se o 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 métrico usando o 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 o AWS CLI

Em um prompt de comando, execute o seguinte comando da . 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 seus servidores Apache retornam uma resposta HTTP 404, que é o código de resposta para a 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
- Nome de usuário
- Timestamp
- 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 métrico usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. Escolha a `Actions` Criar filtro de métrica.
4. Em Padrão de filtro, digite **`[IP, UserInfo, User, Timestamp, RequestInfo, StatusCode=404, Bytes]`**.
5. (Opcional) Para testar seu padrão de filtro, em Test Pattern (Testar padrão), insira um ou mais eventos de log a serem usados para testar o padrão. Cada evento de log deve estar dentro de uma linha, porque as quebras de linha são usadas para separar eventos de log na caixa Log event messages (Mensagens do evento de log).
6. Escolha Próximo e, para 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 métrico usando o 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 após colocar esses exemplos de eventos de log, você pode recuperar a métrica nomeada no CloudWatch console 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.

Para criar um filtro métrico usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. Escolha o nome do grupo de logs para o servidor Apache.
4. Escolha a `Actions` Criar filtro de métrica.
5. Em Filter pattern (Padrão de filtro), insira **`[ip, id, user, timestamp, request, status_code=4*, size]`**.
6. (Opcional) Para testar seu padrão de filtro, em Test Pattern (Testar padrão), insira um ou mais eventos de log a serem usados para testar o padrão. Cada evento de log deve estar dentro de uma linha, porque as quebras de linha são usadas para separar eventos de log na caixa Log event messages (Mensagens do evento de log).
7. Escolha Next (Próximo) e, em Filter name (Nome do filtro), digite **`HTTP4xxErrors`**.
8. Em Metric details (Detalhes da métrica), em Metric namespace (Namespace da métrica), insira **`MyNameSpace`**.
9. Em Metric name (Nome da métrica), insira HTTP4xxErrors.
10. Em Metric value (Valor da métrica), digite 1. Isso especifica que a contagem é aumentada em 1 para cada log que contém um erro 4xx.
11. Em Default value (Valor padrão), insira 0 e escolha Next.(Próximo).
12. Escolha Criar filtro de métrica.

Para criar um filtro métrico usando o 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 e atribuir dimensões

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.

O exemplo também mostra como atribuir dimensões à métrica que você está criando.

Para criar um filtro métrico usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. Escolha o nome do grupo de logs para o servidor Apache.
4. Escolha a `Actions` Criar filtro de métrica.
5. Em Filter pattern (Padrão de filtro), insira **`[ip, id, user, timestamp, request, status_code, size]`**.
6. (Opcional) Para testar seu padrão de filtro, em Test Pattern (Testar padrão), insira um ou mais eventos de log a serem usados para testar o padrão. Cada evento de log deve estar dentro de uma linha, porque as quebras de linha são usadas para separar eventos de log na caixa Log event messages (Mensagens do evento de log).
7. Escolha Next (Próximo) e, em Filter name (Nome do filtro), digite **`size`**.

8. Em Metric details (Detalhes da métrica), em Metric namespace (Namespace da métrica), insira **`MyNameSpace`**. Como se trata de um novo namespace, verifique se a opção Create new (Criar novo) está selecionada.
9. Em Metric name (Nome da métrica), insira **`BytesTransferred`**
10. Em Metric Value (Valor de métrica), insira **`$size`**.
11. Em Unit (Unidade), selecione Bytes.
12. Para Dimension Name, digite **`IP`**.
13. Em Dimension Value (Valor da dimensão), digite **`$ip`** e escolha Next (Próximo).
14. Escolha Criar filtro de métrica.

Para criar esse filtro métrico usando o 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'
```

```
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',unit=Bytes,dimensio
$ip}}'
```

> **Note**
>
> Nesse comando, use esse formato para especificar várias dimensões.
>
> ```
> aws logs put-metric-filter \
> --log-group-name my-log-group-name \
> --filter-name my-filter-name \
> --filter-pattern 'my-filter-pattern' \
> --metric-transformations  \
> ```

```
metricName=my-metric-name,metricNamespace=my-metric-namespace,metricValue=my-token,unit=unit,dimensions='{dimension1=$dim,dimension2=$dim2,dim3=$dim3}'
```

Você pode usar os dados a seguir em put-log-event chamadas 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 métricos usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. No painel de conteúdo, na lista de grupos de log, na coluna Filtros de métrica, escolha o número de filtros.

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

Para listar filtros métricos usando o 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]"
        }
    ]
}
```

Excluir um filtro de métrica

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

Para excluir um filtro métrico usando o CloudWatch console

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. No painel de conteúdo, na lista de grupos de log, na coluna Metric Filter (Filtros de métrica), escolha o número de filtros de métrica para o grupo de logs.
4. Em Metric Filters (Filtros de métrica), selecione a caixa de seleção à direita do nome do filtro que deseja excluir. Em seguida, selecione Excluir.
5. Quando a confirmação for solicitada, escolha Excluir.

Para excluir um filtro métrico usando o 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

Você pode usar assinaturas para ter acesso a um feed em tempo real de eventos de log do CloudWatch Logs e entregá-lo a outros serviços, como um stream do Amazon Kinesis, um stream do Amazon Data Firehose, AWS Lambda ou para processamento, análise ou carregamento personalizados em outros sistemas. Quando os eventos de log são enviados ao serviço de recebimento, eles são codificados em base64 e compactados no formato gzip.

Para começar a assinar eventos de logs, crie o recurso de recebimento, como um fluxo do Kinesis Data Streams, ao qual os eventos serão entregues. Um filtro de assinatura define o padrão de filtro a ser usado para filtrar quais eventos de log são entregues ao seu AWS recurso, bem como informações sobre para onde enviar eventos de log correspondentes.

Você pode criar assinaturas no nível da conta e no nível do grupo de registros. Cada conta pode ter um filtro de assinatura no nível da conta. Cada grupo de logs pode ter até dois filtros de assinatura associados a ele.

> **Note**
>
> Se o serviço de destino retornar um erro que pode ser repetido, como uma exceção de limitação ou uma exceção de serviço que pode ser repetida (HTTP 5xx, por exemplo), o CloudWatch Logs continuará tentando entregar novamente por até 24 horas. CloudWatch Os registros não tentarão ser entregues novamente se o erro for um erro que não possa ser repetido, como ou. AccessDeniedException ResourceNotFoundException Nesses casos, o filtro de assinatura é desativado por até 10 minutos e, em seguida, o CloudWatch Logs tenta enviar os registros novamente para o destino. Durante esse período desativado, os registros são ignorados.

CloudWatch Os registros também produzem CloudWatch métricas sobre o encaminhamento de eventos de registro para assinaturas. Para ter mais informações, consulte Monitoramento com CloudWatch métricas.

Você também pode usar uma assinatura do CloudWatch Logs para transmitir dados de log quase em tempo real para um cluster do Amazon OpenSearch Service. Para obter mais informações, consulte Dados de CloudWatch registros de streaming para o Amazon OpenSearch Service.

As assinaturas são suportadas somente para grupos de registros na classe de registros Standard. Para obter mais informações sobre classes de log, consulteClasses de log.

> **Note**
>
> Os filtros de assinatura podem registrar eventos em lote para otimizar a transmissão e reduzir a quantidade de chamadas feitas para o destino. O agrupamento em lotes não é garantido, mas é usado quando possível.

Conteúdo

- Conceitos
- Registre filtros de assinatura em nível de grupo
- Filtros de assinatura em nível de conta
- Assinaturas entre contas e regiões
- Prevenção de 'confused deputy'
- Prevenção de recursão de registros

Conceitos

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

padrão de filtro

Uma descrição simbólica de como CloudWatch os registros devem interpretar os dados em cada evento de registro, junto com expressões de filtragem que restringem o que é entregue ao AWS recurso de destino. Para obter mais informações sobre a sintaxe de padrões de filtros, consulte Sintaxe de padrões de filtros para filtros de métricas, filtros de assinatura, filtros de eventos de log e Live Tail..

arn de destino

O Amazon Resource Name (ARN) do stream do Kinesis Data Streams, do stream Firehose ou da função Lambda que você deseja usar como destino do feed de assinatura.

arn de função

Uma função do IAM que concede aos CloudWatch Logs as permissões necessárias para colocar dados no destino escolhido. Essa função não é necessária para destinos do Lambda porque

CloudWatch os registros podem obter as permissões necessárias nas configurações de controle de acesso na própria função do Lambda.

distribuição

O método usado para distribuir dados de log ao destino, quando o destino é um fluxo do Amazon Kinesis Data Streams. 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.

Para assinaturas em nível de grupo de registros, o seguinte elemento-chave também está incluído:

nome do grupo de logs

O grupo de logs ao qual associar o filtro de assinatura. Todos os eventos de log carregados para esse grupo de logs estariam sujeitos ao filtro de assinatura, e aqueles que correspondem ao filtro são entregues ao serviço de destino que está recebendo os eventos de log correspondentes.

Para assinaturas em nível de conta, o seguinte elemento-chave também está incluído:

critérios de seleção

Os critérios usados para selecionar quais grupos de registros têm o filtro de assinatura em nível de conta aplicado. Se você não especificar isso, o filtro de assinatura no nível da conta será aplicado a todos os grupos de registros na conta. Esse campo é usado para evitar loops de log infinitos. Para obter mais informações sobre o problema do loop de log infinito, consultePrevenção de recursão de registros.

Os critérios de seleção têm um limite de tamanho de 25 KB.

Registre filtros de assinatura em nível de grupo

Você pode usar um filtro de assinatura com Kinesis Data Streams, Lambda ou Firehose. Os logs enviados a um serviço de recebimento por meio de um filtro de assinatura são codificados em base64 e compactados no formato gzip.

É possível pesquisar seus dados de log usando a Sintaxe de padrões de filtros.

Exemplos

- Exemplo 1: filtros de assinatura com o Kinesis Data Streams

- Exemplo 2: filtros de assinatura com AWS Lambda
- Exemplo 3: filtros de assinatura com o Amazon Data Firehose

Exemplo 1: filtros de assinatura com o Kinesis Data Streams

O exemplo a seguir associa um filtro de assinatura a um grupo de registros contendo AWS CloudTrail eventos. O filtro de assinatura entrega todas as atividades registradas feitas pelas AWS credenciais “Root” a um stream no Kinesis Data Streams chamado "". RootAccess Para obter mais informações sobre como enviar AWS CloudTrail eventos para CloudWatch registros, consulte Enviar CloudTrail eventos para CloudWatch registros no Guia do AWS CloudTrail usuário.

> **Note**
>
> Antes de criar o fluxo do , calcule o volume de dados de log que será gerado. Certifique-se de criar um fluxo do 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, consulte Cotas e limites.
> São feitas novas tentativas de entregáveis com controle de utilização por até 24 horas. Após 24 horas, os entregáveis com falha são descartados.
> Para reduzir o risco de controle de utilização, você pode seguir as seguintes etapas:
>
> - Especifique `random` para `distribution` quando você criar o filtro de assinatura com PutSubscriptionFilterou put-subscription-filter. Por padrão, a distribuição do filtro de fluxo é por fluxo de log e isso pode causar limitação.
> - Monitore seu stream usando CloudWatch métricas. Isso ajuda você a identificar qualquer controle de utilização e ajustar sua configuração adequadamente. Por exemplo, a `DeliveryThrottling` métrica pode ser usada para rastrear o número de eventos de registro para os quais o CloudWatch Logs foi limitado ao encaminhar dados para o destino da assinatura. Para obter mais informações sobre o monitoramento, consulte Monitoramento com CloudWatch métricas.
> - Use o modo de capacidade sob demanda para o fluxo no Kinesis Data Streams. O modo sob demanda acomoda instantaneamente o crescimento e a redução das workloads. Para obter mais informações sobre o modo de capacidade sob demanda, consulte Modo sob demanda.

- Restrinja seu padrão de filtro de CloudWatch assinatura para corresponder à capacidade do seu stream no Kinesis Data Streams. Se você estiver enviando muitos dados para o fluxo, talvez seja necessário reduzir o tamanho do filtro ou ajustar os critérios dele.

Para criar um filtro de assinatura para o Kinesis Data Streams

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

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

2. Aguarde até que o stream do se torne ativo (isso pode levar um ou dois minutos). Você pode usar o seguinte comando describe-stream do Kinesis Data Streams para verificar o. StreamDescription StreamStatuspropriedade. Além disso, observe o valor StreamDescription.streamArn, pois você precisará dele em uma etapa posterior:

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

A seguir está 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 permissão aos CloudWatch Logs para colocar dados em seu stream. Primeiro, você precisará criar uma política de confiança em um arquivo (por exemplo, `~/TrustPolicyForCWL-Kinesis.json`). Use um editor de texto para criar esta política. Não use o console do IAM para criá-la.

   Esta política inclui uma chave de contexto de condição global `aws:SourceArn` para ajudar a evitar o problema de segurança 'confused deputy'. Para ter mais informações, consulte Prevenção de 'confused deputy'.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole",
       "Condition": {
           "StringLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
        }
      }
   }
   ```

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-Kinesis.json
   ```

   A seguir, veja um exemplo de saída.

   ```
   {
       "Role": {
           "AssumeRolePolicyDocument": {
               "Statement": {
                   "Action": "sts:AssumeRole",
                   "Effect": "Allow",
                   "Principal": {
                       "Service": "logs.amazonaws.com"
                   },
                   "Condition": {
                       "StringLike": {
                           "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
   ```

```
                    }
                }
            }
        },
        "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 na sua conta. Primeiro, você criará uma política de permissões em um arquivo (por exemplo, `~/PermissionsForCWL-Kinesis.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 à função usando o seguinte put-role-policycomando:

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

7. Depois que o stream estiver no estado Ativo e você tiver criado a função do IAM, você poderá criar o filtro de assinatura de CloudWatch registros. 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 :

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail/logs" \
    --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 encaminha todos os eventos de registro recebidos que correspondem ao padrão do filtro para o seu stream. Você pode verificar se isso está acontecendo capturando um iterador de fragmentos do Kinesis Data Streams e usando o comando get-records do Kinesis Data Streams para buscar alguns registros do Kinesis Data Streams:

```
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 pode ser necessário fazer esta chamada algumas vezes para que o Kinesis Data Streams comece a retornar dados.

É necessário esperar para ver uma resposta com um conjunto de registros. O atributo Dados em um registro do Kinesis Data Streams é codificado em base64 e compactado no 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 base64 decodificados e descompactados têm o formato JSON com a seguinte estrutura:

```
{
    "owner": "111111111111",
```

```
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":
\"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 AWS conta dos dados de registro 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 Data Streams com o tipo “CONTROL_MESSAGE”, principalmente para verificar se o destino está 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 assinatura com AWS Lambda

Neste exemplo, você criará um filtro de assinatura de CloudWatch registros que envia dados de registro para sua AWS Lambda função.

> **Note**
>
> Antes de criar a função 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 limites do Lambda, consulte Limites do AWS Lambda.

Para criar um filtro de assinatura para o Lambda

1. Crie a AWS Lambda função.

   Certifique-se de ter configurado a função de execução do Lambda. Para obter mais informações, consulte Etapa 2.2: Criar uma função do IAM (função de execução) no Guia do desenvolvedor do AWS Lambda .

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());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Compacte o arquivo helloWorld.js e salve-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 à 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.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 a saída de sua função Lambda, navegue para a função 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 base64 decodificados 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 AWS conta dos dados de registro 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, CloudWatch os Logs podem emitir registros Lambda com o tipo “CONTROL_MESSAGE”, principalmente para verificar se o destino está 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 assinatura com o Amazon Data Firehose

Neste exemplo, você criará uma assinatura do CloudWatch Logs que enviará todos os eventos de log recebidos que correspondam aos seus filtros definidos para o seu stream de entrega do Amazon Data Firehose. Os dados enviados do CloudWatch Logs para o Amazon Data Firehose já estão compactados com a compactação gzip de nível 6, portanto, você não precisa usar a compactação no seu stream de distribuição do Firehose. Em seguida, você pode usar o recurso de descompressão no

Firehose para descompactar automaticamente os registros. Para obter mais informações, consulte Gravando no Kinesis Data CloudWatch Firehose usando registros.

> **Note**
>
> Antes de criar o stream do Firehose, calcule o volume de dados de registro que serão gerados. Certifique-se de criar um stream do Firehose que possa lidar com 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 stream do Firehose, consulte Amazon Data Firehose Data Limits.

Para criar um filtro de assinatura para 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 Region (Região) pela região que você deseja usar:

   ```
   aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
    LocationConstraint=region
   ```

   A seguir está um exemplo de saída:

   ```
   {
       "Location": "/my-bucket"
   }
   ```

2. Crie a função do IAM que concede permissão ao Amazon Data Firehose para colocar dados em seu bucket do Amazon S3.

   Para obter mais informações, consulte Controlando o acesso com o Amazon Data Firehose no Guia do desenvolvedor do Amazon Data Firehose.

   Primeiro, use um editor de texto para criar uma política de confiança em um arquivo ~/TrustPolicyForFirehose.json, da seguinte maneira:

   ```
   {
   ```

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

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 Firehose pode fazer na 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 à função usando o seguinte put-role-policy comando:

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-
Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Crie um stream de entrega de destino do Firehose da seguinte forma, substituindo os valores de espaço reservado para roLearn e bucketArn pelos ARNs de função e bucket que você 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 Firehose usa automaticamente um prefixo no formato de hora UTC YYYYY/MM/DD/HH para objetos Amazon S3 entregues. 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 describe-delivery-streamcomando Firehose para verificar o. DeliveryStreamDescription DeliveryStreamStatuspropriedade. Além disso, observe DeliveryStreamDescriptiono. DeliveryStreamValor do ARN, conforme necessário em uma 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 concede permissão aos CloudWatch Logs para colocar dados em seu stream de entrega do Firehose. Primeiro, use um editor de texto para criar uma política de confiança em um arquivo `~/TrustPolicyForCWL.json`:

   Esta política inclui uma chave de contexto de condição global `aws:SourceArn` para ajudar a evitar o problema de segurança 'confused deputy'. Para ter mais informações, consulte Prevenção de 'confused deputy'.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
         "StringLike": {
             "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
         }
```

```
      }
   }
}
```

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.amazonaws.com"
                },
                "Condition": {
                     "StringLike": {
                         "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                     }
                 }
            }
        },
        "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 na 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:PutRecord"],
        "Resource":[
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"]
      }
    ]
}
```

11. Associe a política de permissões à função usando o put-role-policy comando:

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

12. Depois que o stream de entrega do Amazon Data Firehose estiver 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 do grupo de log escolhido para seu stream de entrega do Amazon 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 registro recebidos que correspondam ao padrão do filtro para o seu stream de entrega do Amazon Data Firehose. Seus dados começarão a aparecer no Amazon S3 com base no intervalo de tempo definido no stream de entrega do Amazon 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
```

Filtros de assinatura em nível de conta

> **Important**
>
> Existe o risco de causar um loop recursivo infinito com filtros de assinatura que pode levar a um grande aumento na cobrança por ingestão se não for resolvido. Para reduzir esse risco, recomendamos que você use critérios de seleção nos filtros de assinatura em nível de conta para excluir grupos de registros que ingerem dados de registro de recursos que fazem parte do fluxo de trabalho de entrega de assinaturas. Para obter mais informações sobre esse problema e determinar quais grupos de registros excluir, consulte Prevenção de recursão de registros.

Você pode definir uma política de assinatura em nível de conta que inclua um subconjunto de grupos de registros na conta. A política de assinatura da conta pode funcionar com Kinesis Data Streams, Lambda ou Firehose. Os registros enviados para um serviço de recebimento por meio de uma política de assinatura em nível de conta são codificados em base64 e compactados com o formato gzip.

> **Note**
>
> Para ver uma lista de todas as políticas de filtro de assinatura em sua conta, use o `describe-account-policies` comando com um valor de `SUBSCRIPTION_FILTER_POLICY` para o `--policy-type` parâmetro. Para obter mais informações, consulte describe-account-policies¶.

Exemplos

- Exemplo 1: filtros de assinatura com o Kinesis Data Streams
- Exemplo 2: filtros de assinatura com AWS Lambda
- Exemplo 3: filtros de assinatura com o Amazon Data Firehose

Exemplo 1: filtros de assinatura com o Kinesis Data Streams

Antes de criar um stream de dados do Kinesis Data Streams para usar com uma política de assinatura em nível de conta, calcule o volume de dados de log que serão gerados. Certifique-se

de criar um fluxo do com estilhaços suficientes para suportar esse volume. Se um stream não tiver fragmentos suficientes, ele será estrangulado. Para obter mais informações sobre limites de volume de stream, consulte Cotas e limites na documentação do Kinesis Data Streams.

> **Warning**
>
> Como os eventos de log de vários grupos de log são encaminhados para o destino, existe o risco de limitação. São feitas novas tentativas de entregáveis com controle de utilização por até 24 horas. Após 24 horas, os entregáveis com falha são descartados.
> Para reduzir o risco de controle de utilização, você pode seguir as seguintes etapas:
>
> - Monitore seu stream do Kinesis Data Streams CloudWatch com métricas. Isso ajuda você a identificar a limitação e ajustar sua configuração adequadamente. Por exemplo, a `DeliveryThrottling` métrica rastreia o número de eventos de registro para CloudWatch os quais o Logs foi limitado ao encaminhar dados para o destino da assinatura. Para ter mais informações, consulte Monitoramento com CloudWatch métricas.
> - Use o modo de capacidade sob demanda para o fluxo no Kinesis Data Streams. O modo sob demanda acomoda instantaneamente o crescimento e a redução das workloads. Para obter mais informações, consulte Modo sob demanda.
> - Restrinja seu padrão de filtro de assinatura do CloudWatch Logs para corresponder à capacidade do seu stream no Kinesis Data Streams. Se você estiver enviando muitos dados para o fluxo, talvez seja necessário reduzir o tamanho do filtro ou ajustar os critérios dele.

O exemplo a seguir usa uma política de assinatura em nível de conta para encaminhar todos os eventos de log para um stream no Kinesis Data Streams. O padrão de filtro combina todos os eventos de registro com o texto `Test` e os encaminha para o stream no Kinesis Data Streams.

Para criar uma política de assinatura em nível de conta para o Kinesis Data Streams

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

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

2. Aguarde alguns minutos até que o stream fique ativo. Você pode verificar se o fluxo está ativo usando o comando describe-stream para verificar o. StreamDescription StreamStatuspropriedade.

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

A seguir está um exemplo de saída:

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "TestStream",
        "StreamARN": "arn:aws:kinesis:region:123456789012:stream/TestStream",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "EXAMPLE84634633746074317682114555",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "EXAMPLE688818456679503831981458784591352702181572610"
                }
            }
        ]
    }
}
```

3. Crie a função do IAM que concederá permissão aos CloudWatch Logs para colocar dados em seu stream. Primeiro, você precisará criar uma política de confiança em um arquivo (por exemplo, `~/TrustPolicyForCWL-Kinesis.json`). Use um editor de texto para criar esta política.

   Esta política inclui uma chave de contexto de condição global `aws:SourceArn` para ajudar a evitar o problema de segurança 'confused deputy'. Para ter mais informações, consulte Prevenção de 'confused deputy'.

   ```
   {
     "Statement": {
       "Effect": "Allow",
       "Principal": { "Service": "logs.amazonaws.com" },
       "Action": "sts:AssumeRole",
       "Condition": {
           "StringLike": { "aws:SourceArn": "arn:aws:logs:region:123456789012:*" }
   ```

```
      }
    }
}
```

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-Kinesis.json
```

A seguir, veja um exemplo de saída.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": { "arn:aws:logs:region:123456789012:*" }
                    }
                }
            }
        },
        "RoleId": "EXAMPLE450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "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 na sua conta. Primeiro, você criará uma política de permissões em um arquivo (por exemplo, `~/PermissionsForCWL-Kinesis.json`). Use um editor de texto para criar esta política. Não use o console do IAM para criá-lo.

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

6. Associe a política de permissões à função usando o seguinte put-role-policycomando:

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

7. Depois que o stream estiver no estado Ativo e você tiver criado a função do IAM, você poderá criar a política de filtro de assinatura de CloudWatch registros. A política inicia imediatamente o fluxo de dados de registro em tempo real para seu stream. Neste exemplo, todos os eventos de log que contêm a string ERROR são transmitidos, exceto aqueles nos grupos de log chamados `LogGroupToExclude1` e. `LogGroupToExclude2`

```
aws logs put-account-policy \
    --policy-name "ExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisRole", "DestinationArn":"arn:aws:kinesis:region:123456789012:stream/
TestStream", "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

8. Depois de configurar o filtro de assinatura, o CloudWatch Logs encaminha todos os eventos de registro recebidos que correspondam ao padrão do filtro e aos critérios de seleção para o seu stream.

   O `selection-criteria` campo é opcional, mas é importante para excluir grupos de registros que podem causar uma recursão infinita de registros de um filtro de assinatura. Para obter mais informações sobre esse problema e determinar quais grupos de registros excluir, consultePrevenção de recursão de registros. Atualmente, NOT IN é o único operador compatível com `selection-criteria` o.

Você pode verificar o fluxo de eventos de log usando um iterador de fragmentos do Kinesis Data Streams e usando o `get-records` comando Kinesis Data Streams para buscar alguns registros do Kinesis Data Streams:

```
aws kinesis get-shard-iterator --stream-name TestStream --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"
```

Talvez você precise usar esse comando algumas vezes antes que o Kinesis Data Streams comece a retornar dados.

É necessário esperar para ver uma resposta com um conjunto de registros. O atributo Dados em um registro do Kinesis Data Streams é codificado em base64 e compactado no 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 base64 decodificados e descompactados têm o formato JSON com a seguinte estrutura:

```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
```

```
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicy"
    ],
    "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\"}"
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```

Os principais elementos da estrutura de dados são os seguintes:

messageType

As mensagens de dados usarão o tipo "DATA_MESSAGE". Às vezes, o CloudWatch Logs pode emitir registros do Kinesis Data Streams com o tipo “CONTROL_MESSAGE”, principalmente para verificar se o destino está acessível.

owner

O ID da AWS conta dos dados de registro 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.

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.

Nível de política

O nível em que a política foi aplicada. “ACCOUNT_LEVEL_POLICY” serve `policyLevel` para uma política de filtro de assinatura em nível de conta.

Exemplo 2: filtros de assinatura com AWS Lambda

Neste exemplo, você criará uma política de filtro de assinatura no nível da conta do CloudWatch Logs que envia dados de registro para sua AWS Lambda função.

> **Warning**
>
> Antes de criar a função 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 conseguir lidar com o volume, o fluxo de log será acelerado. Como os eventos de registro de todos os grupos de registro ou de um subconjunto dos grupos de registro da conta são encaminhados para o destino, existe o risco de limitação. Para obter mais informações sobre limites do Lambda, consulte Limites do AWS Lambda.

Para criar uma política de filtro de assinatura em nível de conta para o Lambda

1. Crie a AWS Lambda função.

   Certifique-se de ter configurado a função de execução do Lambda. Para obter mais informações, consulte Etapa 2.2: Criar uma função do IAM (função de execução) no Guia do desenvolvedor do AWS Lambda .
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());
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Compacte o arquivo helloWorld.js e salve-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 nodejs18.x
```

5. Conceda à CloudWatch Logs a permissão para executar sua função. Use o comando a seguir, substituindo a conta de espaço reservado pela sua própria conta.

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789012:log-group:*" \
    --source-account "123456789012"
```

6. Crie uma política de filtro de assinatura no nível da conta usando o comando a seguir, substituindo a conta de espaço reservado pela sua própria conta. Neste exemplo, todos os eventos de log que contêm a string `ERROR` são transmitidos, exceto aqueles nos grupos de log chamados `LogGroupToExclude1` e. `LogGroupToExclude2`

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyLambda" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:lambda:region:123456789012:function:helloWorld",
 "FilterPattern": "Test", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Depois de configurar o filtro de assinatura, o CloudWatch Logs encaminha todos os eventos de registro recebidos que correspondam ao padrão do filtro e aos critérios de seleção para o seu stream.

O `selection-criteria` campo é opcional, mas é importante para excluir grupos de registros que podem causar uma recursão infinita de registros de um filtro de assinatura. Para obter mais informações sobre esse problema e determinar quais grupos de registros excluir, consultePrevenção de recursão de registros. Atualmente, NOT IN é o único operador compatível com `selection-criteria` o.

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 a saída de sua função Lambda, navegue para a função Lambda na qual você verá a saída em /aws/lambda/helloworld:

   ```
   aws logs put-log-events --log-group-name Example1 --log-stream-name logStream1 --
   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 base64 decodificados e descompactados têm o formato JSON com a seguinte estrutura:


```
{
    "messageType": "DATA_MESSAGE",
    "owner": "123456789012",
    "logGroup": "Example1",
    "logStream": "logStream1",
    "subscriptionFilters": [
        "ExamplePolicyLambda"
    ],
    "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\"}"
        }
    ],
    "policyLevel": "ACCOUNT_LEVEL_POLICY"
}
```


> **Note**
>
> O filtro de assinatura no nível da conta não será aplicado ao grupo de registros da função Lambda de destino. Isso evita uma recursão infinita de registros que pode levar a um aumento no faturamento por ingestão. Para obter mais informações sobre esse problema, consulte Prevenção de recursão de registros.

Os principais elementos da estrutura de dados são os seguintes:

messageType

As mensagens de dados usarão o tipo "DATA_MESSAGE". Às vezes, o CloudWatch Logs pode emitir registros do Kinesis Data Streams com o tipo “CONTROL_MESSAGE”, principalmente para verificar se o destino está acessível.

owner

O ID da AWS conta dos dados de registro 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.

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.

Nível de política

O nível em que a política foi aplicada. “ACCOUNT_LEVEL_POLICY” serve `policyLevel` para uma política de filtro de assinatura em nível de conta.

Exemplo 3: filtros de assinatura com o Amazon Data Firehose

Neste exemplo, você criará uma política de filtro de assinatura em nível de conta do CloudWatch Logs que envia eventos de log recebidos que correspondam aos filtros definidos para o seu stream de entrega do Amazon Data Firehose. Os dados enviados do CloudWatch Logs para o Amazon Data Firehose já estão compactados com a compactação gzip de nível 6, portanto, você não precisa usar a compactação no seu stream de distribuição do Firehose. Em seguida, você pode usar o recurso

de descompressão no Firehose para descompactar automaticamente os registros. Para obter mais informações, consulte Gravando no Kinesis Data CloudWatch Firehose usando registros.

> **Warning**
>
> Antes de criar o stream do Firehose, calcule o volume de dados de registro que serão gerados. Certifique-se de criar um stream do Firehose que possa lidar com 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 stream do Firehose, consulte Amazon Data Firehose Data Limits.

Para criar um filtro de assinatura para 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 Region (Região) pela região que você deseja usar:

   ```
   aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
    LocationConstraint=region
   ```

   A seguir está um exemplo de saída:

   ```
   {
       "Location": "/my-bucket"
   }
   ```

2. Crie a função do IAM que concede permissão ao Amazon Data Firehose para colocar dados em seu bucket do Amazon S3.

   Para obter mais informações, consulte Controlando o acesso com o Amazon Data Firehose no Guia do desenvolvedor do Amazon Data Firehose.

   Primeiro, use um editor de texto para criar uma política de confiança em um arquivo `~/TrustPolicyForFirehose.json`, da seguinte maneira:

   ```
   {
   ```

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

3. Use o comando create-role para criar a função do IAM especificando o arquivo de política de confiança. Anote 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": "EXAMPLE50GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Crie uma política de permissões para definir quais ações o Firehose pode fazer na 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 à função usando o seguinte put-role-policy comando:

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-
Policy-For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Crie um stream de entrega de destino do Firehose da seguinte forma, substituindo os valores de espaço reservado para roLearn e bucketArn pelos ARNs de função e bucket que você 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"}'
```

   O NFirehose usa automaticamente um prefixo no formato de hora UTC YYYYY/MM/DD/HH para objetos Amazon S3 entregues. 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 alguns minutos até que o stream fique ativo. Você pode usar o describe-delivery-streamcomando Firehose para verificar o. DeliveryStreamDescription DeliveryStreamStatuspropriedade. Além disso, observe DeliveryStreamDescriptiono. DeliveryStreamValor do ARN, conforme necessário em uma 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 concede permissão aos CloudWatch Logs para colocar dados em seu stream de entrega do Firehose. Primeiro, use um editor de texto para criar uma política de confiança em um arquivo `~/TrustPolicyForCWL.json`:

   Esta política inclui uma chave de contexto de condição global `aws:SourceArn` para ajudar a evitar o problema de segurança 'confused deputy'. Para ter mais informações, consulte Prevenção de 'confused deputy'.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": {
         "StringLike": {
             "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
         }
```

```
      }
   }
}
```

9. Use o comando create-role para criar a função do IAM especificando o arquivo de política de confiança. Anote 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.amazonaws.com"
                },
                "Condition": {
                     "StringLike": {
                         "aws:SourceArn": "arn:aws:logs:region:123456789012:*"
                     }
                 }
            }
        },
        "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 na 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:PutRecord"],
        "Resource":[
            "arn:aws:firehose:region:account-id:deliverystream/delivery-stream-
name"]
      }
    ]
}
```

11. Associe a política de permissões à função usando o put-role-policy comando:

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

12. Depois que o stream de entrega do Amazon Data Firehose estiver ativo e você tiver criado a função do IAM, você poderá criar a política de filtro de assinatura em nível de conta do CloudWatch Logs. A política inicia imediatamente o fluxo de dados de log em tempo real do grupo de log escolhido para o seu stream de entrega do Amazon Data Firehose:

```
aws logs put-account-policy \
    --policy-name "ExamplePolicyFirehose" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"RoleArn":"arn:aws:iam::123456789012:role/
CWLtoKinesisFirehoseRole", "DestinationArn":"arn:aws:firehose:us-
east-1:123456789012:deliverystream/delivery-stream-name", "FilterPattern": "Test",
 "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

13. Depois de configurar o filtro de assinatura, o CloudWatch Logs encaminha os eventos de log recebidos que correspondem ao padrão do filtro para o seu stream de entrega do Amazon Data Firehose.

    O `selection-criteria` campo é opcional, mas é importante para excluir grupos de registros que podem causar uma recursão infinita de registros de um filtro de assinatura. Para obter mais informações sobre esse problema e determinar quais grupos de registros excluir, consulte Prevenção de recursão de registros. Atualmente, NOT IN é o único operador compatível com `selection-criteria` o.

Seus dados começarão a aparecer no Amazon S3 com base no intervalo de tempo definido no stream de entrega do Amazon 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": "2023-10-29T00:01:25.000Z",
            "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/2023/10/29/00/my-delivery-stream-2023-10-29-00-35-40-
EXAMPLE-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "EXAMPLE6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'my-bucket' --key 'firehose/2023/10/29/00/my-
delivery-stream-2023-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250'
 testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
```

```
    "LastModified": "Thu, 29 Oct 2023 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
```

Assinaturas entre contas e regiões

Você pode colaborar com um proprietário de uma AWS conta diferente e receber seus eventos de log em seus AWS recursos, como um stream do Amazon Kinesis ou do Amazon Data Firehose (isso é conhecido como compartilhamento de dados entre contas). Por exemplo, esses dados de eventos de log podem ser lidos de um stream centralizado do Kinesis Data Streams ou Firehose para realizar processamento e análise personalizados. 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 de dados de eventos em tempo real entre essas contas pode ser montado e entregue aos grupos de segurança de informações, que podem usar o Kinesis Data Streams para anexar os dados aos sistemas analíticos de segurança existentes.

> **Note**
>
> O grupo de registros e o destino devem estar na mesma AWS região. No entanto, o AWS recurso para o qual o destino aponta pode estar localizado em uma região diferente. Nos exemplos das seções a seguir, todos os recursos específicos da região são criados no Leste dos EUA (Norte da Virgínia).

Tópicos

- Compartilhamento de dados de log entre contas usando o Kinesis Data Streams
- Compartilhamento de dados de registro entre contas usando Firehose

- Assinaturas multiregionais em nível de conta usando o Kinesis Data Streams
- Assinaturas multiregionais em nível de conta usando Firehose

Compartilhamento de dados de log entre contas usando o Kinesis Data Streams

Ao criar uma assinatura entre contas, você pode especificar uma única conta ou organização para ser o remetente. Se você especificar uma organização, esse procedimento permitirá que todas as contas da organização enviem logs para a conta do receptor.

Para compartilhar dados de log entre contas, você precisa estabelecer um remetente e um destinatário dos dados de log:

- Remetente dos dados de registro — obtém as informações de destino do destinatário e informa ao CloudWatch Logs que ele está pronto para enviar seus eventos de registro para o destino especificado. Nos procedimentos do restante desta seção, o remetente dos dados de log é mostrado com um número de AWS conta fictício de 111111111111.

  Se você quiser que várias contas em uma organização enviem logs para uma conta de destinatário, você pode criar uma política que conceda a todas as contas da organização a permissão para enviar logs para a conta do destinatário. Você ainda precisa configurar filtros de assinatura diferentes para cada conta de remetente.
- Destinatário dos dados de log — configura um destino que encapsula um stream do Kinesis Data Streams e informa ao Logs que o CloudWatch destinatário deseja receber dados de log. O destinatário, então, compartilha as informações sobre esse destino com o remetente. Nos procedimentos do restante desta seção, o destinatário dos dados de registro é mostrado com um número de AWS conta fictício de 999999999999.

Para começar a receber eventos de registro de usuários de várias contas, o destinatário dos dados de registro primeiro cria um destino de CloudWatch registros. Cada destino consiste nos seguintes elementos-chave:

Nome do destino

O nome do destino que você deseja criar.

ARN de destino

O Amazon Resource Name (ARN) do AWS recurso que você deseja usar como destino do feed de assinatura.

ARN de função

Uma função AWS Identity and Access Management (IAM) que concede aos CloudWatch Logs as permissões necessárias para colocar dados no stream escolhido.

Política de acesso

Documento de políticas 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.

> **Note**
>
> O grupo de registros e o destino devem estar na mesma AWS região. No entanto, o recurso da AWS para o qual o destino aponta pode estar localizado em uma região diferente. Nos exemplos das seções a seguir, todos os recursos específicos da região são criados no Leste dos EUA (Virgínia do Norte).

Tópicos

- Como configurar uma nova assinatura entre contas
- Atualizar uma assinatura existente entre contas

Como configurar uma nova assinatura entre contas

Siga as etapas nestas seções para configurar uma nova assinatura de log entre contas.

Tópicos

- Etapa 1: criar um destino
- Etapa 2: (somente se estiver usando uma organização) Crie uma função do IAM.
- Etapa 3: adicionar/validar as permissões do IAM para o destino entre contas
- Etapa 4: criar um filtro de assinatura
- Validação do fluxo de eventos de logs

- Modificar a associação de destino no runtime

Etapa 1: criar um destino

> **Important**
>
> As etapas deste procedimento devem ser processadas na conta destinatária dos dados do log.

Neste exemplo, a conta do destinatário dos dados de registro tem uma ID de conta de 999999999999, enquanto a ID da AWS conta do remetente dos dados de registro é 111111111111. AWS

Este exemplo cria um destino usando um stream do Kinesis Data RecipientStream Streams chamado, e uma função CloudWatch que permite que o Logs grave dados nele.

Quando o destino é criado, o CloudWatch Logs envia uma mensagem de teste para o destino em nome da conta do destinatário. Quando o filtro de assinatura é ativado posteriormente, o CloudWatch Logs envia eventos de registro para o destino em nome da conta de origem.

Para criar um destino

1. Na conta do destinatário, crie um fluxo de destino no Kinesis Data Streams. Em um prompt de comando, digite:

   ```
   aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
   ```

2. Aguarde até que o fluxo do fique ativo. Você pode usar o comando aws kinesis describe-stream para verificar o. StreamDescription StreamStatuspropriedade. Além disso, anote o valor StreamDescription.streamArn porque você o passará para CloudWatch o Logs posteriormente:

   ```
   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":
 "49551135218688818456679503831981458784591352702 18EXAMPLE"
        }
      }
    ]
  }
}
```

Pode levar um ou dois minutos para o seu stream aparecer no estado ativo.

3. Crie a função do IAM que concede a CloudWatch Logs a permissão para colocar dados em seu stream. Primeiro, você precisará criar uma política de confiança em um arquivo TrustPolicyFor~/CWL.json. Use um editor de texto para criar esse arquivo de política, não use o console do IAM.

   Esta política inclui uma chave de contexto de condição global `aws:SourceArn` que especifica o `sourceAccountId` para evitar o problema de segurança confused deputy. Se você ainda não souber o ID da conta de origem na primeira chamada, recomendamos que você coloque o ARN de destino no campo ARN de origem. Nas chamadas subsequentes, você deve definir o ARN de origem como o ARN de origem real que você coletou da primeira chamada. Para obter mais informações, consulte Prevenção de 'confused deputy'.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "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 passado para Logs posteriormente: CloudWatch

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

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.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 realizar na sua conta. Primeiro, use um editor de texto para criar uma política de permissões em um arquivo PermissionsFor~/CWL.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 put-role-policy comando aws iam:

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

7. Depois que o stream estiver no estado ativo e você tiver criado a função do IAM, você poderá criar o destino dos CloudWatch registros.

   a. Esta etapa não associará uma política de acesso ao seu destino e é apenas a primeira etapa de duas que concluirá uma criação de destino. Anote o DestinationArnque é 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. Essa política deve especificar os registros: PutSubscriptionFilter ação e concede permissão à conta do remetente para acessar o destino.

A política concede permissão à AWS conta que envia os registros. Você pode especificar apenas essa conta na política ou, se a conta de remetente for parte de uma organização, a política poderá especificar o ID da organização. Dessa forma, você pode criar apenas uma política para permitir que várias contas em uma organização enviem logs para essa conta de destino.

Use um editor de texto para criar um arquivo chamado `~/AccessPolicy.json` com uma das seguintes declarações de política.

Este primeiro exemplo de política permite que todas as contas na organização que tenham um ID de `o-1234567890` enviem logs à conta do destinatário.

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : "logs:PutSubscriptionFilter",
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

Este próximo exemplo permite que apenas a conta remetente dos dados de log (111111111111) envie logs à conta destinatária dos dados de log.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
```

```
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

c. Anexe a política criada na etapa anterior ao destino.

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Essa política de acesso permite que os usuários da AWS Conta com ID 111111111111 liguem para o destino com o ARN arn:aws:logs: ***PutSubscriptionFilter****region:9999999999:destination:testDestination.* A tentativa de qualquer outro usuário de ligar PutSubscriptionFilter para esse destino será rejeitada.

Para validar os privilégios de um usuário com base em uma política de acesso, consulte Usar o validador de políticas no Manual do usuário do IAM.

Ao terminar, se estiver usando AWS Organizations suas permissões entre contas, siga as etapas emEtapa 2: (somente se estiver usando uma organização) Crie uma função do IAM.. Se você estiver concedendo permissões diretamente para a outra conta em vez de usar Organizations, você pode pular essa etapa e prosseguir para Etapa 4: criar um filtro de assinatura.

Etapa 2: (somente se estiver usando uma organização) Crie uma função do IAM.

Na seção anterior, se você criou o destino usando uma política de acesso que concede permissões à organização em que está a conta 111111111111, em vez de conceder permissões diretamente para a conta 111111111111, siga as etapas nesta seção. Caso contrário, pule para Etapa 4: criar um filtro de assinatura.

As etapas desta seção criam uma função do IAM, que CloudWatch pode assumir e validar se a conta do remetente tem permissão para criar um filtro de assinatura em relação ao destino do destinatário.

Siga as etapas nesta seção na conta do remetente. A função deve existir na conta do remetente e você especifica o ARN dessa função no filtro de assinatura. Neste exemplo, a conta de remetente é 111111111111.

Criar a função do IAM necessária para assinaturas de log entre contas usando o AWS Organizations

1. Crie a seguinte política de confiança em um arquivo `/TrustPolicyForCWLSubscriptionFilter.json`. Use um editor de texto para criar esse arquivo de política; não use o console do IAM.

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

2. Crie uma função do IAM que use essa política. Anote o valor `Arn` retornado pelo comando, pois você precisará dele posteriormente nesse procedimento. Neste exemplo, usamos `CWLtoSubscriptionFilterRole` como o nome da função que estamos criando.

```
aws iam create-role \
     --role-name CWLtoSubscriptionFilterRole \
     --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. Crie uma política de permissões para definir as ações que o CloudWatch Logs pode realizar na sua conta.

   a. Primeiro, use um editor de texto para criar a seguinte política de permissões em um arquivo chamado `~/PermissionsForCWLSubscriptionFilter.json`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
```

```
    ]
}
```

b. Insira o seguinte comando para associar a política de permissões que você acabou de criar à função criada na etapa 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

Quando terminar, você pode prosseguir para Etapa 4: criar um filtro de assinatura.

Etapa 3: adicionar/validar as permissões do IAM para o destino entre contas

De acordo com a lógica de avaliação de políticas AWS entre contas, para acessar qualquer recurso entre contas (como um stream do Kinesis ou Firehose usado como destino para um filtro de assinatura), você deve ter uma política baseada em identidade na conta de envio que forneça acesso explícito ao recurso de destino entre contas. Para obter mais informações sobre a lógica de avaliação de política, consulte Lógica de avaliação de política entre contas.

Você pode associar a política baseada em identidade ao perfil do IAM ou ao usuário do IAM que você está usando para criar o filtro de assinatura. Essa política deve estar presente na conta de envio. Se você estiver usando a função de administrador para criar o filtro de assinatura, poderá ignorar esta etapa e prosseguir para Etapa 4: criar um filtro de assinatura.

Para adicionar ou validar as permissões do IAM necessárias entre contas

1. Insira o comando a seguir para verificar qual perfil do IAM ou usuário do IAM está sendo usado para executar comandos de log da AWS .

```
aws sts get-caller-identity
```

Esse comando retorna uma saída semelhante à seguinte:

```
{
"UserId": "User ID",
"Account": "sending account id",
"Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
```

```
}
```

Anote o valor representado por *RoleName* ou *UserName*.

2. Faça login AWS Management Console na conta de envio e pesquise as políticas anexadas com a função do IAM ou o usuário do IAM retornado na saída do comando inserido na etapa 1.
3. Verifique se as políticas vinculadas a esse perfil ou usuário fornecem permissões explícitas para chamar `logs:PutSubscriptionFilter` no recurso de destino entre contas. O exemplo de políticas a seguir mostra as permissões recomendadas.

   A política a seguir fornece permissões para criar um filtro de assinatura em qualquer recurso de destino somente em uma única AWS conta, conta`123456789012`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on any resource in one specific
 account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

   A política a seguir fornece permissões para criar um filtro de assinatura somente em um recurso de destino específico chamado `sampleDestination` em uma única AWS conta, conta`123456789012`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in one
 specific account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
```

```
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:sampleDestination"
            ]
        }
    ]
}
```

Etapa 4: 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.

> **Note**
>
> Se você estiver concedendo permissões para o filtro de assinatura a uma organização inteira, precisará usar o ARN do perfil do IAM que criou em Etapa 2: (somente se estiver usando uma organização) Crie uma função do IAM..

No exemplo a seguir, um filtro de assinatura é criado em uma conta de envio. O filtro é associado a um grupo de registros contendo AWS CloudTrail eventos para que todas as atividades registradas feitas pelas AWS credenciais “Root” sejam entregues ao destino que você criou anteriormente. Esse destino encapsula um fluxo chamado ""RecipientStream.

O restante das etapas nas seções a seguir pressupõe que você seguiu as instruções em Enviar CloudTrail eventos para CloudWatch registros no Guia do AWS CloudTrail usuário e criou um grupo de registros que contém seus CloudTrail eventos. Essas etapas pressupõem que o nome desse grupo de logs é `CloudTrail/logs`.

Ao inserir o comando a seguir, certifique-se de estar conectado como usuário do IAM ou usando o perfil do IAM para o qual você adicionou a política, em Etapa 3: adicionar/validar as permissões do IAM para o destino entre contas.

```
aws logs put-subscription-filter \
```

```
    --log-group-name "CloudTrail/logs" \
    --filter-name "RecipientStream" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

O grupo de registros e o destino devem estar na mesma AWS região. No entanto, o destino pode apontar para um AWS recurso, como um stream do Kinesis Data Streams, localizado em uma região diferente.

Validação do fluxo de eventos de logs

Depois de criar o filtro de assinatura, o CloudWatch Logs encaminha todos os eventos de registro de entrada que correspondem ao padrão do filtro para o stream encapsulado no stream de destino chamado "". RecipientStream O proprietário do destino pode verificar se isso está acontecendo usando o get-shard-iterator comando aws kinesis para pegar um fragmento do Kinesis Data Streams e usando o comando aws kinesis get-records para buscar alguns registros do Kinesis Data Streams:

```
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 uma atriz de registros do Kinesis Data Streams. O atributo de dados no registro do Kinesis Data Streams 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 base64 decodificados e descompactados têm o formato JSON com a seguinte estrutura:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_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 AWS conta dos dados de registro 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 usam o tipo "DATA_MESSAGE". Às vezes, o CloudWatch Logs pode emitir registros do Kinesis Data Streams com o tipo “CONTROL_MESSAGE”, principalmente para verificar se o destino está 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.

Modificar a associação de destino no runtime

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 comando `put-destination-policy` em seu destino com uma 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 atualmente associada ao testDestination de destino e anote: AccessPolicy

```
aws logs describe-destinations \
```

```
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. Atualize a política para refletir que a conta 111111111111 foi interrompida e a conta 222222222222 está habilitada. Coloque essa política no arquivo NewAccessPolicy~/.json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. Ligue PutDestinationPolicypara associar a política definida no NewAccessPolicyarquivo.json ao 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. Os eventos de log da ID da conta 222222222222 começarão a fluir para o destino assim que o proprietário da conta 222222222222 criar um filtro de assinatura.

Atualizar uma assinatura existente entre contas

Se você atualmente tiver uma assinatura de logs entre contas em que a conta de destino concede permissões apenas para contas de remetente específicas e quiser atualizar essa assinatura para que a conta de destino conceda acesso a todas as contas em uma organização, siga as etapas desta seção.

Tópicos

- Etapa 1: atualizar os filtros de assinatura
- Etapa 2: atualizar a política de acesso ao destino existente

Etapa 1: atualizar os filtros de assinatura

> **Note**
>
> Essa etapa é necessária apenas para assinaturas entre contas para logs criados pelos serviços listados em Habilitar o registro a partir de AWS serviços. Se você não estiver trabalhando com logs criados por um desses grupos de log, pule para Etapa 2: atualizar a política de acesso ao destino existente.

Em determinados casos, você deve atualizar os filtros de assinatura em todas as contas de remetente que estão enviando logs para a conta de destino. A atualização adiciona uma função do IAM, que CloudWatch pode assumir e validar que a conta do remetente tem permissão para enviar registros para a conta do destinatário.

Siga as etapas desta seção para cada conta de remetente que você deseja atualizar para usar o ID da organização para as permissões de assinatura entre contas.

Nos exemplos desta seção, duas contas, 111111111111 e 222222222222 já têm filtros de assinatura criados para enviar logs para a conta 999999999999. Os valores de filtro de assinatura existentes são os seguintes:

```
## Existing Subscription Filter parameter values
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

Se você precisar encontrar os valores do parâmetro do filtro de assinatura atual, insira o seguinte comando.

```
aws logs describe-subscription-filters
    \ --log-group-name "my-log-group-name"
```

Atualizar um filtro de assinatura para começar a usar IDs da organização para permissões de log entre contas

1. Crie a seguinte 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.amazonaws.com" },
       "Action": "sts:AssumeRole"
     }
   }
   ```

2. Crie uma função do IAM que use essa política. Observe o valor Arn do valor Arn que for retornado pelo comando, pois você precisará dele posteriormente nesse procedimento. Neste exemplo, usamos CWLtoSubscriptionFilterRole como o nome da função que estamos criando.

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

3. Crie uma política de permissões para definir as ações que o CloudWatch Logs pode realizar na sua conta.

   a. Primeiro, use um editor de texto para criar a seguinte política de permissões em um arquivo chamado /PermissionsForCWLSubscriptionFilter.json.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. Insira o seguinte comando para associar a política de permissões que você acabou de criar à função criada na etapa 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

4. Insira o comando a seguir para atualizar o filtro de assinatura.

```
aws logs put-subscription-filter
    \ --log-group-name "my-log-group-name"
    \ --filter-name "RecipientStream"
    \ --filter-pattern "{$.userIdentity.type = Root}"
    \ --destination-arn
 "arn:aws:logs:region:999999999999:destination:testDestination"
    \ --role-arn "arn:aws:iam::111111111111:role/CWLtoSubscriptionFilterRole"
```

Etapa 2: atualizar a política de acesso ao destino existente

Depois de atualizar os filtros de assinatura em todas as contas de remetente, você poderá atualizar a política de acesso de destino na conta do destinatário.

Nos exemplos a seguir, a conta do destinatário é 999999999999 e o destino é chamado de `testDestination`.

A atualização habilita todas as contas que fazem parte da organização com ID de `o-1234567890` a enviar logs à conta destinatária. Somente as contas que tiverem filtros de assinatura criados enviarão logs para a conta do destinatário.

Atualizar a política de acesso de destino na conta do destinatário para começar a usar um ID de organização para permissões

1. Na conta destinatária, use um editor de texto para criar um arquivo `~/AccessPolicy.json` com o seguinte conteúdo.

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : "logs:PutSubscriptionFilter",
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

2. Insira o seguinte comando para anexar a política que você acabou de criar ao destino existente. Para atualizar um destino para usar uma política de acesso com um ID de organização em vez de uma política de acesso que lista IDs de conta específicos da AWS , inclua o parâmetro `force`.

> **Warning**
>
> Se você estiver trabalhando com registros enviados por um AWS serviço listado em Habilitar o registro a partir de AWS serviços, antes de executar essa etapa, você deve primeiro atualizar os filtros de assinatura em todas as contas do remetente, conforme explicado em Etapa 1: atualizar os filtros de assinatura.

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
```

```
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Compartilhamento de dados de registro entre contas usando Firehose

Para compartilhar dados de log entre contas, você precisa estabelecer um remetente e um destinatário dos dados de log:

- Remetente dos dados de registro — obtém as informações de destino do destinatário e informa ao CloudWatch Logs que ele está pronto para enviar seus eventos de registro para o destino especificado. Nos procedimentos do restante desta seção, o remetente dos dados de log é mostrado com um número de AWS conta fictício de 111111111111.
- Destinatário dos dados de log — configura um destino que encapsula um stream do Kinesis Data Streams e informa ao Logs que o CloudWatch destinatário deseja receber dados de log. O destinatário, então, compartilha as informações sobre esse destino com o remetente. Nos procedimentos do restante desta seção, o destinatário dos dados de registro é mostrado com um número de AWS conta fictício de 222222222222.

O exemplo nesta seção usa um stream de entrega do Firehose com armazenamento Amazon S3. Você também pode configurar streams de entrega do Firehose com configurações diferentes. Para obter mais informações, consulte Como criar um stream de entrega do Firehose.

> **Note**
>
> O grupo de registros e o destino devem estar na mesma AWS região. No entanto, o recurso da AWS para o qual o destino aponta pode estar localizado em uma região diferente.

> **Note**
>
> Há suporte para o filtro de assinatura Firehose para a mesma conta e fluxo de entrega entre regiões.

Tópicos

- Etapa 1: criar um stream de entrega do Firehose

- Etapa 2: Criar um destino
- Etapa 3: adicionar/validar as permissões do IAM para o destino entre contas
- Etapa 4: criar um filtro de assinatura
- Validação do fluxo de eventos de log
- Modificação da associação de destino no runtime

Etapa 1: criar um stream de entrega do Firehose

> **Important**
>
> Antes de concluir as etapas a seguir, você deve usar uma política de acesso para que o Firehose possa acessar seu bucket do Amazon S3. Para obter mais informações, consulte Controlling Access no Amazon Data Firehose Developer Guide.
> Todas as etapas desta seção (Etapa 1) devem ser realizadas na conta destinatária dos dados do log.
> Leste dos EUA (N. da Virgínia) é a região usada nos exemplos de comando a seguir. Substitua essa região pela região correta da implantação.

Para criar um stream de entrega do Firehose para ser usado como destino

1. Criar um bucket do Amazon S3:

```
aws s3api create-bucket --bucket firehose-test-bucket1 --create-bucket-
configuration LocationConstraint=us-east-1
```

2. Crie a função do IAM que concede permissão ao Firehose para colocar dados no bucket.

   a. Primeiro, use um editor de texto para criar uma política de confiança em um arquivo `~/TrustPolicyForFirehose.json`.

```
{ "Statement": { "Effect": "Allow", "Principal": { "Service":
 "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
 { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
```

   b. Crie a função do IAM, especificando o arquivo de política de confiança que você acabou de criar.

```
aws iam create-role \
    --role-name FirehosetoS3Role \
    --assume-role-policy-document file://~/TrustPolicyForFirehose.json
```

c. A saída deste comando será parecida com o exemplo a seguir. Anote os valores do nome da função e do ARN da função.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "FirehosetoS3Role",
        "RoleId": "AROAR3BXASEKW7K635M53",
        "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
        "CreateDate": "2021-02-02T07:53:10+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:ExternalId": "222222222222"
                    }
                }
            }
        }
    }
}
```

3. Crie uma política de permissões para definir as ações que o Firehose pode realizar em sua conta.

a. Primeiro, use um editor de texto para criar a seguinte política de permissões em um arquivo chamado `~/PermissionsForFirehose.json`. Dependendo do caso de uso, pode ser necessário adicionar mais permissões a esse arquivo.

```
{
    "Statement": [{
        "Effect": "Allow",
```

```
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::firehose-test-bucket1",
            "arn:aws:s3:::firehose-test-bucket1/*"
        ]
    }]
}
```

b. Insira o seguinte comando para associar a política de permissões que você acabou de criar aoa perfil do IAM.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. Digite o comando a seguir para criar o stream de entrega do Firehose. Substitua *my-role-arn* e *my-bucket-arn* com os valores corretos para sua implantação.

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::firehose-test-bucket1"}'
```

A saída deve ser semelhante à seguinte:

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

Etapa 2: Criar um destino

> **Important**
>
> As etapas deste procedimento devem ser processadas na conta destinatária dos dados do log.

Quando o destino é criado, o CloudWatch Logs envia uma mensagem de teste para o destino em nome da conta do destinatário. Quando o filtro de assinatura é ativado posteriormente, o CloudWatch Logs envia eventos de registro para o destino em nome da conta de origem.

Para criar um destino

1. Espere até que o stream do Firehose que você criou Etapa 1: criar um stream de entrega do Firehose se torne ativo. Você pode usar o comando a seguir para verificar StreamDescriptiono. StreamStatuspropriedade.

   ```
   aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
   ```

   Além disso, anote DeliveryStreamDescriptiono. DeliveryStreamValor do ARN, porque você precisará usá-lo em uma etapa posterior. Exemplo de saída desse comando:

   ```
   {
       "DeliveryStreamDescription": {
           "DeliveryStreamName": "my-delivery-stream",
           "DeliveryStreamARN": "arn:aws:firehose:us-
   east-1:222222222222:deliverystream/my-delivery-stream",
           "DeliveryStreamStatus": "ACTIVE",
           "DeliveryStreamEncryptionConfiguration": {
               "Status": "DISABLED"
           },
           "DeliveryStreamType": "DirectPut",
           "VersionId": "1",
           "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
           "Destinations": [
               {
                   "DestinationId": "destinationId-000000000001",
                   "S3DestinationDescription": {
                       "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                       "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
   ```

```
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

Pode levar um ou dois minutos para seu fluxo de entrega ser exibido com estado ativo.

2. Quando o stream de entrega estiver ativo, crie a função do IAM que concederá a CloudWatch Logs a permissão para colocar dados em seu stream do Firehose. Primeiro, você precisará criar uma política de confiança em um arquivo TrustPolicyFor~/CWL.json. Use um editor de texto para criar esta política. Para obter mais informações sobre endpoints do CloudWatch Logs, consulte Endpoints e cotas do Amazon CloudWatch Logs.

Esta política inclui uma chave de contexto de condição global `aws:SourceArn` que especifica o `sourceAccountId` para evitar o problema de segurança confused deputy. Se você ainda não souber o ID da conta de origem na primeira chamada, recomendamos que você coloque o ARN de destino no campo ARN de origem. Nas chamadas subsequentes, você deve definir o ARN de origem como o ARN de origem real que você coletou da primeira chamada. Para ter mais informações, consulte Prevenção de 'confused deputy'.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.region.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
     }
}
```

3. Use o comando aws iam create-role para criar a função do IAM, especificando o arquivo de política de confiança que você acabou de criar.

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

Este é um exemplo de saída. Anote o valor `Role.Arn` retornado, pois será necessário utilizá-lo em uma etapa posterior.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
```

```
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2021-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                }
            }
        }
    }
}
```

4. Crie uma política de permissões para definir quais ações o CloudWatch Logs pode realizar na sua conta. Primeiro, use um editor de texto para criar uma política de permissões em um arquivo PermissionsFor~/CWL.json:

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
}
```

5. Associe a política de permissões com a função inserindo o seguinte comando:

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

6. Depois que o stream de entrega do Firehose estiver no estado ativo e você tiver criado a função do IAM, você poderá criar o destino dos CloudWatch registros.

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 ARN do novo destino que for retornado na carga útil, pois você usará isso como `destination.arn` em uma etapa posterior.

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

b. Depois que a etapa anterior for concluída, na conta destinatária dos dados de log (222222222222), associe uma política de acesso ao destino.

   Essa política permite que a conta remetente dos dados de log (111111111111 conta) acesse o destino apenas na conta destinatária dos dados de log (222222222222). 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:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
```

```
  ]
}
```

c. Isso cria uma política que define quem tem acesso de gravação ao destino. Essa política deve especificar os registros: PutSubscriptionFilter ação para acessar o destino. Usuários de várias contas usarão a PutSubscriptionFilteração para enviar eventos de log para o destino:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Etapa 3: adicionar/validar as permissões do IAM para o destino entre contas

De acordo com a lógica de avaliação de políticas AWS entre contas, para acessar qualquer recurso entre contas (como um stream do Kinesis ou Firehose usado como destino para um filtro de assinatura), você deve ter uma política baseada em identidade na conta de envio que forneça acesso explícito ao recurso de destino entre contas. Para obter mais informações sobre a lógica de avaliação de política, consulte Lógica de avaliação de política entre contas.

Você pode associar a política baseada em identidade ao perfil do IAM ou ao usuário do IAM que você está usando para criar o filtro de assinatura. Essa política deve estar presente na conta de envio. Se você estiver usando a função de administrador para criar o filtro de assinatura, poderá ignorar esta etapa e prosseguir para Etapa 4: criar um filtro de assinatura.

Para adicionar ou validar as permissões do IAM necessárias entre contas

1. Insira o comando a seguir para verificar qual perfil do IAM ou usuário do IAM está sendo usado para executar comandos de log da AWS .

   ```
   aws sts get-caller-identity
   ```

   Esse comando retorna uma saída semelhante à seguinte:

   ```
   {
   "UserId": "User ID",
   "Account": "sending account id",
   "Arn": "arn:aws:sending account id:role/user:RoleName/UserName"
   }
   ```

Anote o valor representado por *RoleName* ou *UserName*.

2. Faça login AWS Management Console na conta de envio e pesquise as políticas anexadas com a função do IAM ou o usuário do IAM retornado na saída do comando inserido na etapa 1.
3. Verifique se as políticas vinculadas a esse perfil ou usuário fornecem permissões explícitas para chamar `logs:PutSubscriptionFilter` no recurso de destino entre contas. O exemplo de políticas a seguir mostra as permissões recomendadas.

   A política a seguir fornece permissões para criar um filtro de assinatura em qualquer recurso de destino somente em uma única AWS conta, conta`123456789012`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on any resource in one specific
 account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
                "arn:aws:logs:*:123456789012:destination:*"
            ]
        }
    ]
}
```

   A política a seguir fornece permissões para criar um filtro de assinatura somente em um recurso de destino específico chamado `sampleDestination` em uma única AWS conta, conta`123456789012`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow subscription filters on one specific resource in one
 specific account",
            "Effect": "Allow",
            "Action": "logs:PutSubscriptionFilter",
            "Resource": [
                "arn:aws:logs:*:*:log-group:*",
```

```
                "arn:aws:logs:*:123456789012:destination:sampleDestination"
            ]
        }
    ]
}
```

Etapa 4: criar um filtro de assinatura

Alterne para a conta de envio, que é 111111111111 neste exemplo. Agora, você criará o filtro de assinatura na conta de envio. Neste exemplo, o filtro está associado a um grupo de registros contendo AWS CloudTrail eventos para que cada atividade registrada feita pelas AWS credenciais "Raiz" seja entregue ao destino que você criou anteriormente. Para obter mais informações sobre como enviar AWS CloudTrail eventos para CloudWatch registros, consulte Enviar CloudTrail eventos para CloudWatch registros no Guia do AWS CloudTrail usuário.

Ao inserir o comando a seguir, certifique-se de estar conectado como usuário do IAM ou usando o perfil do IAM para o qual você adicionou a política, em Etapa 3: adicionar/validar as permissões do IAM para o destino entre contas.

```
aws logs put-subscription-filter \
    --log-group-name "aws-cloudtrail-logs-111111111111-300a971e" \
    --filter-name "firehose_test" \
    --filter-pattern "{$.userIdentity.type = AssumedRole}" \
    --destination-arn "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
```

O grupo de registros e o destino devem estar na mesma AWS região. No entanto, o destino pode apontar para um AWS recurso, como um stream do Firehose, 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 registro de entrada que correspondem ao padrão do filtro para o stream de entrega do Firehose. Os dados começam a aparecer em seu bucket do Amazon S3 com base no intervalo de tempo definido no stream de entrega do Firehose. Quando tiver passado tempo suficiente, você poderá conferir seus dados verificando o bucket do Amazon S3. Para verificar o bucket, insira este comando:

```
aws s3api list-objects --bucket 'firehose-test-bucket1'
```

A saída desse comando será semelhante a:

```
{
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2021-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
    ]
}
```

Em seguida, você pode recuperar um objeto específico do bucket digitando o comando a seguir. Substitua o valor de `key` pelo valor encontrado no comando anterior.

```
aws s3api get-object --bucket 'firehose-test-bucket1' --key '2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

Os dados no objeto do Amazon S3 são compactados com o formato gzip. É possível examinar os dados brutos na linha de comando usando os seguintes comandos:

Linux

```
zcat testfile.gz
```

macOS:

```
zcat <testfile.gz
```

Modificação da associação de destino no runtime

Pode haver situações em que você precise adicionar ou remover remetentes de log de um destino que você possua. Você pode usar a PutDestinationPolicyação em 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 333333333333 é habilitada.

1. Busque a política atualmente associada ao testDestination de destino e anote: AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testFirehoseDestination"

{
    "destinations": [
        {
            "destinationName": "testFirehoseDestination",
            "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
            "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
            "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
\" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
\" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
            "arn": "arn:aws:logs:us-east-1:
 222222222222:destination:testFirehoseDestination",
            "creationTime": 1612256124430
        }
    ]
}
```

2. Atualize a política para refletir que a conta 111111111111 foi interrompida e a conta 333333333333 está habilitada. Coloque essa política no arquivo NewAccessPolicy~/.json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "333333333333 "
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
```

```
  ]
}
```

3. Use o comando a seguir para associar a política definida no NewAccessPolicyarquivo.json ao destino:

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \

    --access-policy file://~/NewAccessPolicy.json
```

Por fim, isso desativa os eventos de log do ID da conta 111111111111. Os eventos de log da ID da conta 333333333333 começarão a fluir para o destino assim que o proprietário da conta 333333333333 criar um filtro de assinatura.

Assinaturas multiregionais em nível de conta usando o Kinesis Data Streams

Ao criar uma assinatura entre contas, você pode especificar uma única conta ou organização para ser o remetente. Se você especificar uma organização, esse procedimento permitirá que todas as contas da organização enviem logs para a conta do receptor.

Para compartilhar dados de log entre contas, você precisa estabelecer um remetente e um destinatário dos dados de log:

- Remetente dos dados de registro — obtém as informações de destino do destinatário e informa ao CloudWatch Logs que ele está pronto para enviar seus eventos de registro para o destino especificado. Nos procedimentos do restante desta seção, o remetente dos dados de log é mostrado com um número de AWS conta fictício de 111111111111.

  Se você quiser que várias contas em uma organização enviem logs para uma conta de destinatário, você pode criar uma política que conceda a todas as contas da organização a permissão para enviar logs para a conta do destinatário. Você ainda precisa configurar filtros de assinatura diferentes para cada conta de remetente.
- Destinatário dos dados de log — configura um destino que encapsula um stream do Kinesis Data Streams e informa ao Logs que o CloudWatch destinatário deseja receber dados de log. O destinatário, então, compartilha as informações sobre esse destino com o remetente. Nos

procedimentos do restante desta seção, o destinatário dos dados de registro é mostrado com um número de AWS conta fictício de 999999999999.

Para começar a receber eventos de registro de usuários de várias contas, o destinatário dos dados de registro primeiro cria um destino de CloudWatch registros. Cada destino consiste nos seguintes elementos-chave:

Nome do destino

O nome do destino que você deseja criar.

ARN de destino

O Amazon Resource Name (ARN) do AWS recurso que você deseja usar como destino do feed de assinatura.

ARN de função

Uma função AWS Identity and Access Management (IAM) que concede aos CloudWatch Logs as permissões necessárias para colocar dados no stream escolhido.

Política de acesso

Documento de políticas 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.

> **Note**
>
> O grupo de registros e o destino devem estar na mesma AWS região. No entanto, o recurso da AWS para o qual o destino aponta pode estar localizado em uma região diferente. Nos exemplos das seções a seguir, todos os recursos específicos da região são criados no Leste dos EUA (Virgínia do Norte).

Tópicos

- Como configurar uma nova assinatura entre contas
- Atualizar uma assinatura existente entre contas

Como configurar uma nova assinatura entre contas

Siga as etapas nestas seções para configurar uma nova assinatura de log entre contas.

Tópicos

- Etapa 1: criar um destino
- Etapa 2: (somente se estiver usando uma organização) Crie uma função do IAM.
- Etapa 3: criar uma política de filtro de assinatura em nível de conta
- Validação do fluxo de eventos de logs
- Modificar a associação de destino no runtime

Etapa 1: criar um destino

> **Important**
>
> As etapas deste procedimento devem ser processadas na conta destinatária dos dados do log.

Neste exemplo, a conta do destinatário dos dados de registro tem uma ID de conta de 999999999999, enquanto a ID da AWS conta do remetente dos dados de registro é 111111111111. AWS

Este exemplo cria um destino usando um stream do Kinesis Data RecipientStream Streams chamado, e uma função CloudWatch que permite que o Logs grave dados nele.

Quando o destino é criado, o CloudWatch Logs envia uma mensagem de teste para o destino em nome da conta do destinatário. Quando o filtro de assinatura é ativado posteriormente, o CloudWatch Logs envia eventos de registro para o destino em nome da conta de origem.

Para criar um destino

1. Na conta do destinatário, crie um fluxo de destino no Kinesis Data Streams. Em um prompt de comando, digite:

```
aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
```

2. Aguarde até que o fluxo do fique ativo. Você pode usar o comando aws kinesis describe-stream para verificar o. StreamDescription StreamStatuspropriedade. Além disso, anote o valor StreamDescription.streamArn porque você o passará para CloudWatch o Logs posteriormente:

```
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":
 "49551135218688818456679503831981458784591352702 18EXAMPLE"
        }
      }
    ]
  }
}
```

Pode levar um ou dois minutos para o seu stream aparecer no estado ativo.

3. Crie a função do IAM que concede a CloudWatch Logs a permissão para colocar dados em seu stream. Primeiro, você precisará criar uma política de confiança em um arquivo TrustPolicyFor~/CWL.json. Use um editor de texto para criar esse arquivo de política, não use o console do IAM.

   Esta política inclui uma chave de contexto de condição global `aws:SourceArn` que especifica o `sourceAccountId` para evitar o problema de segurança confused deputy. Se você ainda não souber o ID da conta de origem na primeira chamada, recomendamos que você coloque o ARN de destino no campo ARN de origem. Nas chamadas subsequentes, você deve definir o ARN de origem como o ARN de origem real que você coletou da primeira chamada. Para obter mais informações, consulte Prevenção de ‘confused deputy’.

```
{
    "Statement": {
```

```
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        },
        "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 passado para Logs posteriormente: CloudWatch

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

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
                    }
                },
                "Principal": {
                    "Service": "logs.amazonaws.com"
                }
            }
        },
```

```
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2023-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Crie uma política de permissões para definir quais ações o CloudWatch Logs pode realizar na sua conta. Primeiro, use um editor de texto para criar uma política de permissões em um arquivo PermissionsFor~/CWL.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 put-role-policy comando aws iam:

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

7. Depois que o stream estiver no estado ativo e você tiver criado a função do IAM, você poderá criar o destino dos CloudWatch registros.

   a. Esta etapa não associará uma política de acesso ao seu destino e é apenas a primeira etapa de duas que concluirá uma criação de destino. Anote o DestinationArnque é 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. Essa política deve especificar os registros: PutSubscriptionFilter ação e concede permissão à conta do remetente para acessar o destino.

   A política concede permissão à AWS conta que envia os registros. Você pode especificar apenas essa conta na política ou, se a conta de remetente for parte de uma organização, a política poderá especificar o ID da organização. Dessa forma, você pode criar apenas uma política para permitir que várias contas em uma organização enviem logs para essa conta de destino.

   Use um editor de texto para criar um arquivo chamado `~/AccessPolicy.json` com uma das seguintes declarações de política.

   Este primeiro exemplo de política permite que todas as contas na organização que tenham um ID de `o-1234567890` enviem logs à conta do destinatário.

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
               "StringEquals" : {
                   "aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
```

```
}
```

Este próximo exemplo permite que apenas a conta remetente dos dados de log (111111111111) envie logs à conta destinatária dos dados de log.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

c. Anexe a política criada na etapa anterior ao destino.

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Essa política de acesso permite que os usuários da AWS Conta com ID 111111111111 liguem para o destino com o ARN arn:aws:logs: ***PutSubscriptionFilter****region:9999999999:destination:testDestination.* A tentativa de qualquer outro usuário de ligar PutSubscriptionFilter para esse destino será rejeitada.

Para validar os privilégios de um usuário com base em uma política de acesso, consulte Usar o validador de políticas no Manual do usuário do IAM.

Ao terminar, se estiver usando AWS Organizations suas permissões entre contas, siga as etapas emEtapa 2: (somente se estiver usando uma organização) Crie uma função do IAM.. Se você estiver concedendo permissões diretamente para a outra conta em vez de usar Organizations, você pode

pular essa etapa e prosseguir para Etapa 3: criar uma política de filtro de assinatura em nível de conta.

Etapa 2: (somente se estiver usando uma organização) Crie uma função do IAM.

Na seção anterior, se você criou o destino usando uma política de acesso que concede permissões à organização em que está a conta 111111111111, em vez de conceder permissões diretamente para a conta 111111111111, siga as etapas nesta seção. Caso contrário, pule para Etapa 3: criar uma política de filtro de assinatura em nível de conta.

As etapas desta seção criam uma função do IAM, que CloudWatch pode assumir e validar se a conta do remetente tem permissão para criar um filtro de assinatura em relação ao destino do destinatário.

Siga as etapas nesta seção na conta do remetente. A função deve existir na conta do remetente e você especifica o ARN dessa função no filtro de assinatura. Neste exemplo, a conta de remetente é 111111111111.

Criar a função do IAM necessária para assinaturas de log entre contas usando o AWS Organizations

1. Crie a seguinte política de confiança em um arquivo `/TrustPolicyForCWLSubscriptionFilter.json`. Use um editor de texto para criar esse arquivo de política; não use o console do IAM.

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

2. Crie uma função do IAM que use essa política. Anote o valor `Arn` retornado pelo comando, pois você precisará dele posteriormente nesse procedimento. Neste exemplo, usamos `CWLtoSubscriptionFilterRole` como o nome da função que estamos criando.

```
aws iam create-role \
     --role-name CWLtoSubscriptionFilterRole \
     --assume-role-policy-document file://~/
TrustPolicyForCWLSubscriptionFilter.json
```

3. Crie uma política de permissões para definir as ações que o CloudWatch Logs pode realizar na sua conta.

   a. Primeiro, use um editor de texto para criar a seguinte política de permissões em um arquivo chamado ~/PermissionsForCWLSubscriptionFilter.json.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

   b. Insira o seguinte comando para associar a política de permissões que você acabou de criar à função criada na etapa 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

Quando terminar, você pode prosseguir para Etapa 3: criar uma política de filtro de assinatura em nível de conta.

Etapa 3: criar uma política de filtro de assinatura em nível de conta

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.

> **Note**
>
> Se você estiver concedendo permissões para o filtro de assinatura a uma organização inteira, precisará usar o ARN do perfil do IAM que criou em Etapa 2: (somente se estiver usando uma organização) Crie uma função do IAM..

No exemplo a seguir, uma política de filtro de assinatura em nível de conta é criada em uma conta de envio. O filtro é associado à conta do remetente 111111111111 para que cada evento de registro que corresponda ao filtro e aos critérios de seleção seja entregue ao destino que você criou anteriormente. Esse destino encapsula um fluxo chamado ""RecipientStream.

O `selection-criteria` campo é opcional, mas é importante para excluir grupos de registros que podem causar uma recursão infinita de registros de um filtro de assinatura. Para obter mais informações sobre esse problema e determinar quais grupos de registros excluir, consultePrevenção de recursão de registros. Atualmente, NOT IN é o único operador compatível com `selection-criteria` o.

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Os grupos de registros da conta do remetente e o destino devem estar na mesma AWS região. No entanto, o destino pode apontar para um AWS recurso, como um stream do Kinesis Data Streams, localizado em uma região diferente.

Validação do fluxo de eventos de logs

Depois de criar a política de filtro de assinatura em nível de conta, o CloudWatch Logs encaminha todos os eventos de registro de entrada que correspondam ao padrão de filtro e aos critérios de seleção para o stream encapsulado no stream de destino chamado "". RecipientStream O proprietário do destino pode verificar se isso está acontecendo usando o get-shard-iterator comando aws kinesis para pegar um fragmento do Kinesis Data Streams e usando o comando aws kinesis get-records para buscar alguns registros do Kinesis Data Streams:

```
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**
>
> Talvez seja necessário executar novamente o `get-records` comando algumas vezes antes que o Kinesis Data Streams comece a retornar os dados.

Você deve ver uma resposta com uma atriz de registros do Kinesis Data Streams. O atributo de dados no registro do Kinesis Data Streams 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 base64 decodificados e descompactados têm o formato JSON com a seguinte estrutura:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail/logs",
    "logStream": "111111111111_CloudTrail/logs_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 da estrutura de dados são os seguintes:

messageType

As mensagens de dados usarão o tipo "DATA_MESSAGE". Às vezes, o CloudWatch Logs pode emitir registros do Kinesis Data Streams com o tipo “CONTROL_MESSAGE”, principalmente para verificar se o destino está acessível.

owner

O ID da AWS conta dos dados de registro 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.

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.

Nível de política

O nível em que a política foi aplicada. “ACCOUNT_LEVEL_POLICY” serve `policyLevel` para uma política de filtro de assinatura em nível de conta.

Modificar a associação de destino no runtime

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 comando `put-destination-policy` em seu destino com uma 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 atualmente associada ao testDestination de destino e anote: AccessPolicy

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
     "DestinationArn":
 "arn:aws:logs:region:999999999999:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:999999999999:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:999999999999:destination:testDestination\"}] }"
   }
 ]
}
```

2. Atualize a política para refletir que a conta 111111111111 foi interrompida e a conta 222222222222 está habilitada. Coloque essa política no arquivo NewAccessPolicy~/.json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

3. Ligue PutDestinationPolicypara associar a política definida no NewAccessPolicyarquivo.json ao 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. Os eventos de log da ID da conta 222222222222 começarão a fluir para o destino assim que o proprietário da conta 222222222222 criar um filtro de assinatura.

Atualizar uma assinatura existente entre contas

Se você atualmente tiver uma assinatura de logs entre contas em que a conta de destino concede permissões apenas para contas de remetente específicas e quiser atualizar essa assinatura para que a conta de destino conceda acesso a todas as contas em uma organização, siga as etapas desta seção.

Tópicos

- Etapa 1: atualizar os filtros de assinatura
- Etapa 2: atualizar a política de acesso ao destino existente

Etapa 1: atualizar os filtros de assinatura

> **Note**
>
> Essa etapa é necessária apenas para assinaturas entre contas para logs criados pelos serviços listados em Habilitar o registro a partir de AWS serviços. Se você não estiver trabalhando com logs criados por um desses grupos de log, pule para Etapa 2: atualizar a política de acesso ao destino existente.

Em determinados casos, você deve atualizar os filtros de assinatura em todas as contas de remetente que estão enviando logs para a conta de destino. A atualização adiciona uma função do IAM, que CloudWatch pode assumir e validar que a conta do remetente tem permissão para enviar registros para a conta do destinatário.

Siga as etapas desta seção para cada conta de remetente que você deseja atualizar para usar o ID da organização para as permissões de assinatura entre contas.

Nos exemplos desta seção, duas contas, 111111111111 e 222222222222 já têm filtros de assinatura criados para enviar logs para a conta 999999999999. Os valores de filtro de assinatura existentes são os seguintes:

```
## Existing Subscription Filter parameter values
{
    "DestinationArn": "arn:aws:logs:region:999999999999:destination:testDestination",
    "FilterPattern": "{$.userIdentity.type = Root}",
    "Distribution": "Random"
}
```

Se você precisar encontrar os valores do parâmetro do filtro de assinatura atual, insira o seguinte comando.

```
aws logs describe-account-policies \
--policy-type "SUBSCRIPTION_FILTER_POLICY" \
--policy-name "CrossAccountStreamsExamplePolicy"
```

Atualizar um filtro de assinatura para começar a usar IDs da organização para permissões de log entre contas

1. Crie a seguinte 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.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

2. Crie uma função do IAM que use essa política. Observe o valor Arn do valor Arn que for retornado pelo comando, pois você precisará dele posteriormente nesse procedimento. Neste exemplo, usamos CWLtoSubscriptionFilterRole como o nome da função que estamos criando.

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

3. Crie uma política de permissões para definir as ações que o CloudWatch Logs pode realizar na sua conta.

   a. Primeiro, use um editor de texto para criar a seguinte política de permissões em um arquivo chamado /PermissionsForCWLSubscriptionFilter.json.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:region:111111111111:log-
group:LogGroupOnWhichSubscriptionFilterIsCreated:*"
        }
    ]
}
```

b. Insira o seguinte comando para associar a política de permissões que você acabou de criar à função criada na etapa 2.

```
aws iam put-role-policy
    --role-name CWLtoSubscriptionFilterRole
    --policy-name Permissions-Policy-For-CWL-Subscription-filter
    --policy-document file://~/PermissionsForCWLSubscriptionFilter.json
```

4. Digite o comando a seguir para atualizar a política de filtro de assinatura.

```
aws logs put-account-policy \
    --policy-name "CrossAccountStreamsExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document
 '{"DestinationArn":"arn:aws:logs:region:999999999999:destination:testDestination",
 "FilterPattern": "{$.userIdentity.type = Root}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Etapa 2: atualizar a política de acesso ao destino existente

Depois de atualizar os filtros de assinatura em todas as contas de remetente, você poderá atualizar a política de acesso de destino na conta do destinatário.

Nos exemplos a seguir, a conta do destinatário é 999999999999 e o destino é chamado de `testDestination`.

A atualização habilita todas as contas que fazem parte da organização com ID de `o-1234567890` a enviar logs à conta destinatária. Somente as contas que tiverem filtros de assinatura criados enviarão logs para a conta do destinatário.

Atualizar a política de acesso de destino na conta do destinatário para começar a usar um ID de organização para permissões

1. Na conta destinatária, use um editor de texto para criar um arquivo `~/AccessPolicy.json` com o seguinte conteúdo.

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

```
    "Statement" : [
        {
"Sid" : "",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
            "Resource" :
 "arn:aws:logs:region:999999999999:destination:testDestination",
            "Condition": {
"StringEquals" : {
"aws:PrincipalOrgID" : ["o-1234567890"]
                }
            }
        }
    ]
}
```

2. Insira o seguinte comando para anexar a política que você acabou de criar ao destino existente. Para atualizar um destino para usar uma política de acesso com um ID de organização em vez de uma política de acesso que lista IDs de conta específicos da AWS , inclua o parâmetro `force`.

> **Warning**
>
> Se você estiver trabalhando com registros enviados por um AWS serviço listado emHabilitar o registro a partir de AWS serviços, antes de executar essa etapa, você deve primeiro atualizar os filtros de assinatura em todas as contas do remetente, conforme explicado emEtapa 1: atualizar os filtros de assinatura.

```
aws logs put-destination-policy
    \ --destination-name "testDestination"
    \ --access-policy file://~/AccessPolicy.json
    \ --force
```

Assinaturas multiregionais em nível de conta usando Firehose

Para compartilhar dados de log entre contas, você precisa estabelecer um remetente e um destinatário dos dados de log:

- Remetente dos dados de registro — obtém as informações de destino do destinatário e informa ao CloudWatch Logs que ele está pronto para enviar seus eventos de registro para o destino especificado. Nos procedimentos do restante desta seção, o remetente dos dados de log é mostrado com um número de AWS conta fictício de 111111111111.
- Destinatário dos dados de log — configura um destino que encapsula um stream do Kinesis Data Streams e informa ao Logs que o CloudWatch destinatário deseja receber dados de log. O destinatário, então, compartilha as informações sobre esse destino com o remetente. Nos procedimentos do restante desta seção, o destinatário dos dados de registro é mostrado com um número de AWS conta fictício de 222222222222.

O exemplo nesta seção usa um stream de entrega do Firehose com armazenamento Amazon S3. Você também pode configurar streams de entrega do Firehose com configurações diferentes. Para obter mais informações, consulte Como criar um stream de entrega do Firehose.

> **Note**
>
> O grupo de registros e o destino devem estar na mesma AWS região. No entanto, o recurso da AWS para o qual o destino aponta pode estar localizado em uma região diferente.

> **Note**
>
> Há suporte para o filtro de assinatura Firehose para a mesma conta e fluxo de entrega entre regiões.

Tópicos

- Etapa 1: criar um stream de entrega do Firehose
- Etapa 2: Criar um destino
- Etapa 3: criar uma política de filtro de assinatura em nível de conta
- Validação do fluxo de eventos de log
- Modificação da associação de destino no runtime

Etapa 1: criar um stream de entrega do Firehose

> **Important**
>
> Antes de concluir as etapas a seguir, você deve usar uma política de acesso para que o Firehose possa acessar seu bucket do Amazon S3. Para obter mais informações, consulte Controlling Access no Amazon Data Firehose Developer Guide.
> Todas as etapas desta seção (Etapa 1) devem ser realizadas na conta destinatária dos dados do log.
> Leste dos EUA (N. da Virgínia) é a região usada nos exemplos de comando a seguir. Substitua essa região pela região correta da implantação.

Para criar um stream de entrega do Firehose para ser usado como destino

1. Criar um bucket do Amazon S3:

```
aws s3api create-bucket --bucket firehose-test-bucket1 --create-bucket-
configuration LocationConstraint=us-east-1
```

2. Crie a função do IAM que concede permissão ao Firehose para colocar dados no bucket.

   a. Primeiro, use um editor de texto para criar uma política de confiança em um arquivo `~/TrustPolicyForFirehose.json`.

```
{ "Statement": { "Effect": "Allow", "Principal": { "Service":
 "firehose.amazonaws.com" }, "Action": "sts:AssumeRole", "Condition":
 { "StringEquals": { "sts:ExternalId":"222222222222" } } } }
```

   b. Crie a função do IAM, especificando o arquivo de política de confiança que você acabou de criar.

```
aws iam create-role \
    --role-name FirehosetoS3Role \
    --assume-role-policy-document file://~/TrustPolicyForFirehose.json
```

   c. A saída deste comando será parecida com o exemplo a seguir. Anote os valores do nome da função e do ARN da função.

```
{
```

```
    "Role": {
        "Path": "/",
        "RoleName": "FirehosetoS3Role",
        "RoleId": "AROAR3BXASEKW7K635M53",
        "Arn": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
        "CreateDate": "2021-02-02T07:53:10+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:ExternalId": "222222222222"
                    }
                }
            }
        }
    }
}
```

3. Crie uma política de permissões para definir as ações que o Firehose pode realizar em sua conta.

   a. Primeiro, use um editor de texto para criar a seguinte política de permissões em um arquivo chamado `~/PermissionsForFirehose.json`. Dependendo do caso de uso, pode ser necessário adicionar mais permissões a esse arquivo.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:ListBucket"
        ],
        "Resource": [
            "arn:aws:s3:::firehose-test-bucket1",
            "arn:aws:s3:::firehose-test-bucket1/*"
        ]
    }]
```

```
}
```

b. Insira o seguinte comando para associar a política de permissões que você acabou de criar aoa perfil do IAM.

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name
 Permissions-Policy-For-Firehose-To-S3 --policy-document file://~/
PermissionsForFirehose.json
```

4. Digite o comando a seguir para criar o stream de entrega do Firehose. Substitua *my-role-arn* e *my-bucket-arn* com os valores corretos para sua implantação.

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::firehose-test-bucket1"}'
```

A saída deve ser semelhante à seguinte:

```
{
    "DeliveryStreamARN": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream"
}
```

Etapa 2: Criar um destino

> **Important**
>
> As etapas deste procedimento devem ser processadas na conta destinatária dos dados do log.

Quando o destino é criado, o CloudWatch Logs envia uma mensagem de teste para o destino em nome da conta do destinatário. Quando o filtro de assinatura é ativado posteriormente, o CloudWatch Logs envia eventos de registro para o destino em nome da conta de origem.

Para criar um destino

1. Espere até que o stream do Firehose que você criou Etapa 1: criar um stream de entrega do Firehose se torne ativo. Você pode usar o comando a seguir para verificar StreamDescriptiono. StreamStatuspropriedade.

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
```

   Além disso, anote DeliveryStreamDescriptiono. DeliveryStreamValor do ARN, porque você precisará usá-lo em uma etapa posterior. Exemplo de saída desse comando:

```
{
    "DeliveryStreamDescription": {
        "DeliveryStreamName": "my-delivery-stream",
        "DeliveryStreamARN": "arn:aws:firehose:us-
east-1:222222222222:deliverystream/my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamEncryptionConfiguration": {
            "Status": "DISABLED"
        },
        "DeliveryStreamType": "DirectPut",
        "VersionId": "1",
        "CreateTimestamp": "2021-02-01T23:59:15.567000-08:00",
        "Destinations": [
            {
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    }
                },
                "ExtendedS3DestinationDescription": {
```

```
                    "RoleARN": "arn:aws:iam::222222222222:role/FirehosetoS3Role",
                    "BucketARN": "arn:aws:s3:::firehose-test-bucket1",
                    "BufferingHints": {
                        "SizeInMBs": 5,
                        "IntervalInSeconds": 300
                    },
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "CloudWatchLoggingOptions": {
                        "Enabled": false
                    },
                    "S3BackupMode": "Disabled"
                }
            }
        ],
        "HasMoreDestinations": false
    }
}
```

Pode levar um ou dois minutos para seu fluxo de entrega ser exibido com estado ativo.

2. Quando o stream de entrega estiver ativo, crie a função do IAM que concederá a CloudWatch Logs a permissão para colocar dados em seu stream do Firehose. Primeiro, você precisará criar uma política de confiança em um arquivo TrustPolicyFor~/CWL.json. Use um editor de texto para criar esta política. Para obter mais informações sobre endpoints do CloudWatch Logs, consulte Endpoints e cotas do Amazon CloudWatch Logs.

   Esta política inclui uma chave de contexto de condição global `aws:SourceArn` que especifica o `sourceAccountId` para evitar o problema de segurança confused deputy. Se você ainda não souber o ID da conta de origem na primeira chamada, recomendamos que você coloque o ARN de destino no campo ARN de origem. Nas chamadas subsequentes, você deve definir o ARN de origem como o ARN de origem real que você coletou da primeira chamada. Para ter mais informações, consulte Prevenção de 'confused deputy'.

```
{
    "Statement": {
        "Effect": "Allow",
        "Principal": {
            "Service": "logs.amazonaws.com"
        },
```

```
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringLike": {
                "aws:SourceArn": [
                    "arn:aws:logs:region:sourceAccountId:*",
                    "arn:aws:logs:region:recipientAccountId:*"
                ]
            }
        }
    }
}
```

3. Use o comando aws iam create-role para criar a função do IAM, especificando o arquivo de política de confiança que você acabou de criar.

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

Este é um exemplo de saída. Anote o valor `Role.Arn` retornado, pois será necessário utilizá-lo em uma etapa posterior.

```
{
    "Role": {
        "Path": "/",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "RoleId": "AROAR3BXASEKYJYWF243H",
        "Arn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "CreateDate": "2023-02-02T08:10:43+00:00",
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:region:sourceAccountId:*",
                            "arn:aws:logs:region:recipientAccountId:*"
                        ]
```

```
                    }
                }
            }
        }
    }
}
```

4. Crie uma política de permissões para definir quais ações o CloudWatch Logs pode realizar na sua conta. Primeiro, use um editor de texto para criar uma política de permissões em um arquivo PermissionsFor~/CWL.json:

```
{
    "Statement":[
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:222222222222:*"]
      }
    ]
}
```

5. Associe a política de permissões com a função inserindo o seguinte comando:

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

6. Depois que o stream de entrega do Firehose estiver no estado ativo e você tiver criado a função do IAM, você poderá criar o destino dos CloudWatch registros.

   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 ARN do novo destino que for retornado na carga útil, pois você usará isso como `destination.arn` em uma etapa posterior.

```
aws logs put-destination \

    --destination-name "testFirehoseDestination" \
    --target-arn "arn:aws:firehose:us-east-1:222222222222:deliverystream/my-
delivery-stream" \
    --role-arn "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole"

{
    "destination": {
```

```
        "destinationName": "testFirehoseDestination",
        "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
        "roleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisFirehoseRole",
        "arn": "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"}
}
```

b. Depois que a etapa anterior for concluída, na conta destinatária dos dados de log (222222222222), associe uma política de acesso ao destino. Essa política permite que a conta remetente dos dados de log (111111111111 conta) acesse o destino apenas na conta destinatária dos dados de log (222222222222). Você pode usar um editor de texto para colocar essa política no `~/AccessPolicy.json` arquivo:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

c. Isso cria uma política que define quem tem acesso de gravação ao destino. Essa política deve especificar as `logs:PutAccountPolicy` ações `logs:PutSubscriptionFilter` e para acessar o destino. Usuários de várias contas usarão as `PutAccountPolicy` ações `PutSubscriptionFilter` e para enviar eventos de registro ao destino.

```
aws logs put-destination-policy \
    --destination-name "testFirehoseDestination" \
    --access-policy file://~/AccessPolicy.json
```

Etapa 3: criar uma política de filtro de assinatura em nível de conta

Alterne para a conta de envio, que é 111111111111 neste exemplo. Agora você criará a política de filtro de assinatura em nível de conta na conta de envio. Neste exemplo, o filtro faz com que cada evento de log contendo a string ERR0R em todos os grupos de log, exceto dois, seja entregue ao destino que você criou anteriormente.

```
aws logs put-account-policy \
    --policy-name "CrossAccountFirehoseExamplePolicy" \
    --policy-type "SUBSCRIPTION_FILTER_POLICY" \
    --policy-document '{"DestinationArn":"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination", "FilterPattern":
 "{$.userIdentity.type = AssumedRole}", "Distribution": "Random"}' \
    --selection-criteria 'LogGroupName NOT IN ["LogGroupToExclude1",
 "LogGroupToExclude2"]' \
    --scope "ALL"
```

Os grupos de registros da conta de envio e o destino devem estar na mesma AWS região. No entanto, o destino pode apontar para um AWS recurso, como um stream do Firehose, 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 registro de entrada que correspondem ao padrão do filtro e aos critérios de seleção para o stream de entrega do Firehose. Os dados começam a aparecer em seu bucket do Amazon S3 com base no intervalo de tempo definido no stream de entrega do Firehose. Quando tiver passado tempo suficiente, você poderá conferir seus dados verificando o bucket do Amazon S3. Para verificar o bucket, insira este comando:

```
aws s3api list-objects --bucket 'firehose-test-bucket1'
```

A saída desse comando será semelhante a:

```
{
    "Contents": [
        {
            "Key": "2021/02/02/08/my-delivery-
stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba",
            "LastModified": "2023-02-02T09:00:26+00:00",
            "ETag": "\"EXAMPLEa817fb88fc770b81c8f990d\"",
```

```
            "Size": 198,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "firehose+2test",
                "ID": "EXAMPLE27fd05889c665d2636218451970ef79400e3d2aecca3adb1930042e0"
            }
        }
    ]
}
```

Em seguida, você pode recuperar um objeto específico do bucket digitando o comando a seguir. Substitua o valor de `key` pelo valor encontrado no comando anterior.

```
aws s3api get-object --bucket 'firehose-test-bucket1' --key '2021/02/02/08/my-delivery-stream-1-2021-02-02-08-55-24-5e6dc317-071b-45ba-a9d3-4805ba39c2ba' testfile.gz
```

Os dados no objeto do Amazon S3 são compactados com o formato gzip. É possível examinar os dados brutos na linha de comando usando os seguintes comandos:

Linux

```
zcat testfile.gz
```

macOS:

```
zcat <testfile.gz
```

Modificação da associação de destino no runtime

Pode haver situações em que você precise adicionar ou remover remetentes de log de um destino que você possua. Você pode usar as `PutAccountPolicy` ações PutDestinationPolicye em 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 333333333333 é habilitada.

1. Busque a política atualmente associada ao testDestination de destino e anote: AccessPolicy

   ```
   aws logs describe-destinations \
       --destination-name-prefix "testFirehoseDestination"
   ```

   Os dados retornados podem ter essa aparência.

```
{
    "destinations": [
        {
            "destinationName": "testFirehoseDestination",
            "targetArn": "arn:aws:firehose:us-east-1:222222222222:deliverystream/
my-delivery-stream",
            "roleArn": "arn:aws:iam:: 222222222222:role/CWLtoKinesisFirehoseRole",
            "accessPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement
\" : [\n    {\n      \"Sid\" : \"\",\n      \"Effect\" : \"Allow\",\n
 \"Principal\" : {\n        \"AWS\" : \"111111111111 \"\n      },\n      \"Action
\" : \"logs:PutSubscriptionFilter\",\n      \"Resource\" : \"arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination\"\n    }\n  ]\n}\n\n",
            "arn": "arn:aws:logs:us-east-1:
 222222222222:destination:testFirehoseDestination",
            "creationTime": 1612256124430
        }
    ]
}
```

2. Atualize a política para refletir que a conta 111111111111 foi interrompida e a conta 333333333333 está habilitada. Coloque essa política no arquivo NewAccessPolicy~/.json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "333333333333 "
      },
      "Action" : ["logs:PutSubscriptionFilter","logs:PutAccountPolicy"],
      "Resource" : "arn:aws:logs:us-
east-1:222222222222:destination:testFirehoseDestination"
    }
  ]
}
```

3. Use o comando a seguir para associar a política definida no NewAccessPolicyarquivo.json ao destino:

```
aws logs put-destination-policy \
```

```
    --destination-name "testFirehoseDestination" \

    --access-policy file://~/NewAccessPolicy.json
```

Por fim, isso desativa os eventos de log do ID da conta 111111111111. Os eventos de log da ID da conta 333333333333 começarão a fluir para o destino assim que o proprietário da conta 333333333333 criar um filtro de assinatura.

Prevenção de ‘confused deputy’

O problema de "confused deputy" é uma questão de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar em um problema confuso de delegado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com diretores de serviços que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição `aws:SourceOrgPaths`global `aws:SourceArn`aws:SourceAccountaws:SourceOrgID,,, e nas políticas de recursos para limitar as permissões que fornecem outro serviço ao recurso. Use `aws:SourceArn` se quiser associar apenas um recurso ao acesso entre serviços. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços. Use `aws:SourceOrgID` se quiser permitir que qualquer recurso de qualquer conta de uma organização seja associado ao uso entre serviços. Use `aws:SourceOrgPaths` se quiser associar qualquer recurso das contas em um caminho do AWS Organizations seja associado ao uso entre serviços. Para obter mais informações sobre como usar e entender os caminhos, consulte Compreender o caminho da AWS Organizations entidade.

A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou especificar vários recursos, use a chave de condição de contexto global `aws:SourceArn` com caracteres curinga (*) para as partes desconhecidas do ARN. Por exemplo, .`arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`

Se o valor do `aws:SourceArn` não contiver o ID da conta, como um ARN de bucket do Amazon S3, você deverá usar ambos, a `aws:SourceAccount` e o `aws:SourceArn` para limitar as permissões.

Para se proteger do problema de "confused deputy" em grande escala, use a chave de contexto de condição global `aws:SourceOrgID` ou `aws:SourceOrgPaths` com o ID ou o caminho da organização do recurso nas políticas baseadas em recursos. As políticas que incluem a chave `aws:SourceOrgID` ou `aws:SourceOrgPaths` incluem automaticamente as contas corretas e você não tem que atualizar manualmente as políticas quando adiciona, remove ou move contas na organização.

As políticas documentadas para conceder acesso ao CloudWatch Logs para gravar dados no Kinesis Data Streams e Etapa 1: criar um destino no Firehose Etapa 2: Criar um destino e mostram como você pode usar a chave de contexto de condição global awsSourceArn : para ajudar a evitar o confuso problema do deputado.

Prevenção de recursão de registros

Existe o risco de causar uma recursão infinita de registros com filtros de assinatura que pode levar a um grande aumento na cobrança por ingestão tanto nos CloudWatch registros quanto no seu destino, se não for evitada. Isso pode ocorrer quando um filtro de assinatura é associado a um grupo de registros que recebe eventos de registro como resultado do fluxo de trabalho de entrega da assinatura. Os registros ingeridos no grupo de registros serão entregues ao destino, fazendo com que o grupo de registros consuma mais registros, que serão encaminhados novamente para o destino, criando um loop de recursão.

Por exemplo, considere um filtro de assinatura com o destino como Firehose, que entrega eventos de log para o Amazon S3. Além disso, há também uma função Lambda que processa novos eventos entregues ao Amazon S3 e produz alguns registros por conta própria. Se o filtro de assinatura for aplicado ao grupo de registros da função Lambda, os eventos de log produzidos pela função serão encaminhados para o Firehose e o Amazon S3 no destino, que então invocarão a função novamente, fazendo com que mais registros sejam produzidos e encaminhados para o Firehose e o Amazon S3, causando outra invocação da função e assim por diante. Isso ocorrerá em um loop infinito, levando a um aumento inesperado no faturamento na ingestão de registros, no Firehose e no Amazon S3.

Se a função Lambda estiver anexada a uma VPC com registros de fluxo habilitados para Logs, o grupo de CloudWatch logs da VPC também poderá causar uma recursão de log.

Recomendamos que você não aplique filtros de assinatura a grupos de registros que fazem parte do seu fluxo de trabalho de entrega de assinaturas. Para filtros de assinatura em nível de conta, use o `selectionCriteria` parâmetro na `PutAccountPolicy` API para excluir esses grupos de registros da política.

Ao excluir grupos de registros, considere os seguintes AWS serviços que produzem registros e podem fazer parte de seus fluxos de trabalho de entrega de assinaturas:

- Amazon EC2 com Fargate
- Lambda
- AWS Step Functions
- Logs de fluxo da Amazon VPC que estão habilitados para Logs CloudWatch

> **Note**
>
> Eventos de registro produzidos pelo grupo de registros de um destino do Lambda não serão encaminhados de volta para a função do Lambda para uma política de filtro de assinatura em nível de conta. Nesse caso, não é necessário excluir o `selectionCriteria` uso do grupo de registros da função Lambda de destino para as políticas de assinatura da conta.

Sintaxe de padrões de filtros para filtros de métricas, filtros de assinatura, filtros de eventos de log e Live Tail.

> **Note**
>
> Para obter informações sobre como consultar seus grupos de CloudWatch logs com a linguagem de consulta Amazon Logs Insights, consulteCloudWatch Sintaxe de consulta do Logs Insights.

Com o CloudWatch Logs, você pode usar filtros métricos para transformar dados de log em métricas acionáveis, filtros de assinatura para encaminhar eventos de log para outros AWS serviços, filtrar eventos de log para pesquisar eventos de log e Live Tail para visualizar interativamente seus registros em tempo real à medida que são ingeridos.

Os padrões de filtros compõem a sintaxe que os filtros de métrica, filtros de assinatura, filtros de eventos de log e Live Tail usam para fazer a correspondência de termos em eventos de log. Os termos podem ser palavras, frases exatas ou valores numéricos. Expressões regulares (regex) podem ser usadas para criar padrões de filtros autônomos ou podem ser incorporadas a padrões de filtro JSON e delimitados por espaços.

Crie padrões de filtro com os termos que você deseja corresponder. Os padrões de filtro retornam apenas os eventos de log que contêm os termos que você definiu. Você pode testar padrões de filtro no CloudWatch console.

Tópicos

- Sintaxe de expressões regulares (regex) compatíveis
- Como usar padrões de filtros para fazer a correspondência de termos usando uma expressão regular (regex)
- Como usar padrões de filtros para fazer a correspondência de termos em eventos de log não estruturados
- Como usar padrões de filtros para fazer a correspondência de termos em eventos de log JSON
- Como usar padrões de filtros para fazer correspondência de termos em eventos de log delimitados por espaços

Sintaxe de expressões regulares (regex) compatíveis

Sintaxe de regex compatível

Ao usar regex para pesquisar e filtrar dados de log, é necessário envolver suas expressões com `%`.

Os padrões de filtros com regex só podem incluir o seguinte:

- Caracteres alfanuméricos: um caractere alfanumérico é um caractere que é uma letra (A a Z ou a a z) ou um dígito (0 a 9).
- Caracteres de símbolo compatíveis: "_", "#", "=", "@", "/", ";", "," e "-", entre outros. Por exemplo, `%something!%` seria rejeitado porque "!" não é aceito.
- Operadores compatíveis: "^", "$", "?", "[", "]", "{", "}", "|", "\", "*", "+", e ".", entre outros.

Os operadores (e) e não são aceitos. Não é possível usar parênteses para definir um subpadrão.

Caracteres multibytes não são aceitos.

> **Note**
>
> Cotas
> Há um máximo de 5 padrões de filtros contendo regex para cada grupo de logs na criação de filtros de métricas ou filtros de assinatura.
> Há um limite de dois regex para cada padrão de filtro na criação de um padrão de filtro delimitado ou JSON para filtros de métricas e filtros de assinatura, ou ao filtrar eventos de log ou Live Tail.

Uso de operadores compatíveis

- ^: fixa a correspondência no início de uma string. Por exemplo, `%^[hc]at%` corresponde a "hat" e "cat", mas somente no início de uma string.
- $: fixa a correspondência no final de uma string. Por exemplo, `%[hc]at$%` corresponde a "hat" e "cat", mas somente no final de uma string.
- ?: corresponde a zero ou mais instâncias do termo anterior. Por exemplo, `%colou?r%` pode corresponder a "color" e "colour".
- `[]`: define uma classe de caracteres. Corresponde à lista de caracteres ou ao intervalo de caracteres contido nos colchetes. Por exemplo, `%[abc]%` corresponde a "a", "b", ou "c"; `%[a-z]%`

corresponde a qualquer letra minúscula de "a" a "z"; e `%[abcx-z]%` corresponde a "a", "b", "c", "x", "y", ou "z".

- `{m, n}`: corresponde ao termo anterior no mínimo m e não mais que n vezes. Por exemplo, `%a{3,5}%` corresponde somente a "aaa", "aaaa", e "aaaaa".

> **Note**
>
> Tanto m quanto n podem ser omitidos se você optar por não definir um mínimo ou máximo.

- `|`: booleano "Or", que corresponde ao termo em ambos os lados da barra vertical. Por exemplo, `%gra|ey%` pode corresponder a "gray" ou "grey".

> **Note**
>
> Um termo é como um único caractere ou uma classe de caracteres repetidos que usa um dos seguintes operadores: `?`, `*`, `+`, ou `{n,m}`.

- `\`: caractere de escape que permite usar o significado literal de um operador em vez de seu significado especial. Por exemplo, `%\[.\]%` corresponde a qualquer caractere único cercado por "[" e "]", pois escape antecede os colchetes, como "[a]", "[b]", "[7]", "[@]", "[]]", e "[ ]".

> **Note**
>
> `%10\.10\.0\.1%` é a maneira correta de criar uma regex que corresponda ao endereço IP 10.10.0.1.

- `*`: corresponde a zero ou mais instâncias do termo anterior. Por exemplo, `%ab*c%` pode corresponder a "ac", "abc" e "abbbc"; `%ab[0-9]*%` pode corresponder a "ab", "ab0" e "ab129".
- `+`: corresponde a uma ou mais instâncias do termo anterior. Por exemplo, `%ab+c%` pode corresponder a "abc", "abbc" e "abbbc", mas não a "ac".
- `.`: corresponde a qualquer caractere sozinho. Por exemplo, `%.at%` corresponde a qualquer string de três caracteres que termina com "at", incluindo "hat", "cat", "bat", "4at", "#at" e " at" (começando com um espaço).

> **Note**
>
> Ao criar uma regex para fazer a correspondência de endereços IP, é importante colocar escape antes do operador `.`. Por exemplo, `%10.10.0.1%` pode corresponder a "10010,051", o que pode não ser o verdadeiro objetivo da expressão.

- `\d`, `\D`: corresponde a um caractere de dígito/não dígito. Por exemplo, `%\d%` é equivalente a `%[0-9]%` e `%\D%` é equivalente a `%[^0-9]%`.

> **Note**
>
> O operador maiúsculo indica o inverso de sua contraparte minúscula.

- `\s`, `\S`: corresponde a um caractere de espaço em branco/caractere sem espaço em branco.

> **Note**
>
> O operador maiúsculo indica o inverso de sua contraparte minúscula. Os caracteres de espaço em branco incluem os caracteres tab (`\t`), espaço ( ) e nova linha (`\n`).

- `\w`, `\W`: corresponde a um caractere alfanumérico/caractere não alfanumérico. Por exemplo, `%\w%` é equivalente a `%[a-zA-Z_0-9]%` e `%\W%` é equivalente a `%[^a-zA-Z_0-9]%`.

> **Note**
>
> O operador maiúsculo indica o inverso de sua contraparte minúscula.

- `\xhh`: corresponde ao mapeamento ASCII para um caractere hexadecimal de dois dígitos. `\x` é a sequência de escape que indica que os caracteres a seguir representam o valor hexadecimal para ASCII. hh especifica os dois dígitos hexadecimais (0-9 e A-F) que apontam para um caractere na tabela ASCII.

> **Note**
>
> É possível usar `\xhh` para combinar caracteres de símbolo que não são compatíveis com o padrão de filtros. Por exemplo, `%\x3A%` corresponde a `:`; e `%\x28%` corresponde a `(`.

Como usar padrões de filtros para fazer a correspondência de termos usando uma expressão regular (regex)

Fazer a correspondência de termos usando regex

É possível fazer a correspondência de termos em seus eventos de log usando um padrão regex cercado por % (sinais de percentual antes e depois do padrão regex). O trecho de código a seguir mostra um exemplo de um padrão de filtro que retorna todos os eventos de log que consistem na palavra-chave AUTORIZADO.

Para obter uma lista de expressões regulares compatíveis, consulte Expressões regulares aceitas.

```
%AUTHORIZED%
```

O padrão de filtro retorna mensagens de eventos de log da seguinte forma:

- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[SUCCESS 200] AUTHORIZED REQUEST`

Como usar padrões de filtros para fazer a correspondência de termos em eventos de log não estruturados

Fazer a correspondência de termos em eventos de log não estruturados

Os exemplos a seguir contêm trechos de código que mostram como é possível usar padrões de filtros para fazer a correspondência de termos em seus eventos de log não estruturados.

> **Note**
>
> Os padrões de filtro diferenciam letras maiúsculas de minúsculas. Coloque frases exatas e termos que incluam caracteres não alfanuméricos entre aspas duplas ("").

Example: Match a single term

O trecho de código a seguir mostra um exemplo de um padrão de filtro de termo único que retorna todos os eventos de log em que as mensagens contêm a palavra ERROR (ERRO).

```
ERROR
```

Este padrão de filtro procura mensagens de eventos de log da seguinte forma:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match multiple terms

O trecho de código a seguir mostra um exemplo de um padrão de filtro de termos múltiplos que retorna todos os eventos de log em que as mensagens contêm as palavras ERRO e ARGUMENTOS.

```
ERROR ARGUMENTS
```

O filtro retorna mensagens de eventos de log da seguinte forma:

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

O padrão de filtro não retorna as seguintes mensagens de eventos de log porque elas não contêm os dois termos especificados no padrão do filtro.

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

Example: Match optional terms

É possível usar a correspondência de padrões para criar padrões de filtros que retornem eventos de log contendo termos opcionais. Coloque um ponto de interrogação ("?") antes dos termos que você deseja corresponder. O trecho de código a seguir mostra um exemplo de um padrão de filtro que retorna todos os eventos de log em que as mensagens contêm a palavra ERRO ou ARGUMENTOS.

```
?ERROR ?ARGUMENTS
```

Este padrão de filtro procura mensagens de eventos de log da seguinte forma:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`
- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

> **Note**
>
> Não é possível combinar o ponto de interrogação (“?”) com outros padrões de filtro, como termos de inclusão e exclusão. Se você combinar “?” com outros padrões de filtro, o ponto de interrogação (“?”) será ignorado.
> Por exemplo, o padrão de filtro a seguir corresponde a todos os eventos que contêm a palavra `REQUEST`, mas o filtro de ponto de interrogação (“?”) é ignorado e não tem efeito.
>
> ```
> ?ERROR ?ARGUMENTS REQUEST
> ```
>
> Correspondências de eventos de log
>
> - `[INFO] REQUEST FAILED`
> - `[WARN] UNAUTHORIZED REQUEST`
> - `[ERROR] 400 BAD REQUEST`

Example: Match exact phrases

O trecho de código a seguir mostra um exemplo de um padrão de filtro que retorna os eventos de log nos quais as mensagens contêm a frase ERRO INTERNO DO SERVIDOR.

```
"INTERNAL SERVER ERROR"
```

O padrão de filtro retorna a seguinte mensagem de eventos de log:

- `[ERROR 500] INTERNAL SERVER ERROR`

Example: Include and exclude terms

Você pode criar padrões de filtro que retornem eventos de log nos quais as mensagens incluam alguns termos e excluam outros. Coloque um símbolo de menos ("-") antes dos termos que você deseja excluir da busca. O trecho de código a seguir mostra um exemplo de um padrão de filtro que retorna eventos de log onde as mensagens incluem o termo ERROR e não incluem o termo ARGUMENTS.

```
ERROR -ARGUMENTS
```

O padrão de filtro retorna mensagens de eventos de log da seguinte forma:

- `[ERROR 400] BAD REQUEST`
- `[ERROR 401] UNAUTHORIZED REQUEST`

O padrão de filtro não retorna as mensagens de eventos de log a seguir porque elas contêm a palavra ARGUMENTS.

- `[ERROR 419] MISSING ARGUMENTS`
- `[ERROR 420] INVALID ARGUMENTS`

Example: Match everything

É possível corresponder a tudo nos eventos de log usando aspas duplas. O trecho de código a seguir mostra um exemplo de um padrão de filtro que retorna todos os eventos de log.

```
" "
```

Como usar padrões de filtros para fazer a correspondência de termos em eventos de log JSON

Escrever padrões de filtros para eventos de log JSON

Os exemplos a seguir descrevem a sintaxe para padrões de filtros que correspondem a termos JSON contendo strings e valores numéricos.

Writing filter patterns that match strings

É possível criar padrões de filtros para corresponder a strings em eventos de log JSON. O trecho de código a seguir mostra um exemplo de sintaxe de padrão de filtro baseado em string.

```
{ PropertySelector EqualityOperator String }
```

Coloque padrões de filtros entre chaves ("{}"). Padrões de filtros baseados em string devem conter as seguintes partes:

- Seletor de propriedades

  Defina seletores de propriedades com um cifrão seguido de um ponto ("$."). Os seletores de propriedade são strings alfanuméricas compatíveis com os caracteres de hífen ("-") e underline ("_"). Strings não são compatíveis com notação científica. Seletores de propriedades apontam para nós de valor em eventos de log JSON. Os nós de valor podem ser strings ou números. Coloque matrizes após seletores de propriedades. Os elementos nas matrizes seguem um sistema de numeração baseado em zero, o que significa que o primeiro elemento na matriz é o elemento 0, o segundo elemento é o elemento 1 e assim por diante. Coloque elementos entre

colchetes ("[]"). Se um seletor de propriedades apontar para uma matriz ou um objeto, o padrão de filtro não corresponderá ao formato do log. Se a propriedade JSON contiver um ponto (" . "), a notação de colchetes poderá ser usada para selecionar essa propriedade.

> **Note**
>
> Seletor de curingas
> É possível usar o curinga JSON para selecionar qualquer elemento da matriz ou qualquer campo de objeto JSON.
> Cotas
> É possível usar no máximo um seletor de curinga em um seletor de propriedades.

- Operador de igualdade

  Defina operadores de igualdade com um dos seguintes símbolos: igual ("=") ou diferente ("!="). Os operadores de igualdade retornam um valor booliano (true ou false).

- String

  Você pode colocar strings entre aspas duplas (""). Strings que contenham tipos que não sejam caracteres alfanuméricos ou underline devem ser colocadas entre aspas duplas. Use o asterisco ("*") como curinga para corresponder ao texto.

> **Note**
>
> É possível usar qualquer expressão regular condicional ao criar padrões de filtros para fazer a correspondência com termos em eventos de log JSON. Para obter uma lista de expressões regulares compatíveis, consulte Expressões regulares aceitas.

O trecho de código a seguir contém um exemplo de um padrão de filtro que mostra como formatar um padrão de filtro para corresponder um termo JSON a uma string.

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

Writing filter patterns that match numeric values

É possível criar padrões de filtros para fazer a correspondência de valores numéricos em eventos de log JSON. O seguinte trecho de código mostra um exemplo de sintaxe para padrões de filtros que correspondem a valores numéricos.

```
{ PropertySelector NumericOperator Number }
```

Coloque padrões de filtros entre chaves ("{}"). Os padrões de filtros que correspondem a valores numéricos devem ter as seguintes partes:

- Seletor de propriedades

  Defina seletores de propriedades com um cifrão seguido de um ponto ("$."). Os seletores de propriedade são strings alfanuméricas compatíveis com os caracteres de hífen ("-") e underline ("_"). Strings não são compatíveis com notação científica. Seletores de propriedades apontam para nós de valor em eventos de log JSON. Os nós de valor podem ser strings ou números. Coloque matrizes após seletores de propriedades. Os elementos nas matrizes seguem um sistema de numeração baseado em zero, o que significa que o primeiro elemento na matriz é o elemento 0, o segundo elemento é o elemento 1 e assim por diante. Coloque elementos entre colchetes ("[]"). Se um seletor de propriedades apontar para uma matriz ou um objeto, o padrão de filtro não corresponderá ao formato do log. Se a propriedade JSON contiver um ponto ("."), a notação de colchetes poderá ser usada para selecionar essa propriedade.

  > **Note**
  >
  > Seletor de curingas
  > É possível usar o curinga JSON para selecionar qualquer elemento da matriz ou qualquer campo de objeto JSON.
  > Cotas
  > É possível usar no máximo um seletor de curinga em um seletor de propriedades.

- Operador numérico

  Defina operadores numéricos com um dos seguintes símbolos: maior que (">"), menor que ("<"), igual a ("="), diferente de ("! ="), maior ou igual a (">="), ou menor ou igual a ("<=").

- Número

Você pode usar números inteiros que contenham os símbolos de mais ("+") ou menos ("-") e seguir a notação científica. Use o asterisco ("*") como curinga para corresponder números.

O trecho de código a seguir contém exemplos que mostram como formatar padrões de filtros para fazer a correspondência de termos JSON com valores numéricos.

```
// Filter pattern with greater than symbol
{ $.bandwidth > 75 }
// Filter pattern with less than symbol
{ $.latency < 50 }
// Filter pattern with greater than or equal to symbol
{ $.refreshRate >= 60 }
// Filter pattern with less than or equal to symbol
{ $.responseTime <= 5 }
// Filter pattern with equal sign
{ $.errorCode = 400}
// Filter pattern with not equal sign
{ $.errorCode != 500 }
// Filter pattern with scientific notation and plus symbol
{ $.number[0] = 1e-3 }
// Filter pattern with scientific notation and minus symbol
{ $.number[0] != 1e+3 }
```

Fazer a correspondência de termos em eventos de log JSON usando expressões simples

Os exemplos a seguir contêm trechos de código que mostram como padrões de filtros podem fazer a correspondência de termos em um evento de log JSON.

> **Note**
>
> Se você testar um padrão de filtro de exemplo com o evento de log JSON de exemplo, deverá inserir o log JSON de exemplo em uma única linha.

Evento de log JSON

```
{
```

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

Example: Filter pattern that matches string values

Esse padrão de filtro corresponde à string `"UpdateTrail"` na propriedade `"eventType"`.

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

Example: Filter pattern that matches string values (IP address)

O padrão de filtro contém um curinga e corresponde à propriedade `"sourceIPAddress"` porque ele não contém um número com o prefixo `"123.123."`.

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

Example: Filter pattern that matches a specific array element with a string value

Esse padrão de filtro corresponde ao elemento `"value"` na matriz `"arrayKey"`.

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

Example: Filter pattern that matches a string using regex

Esse padrão de filtro corresponde à string `"Trail"` na propriedade `"eventType"`.

```
{ $.eventType = %Trail% }
```

Example: Filter pattern that uses a wildcard to match values of any element in the array using regex

O padrão de filtro contém uma regex que corresponde ao elemento `"value"` na matriz `"arrayKey"`.

```
{ $.arrayKey[*] = %val.{2}% }
```

Example: Filter pattern that uses a wildcard to match values of any element with a specific prefix and subnet using regex (IP address)

O padrão de filtro contém uma regex que corresponde ao elemento `"111.111.111.111"` na propriedade `"sourceIPAddress"`.

```
{ $.* = %111\.111\.111\.1[0-9]{1,2}% }
```

> **Note**
>
> Cotas
> É possível usar no máximo um seletor de curinga em um seletor de propriedades.

Example: Filter pattern that matches a JSON property with a period (.) in the key

```
{ $.['cluster.name'] = "c" }
```

Example: Filter pattern that matches JSON logs using IS

É possível criar padrões de filtros que correspondam a campos em logs JSON com a variável `IS`. A variável `IS` pode corresponder campos que contenham os valores `NULL`, `TRUE` ou `FALSE`. O padrão de filtro a seguir retorna logs JSON para os quais o valor de `SomeObject` é `NULL`.

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

Example: Filter pattern that matches JSON logs using NOT EXISTS

Você pode criar padrões de filtro com a `NOT EXISTS` variável para retornar registros JSON que não contêm campos específicos nos dados do registro. O padrão de filtro a seguir usa `NOT EXISTS` para retornar logs JSON que não contenham o campo `SomeOtherObject`.

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

> **Note**
>
> No momento, não há compatibilidade com as variáveis `IS NOT` e `EXISTS`.

Fazer a correspondência de termos em objetos JSON usando expressões compostas

É possível usar os operadores lógicos AND ("&&") e OR ("||") em padrões de filtros para criar expressões compostas que correspondam a eventos de log nos quais duas ou mais condições são verdadeiras. Expressões compostas são compatíveis com o uso de parênteses ("()") e com a seguinte ordem padrão de operações: () > && > ||. Os exemplos a seguir contêm trechos de código que mostram como usar padrões de filtros com expressões compostas para corresponder a termos em um objeto JSON.

Objeto JSON

```
{
    "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]
    ]
}
```

Example: Expression that matches using AND (&&)

Este padrão de filtro contém uma expressão composta que faz a correspondência de `"id"` em `"user"` com um valor numérico de 1 e de `"email"` no primeiro elemento da matriz `"users"` com a string `"John.Doe@example.com"`.

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

Example: Expression that matches using OR (||)

Esse padrão de filtro contém uma expressão composta que faz a correspondência de `"email"` em `"user"` com a string `"John.Stiles@example.com"`.

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

Example: Expression that doesn't match using AND (&&)

Esse padrão de filtro contém uma expressão composta que não encontra uma correspondência porque a expressão não corresponde à terceira ação em `"actions"`.

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

> **Note**
>
> Cotas
> É possível usar no máximo um seletor de curinga em um seletor de propriedades e até três seletores de curinga em um padrão de filtro com expressões compostas.

Example: Expression that doesn't match using OR (||)

O padrão de filtro contém uma expressão composta que não encontra uma correspondência porque a expressão não corresponde à primeira propriedade em `"users"` ou à terceira ação em `"actions"`.

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

Como usar padrões de filtros para fazer correspondência de termos em eventos de log delimitados por espaços

Escrever padrões de filtros para eventos de log delimitados por espaços

É possível criar padrões de filtros para fazer a correspondência de termos em eventos de log delimitados por espaços Veja a seguir um exemplo de evento de log delimitado por espaços e uma descrição de como escrever a sintaxe para padrões de filtros que correspondam aos termos do evento de log delimitado por espaços.

> **Note**
>
> É possível usar qualquer expressão regular condicional ao criar padrões de filtros para fazer a correspondência de termos em eventos de log delimitados por espaços. Para obter uma lista de expressões regulares compatíveis, consulte Expressões regulares aceitas.

Example: Space-delimited log event

O trecho de código a seguir mostra um evento de log delimitado por espaço que contém sete campos: `ip`, `user`, `username`, `timestamp`, `request`, `status_code` e `bytes`.

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

> **Note**
>
> Caracteres entre colchetes ("[]") e entre aspas duplas ("") são considerados campos únicos.

Writing filter patterns that match terms in a space-delimited log event

Para criar um padrão de filtro que corresponda a termos em um evento de log delimitado por espaços, coloque o padrão de filtro entre colchetes ("[]") e especifique campos com nomes separados por vírgulas (","). O padrão de filtro a seguir analisa sete campos.

```
[ip=%127\.0\.0\.[1-9]%, user, username, timestamp, request =*.html*, status_code =
 4*, bytes]
```

É possível usar operadores numéricos (**>**, **<**, **=**, **!=**, **>=** ou **<=**) e o asterisco (*) como um curinga ou uma regex para fornecer condições de padrão de filtro. No exemplo de padrão de filtro, `ip` usa uma regex que corresponde ao intervalo de endereços IP 127.0.0.1 a 127.0.0.9, `request` contém um curinga que indica que ele deve extrair um valor com `.html` e `status_code` contém um curinga que afirma que deve extrair um valor começando com `4`.

Se não souber o número de campos que está analisando em um evento de log delimitado por espaço, você poderá usar reticências (…) para fazer referência a qualquer campo sem nome. As reticências podem fazer referência a quantos campos forem necessários. O exemplo a seguir mostra um padrão de filtro com reticências que representam os primeiros quatro campos sem nome mostrados no padrão de filtro de exemplo anterior.

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

Você também pode usar os operadores lógicos AND (&&) e OR (||) para criar expressões compostas. O padrão de filtro a seguir contém uma expressão composta que indica que o valor de `status_code` deve ser `404` ou `410`.

```
[ip, user, username, timestamp, request =*.html*, status_code = 404 || status_code =
 410, bytes]
```

Fazer a correspondência de termos em eventos de log delimitados por espaços usando correspondência de padrões

É possível usar a correspondência de padrões para criar padrões de filtros delimitados por espaços que correspondam a termos em uma ordem específica. Especifique a ordem de seus termos com indicadores. Use w1 para representar seu primeiro termo e w2 e assim por diante para representar a ordem dos termos subsequentes. Coloque vírgulas (",") entre os termos. Os exemplos a seguir contêm trechos de código que mostram como é possível usar a correspondência de padrões com padrões de filtros delimitados por espaços.

> **Note**
>
> É possível usar qualquer expressão regular condicional ao criar padrões de filtros para fazer a correspondência de termos em eventos de log delimitados por espaços. Para obter uma lista de expressões regulares compatíveis, consulte Expressões regulares aceitas.

Evento de log delimitado por espaços

```
INFO 09/25/2014 12:00:00 GET /service/resource/67 1200
INFO 09/25/2014 12:00:01 POST /service/resource/67/part/111 1310
WARNING 09/25/2014 12:00:02 Invalid user request
ERROR 09/25/2014 12:00:02 Failed to process request
```

Example: Match terms in order

O padrão de filtro delimitado por espaços retorna eventos de log em que a primeira palavra nos eventos de log é ERROR.

```
[w1=ERROR, w2]
```

> **Note**
>
> Ao criar filtros de métrica delimitados por espaços que usam correspondência de padrões, é necessário incluir um indicador em branco depois de especificar a ordem dos termos.

Por exemplo, se você criar um padrão de filtro que retorne eventos de log onde a primeira palavra seja ERROR, inclua um indicador w2 em branco após o termo w1.

Example: Match terms with AND (&&) and OR (||)

É possível usar os operadores lógicos AND ("&&") e OR ("||") para criar padrões de filtros delimitados por espaços que contenham condições. O padrão de filtro a seguir retorna eventos de log onde a primeira palavra nos eventos é ERROR ou WARNING.

```
[w1=ERROR || w1=WARNING, w2]
```

Example: Exclude terms from matches

É possível criar padrões de filtros delimitados por espaços que retornam eventos de log excluindo um ou mais termos. Coloque um símbolo de diferente ("-") antes do termo ou dos termos que você deseja excluir. O trecho de código a seguir mostra um exemplo de um padrão de filtro que retorna eventos de log em que as primeiras palavras não são ERROR e WARNING.

```
[w1!=ERROR && w1!=WARNING, w2]
```

Example: Match the top level item in a resource URI

O trecho de código a seguir mostra um exemplo de um padrão de filtro que corresponde ao item de nível superior em um URI de recurso usando regex.

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+$%, response_time]
```

Example: Match the child level item in a resource URI

O trecho de código a seguir mostra um exemplo de um padrão de filtro que corresponde ao item de nível filho em um URI de recurso usando regex.

```
[logLevel, date, time, method, url=%/service/resource/[0-9]+/part/[0-9]+$%,
 response_time]
```

Habilitar o registro a partir de AWS serviços

Embora muitos serviços publiquem registros somente no CloudWatch Logs, alguns AWS serviços podem publicar registros diretamente no Amazon Simple Storage Service ou no Amazon Data Firehose. Se seu principal requisito para registros for armazenamento ou processamento em um desses serviços, você pode facilmente fazer com que o serviço que produz os registros os envie diretamente para o Amazon S3 ou Firehose sem configuração adicional.

Mesmo quando os registros são publicados diretamente no Amazon S3 ou no Firehose, cobranças são aplicadas. Para obter mais informações, consulte Vended Logs na guia Logs em Amazon CloudWatch Pricing.

Alguns AWS serviços usam uma infraestrutura comum para enviar seus registros. Para ativar o log desses produtos, você deve estar registrado como um usuário com certas permissões. Além disso, você deve conceder permissões AWS para permitir que os registros sejam enviados.

Para serviços que exijam essas permissões, há duas versões das permissões necessárias. Os produtos que exigem essas permissões extras são indicados como [Permissões v1] compatíveis e [Permissões v2] compatíveis na tabela. Para obter informações sobre essas permissões necessárias, consulte as seções após a tabela.

Tipo de log	CloudWatch Logs	Amazon S3	Firehose
Logs de acesso do Amazon API Gateway	[Permissões v1] compatíveis		
AWS AppSync logs	Compatível		
Logs da Amazon Aurora MySQL	Compatível		
Amazon Bedrock Registro de bases de conhecimento	[Permissões v2] compatíveis	[Permissões v2] compatíveis	[Permissões v2] compatíveis

Tipo de log	CloudWatch Logs	Amazon S3	Firehose
Logs de métricas de qualidade de mídia do Amazon Chime e logs de mensagens SIP	[Permissões v1] compatíveis		
CloudFront: registros de acesso		[Permissões v1] compatíveis	
AWS CloudHSM registros de auditoria	Compatível		
CloudWatch Evidentemente, os registros de eventos de avaliação	[Permissões v1] compatíveis	[Permissões v1] compatíveis	
CloudWatch Registros do Internet Monitor		[Permissões v1] compatíveis	
CloudTrail troncos	Compatível		
AWS CodeBuild logs	Compatível		
Amazon CodeWhisperer registros de eventos	[Permissões v2] compatíveis	[Permissões v2] compatíveis	[Permissões v2] compatíveis
Amazon Cognito logs	[Permissões v1] compatíveis		
Logs do Amazon Connect	Compatível		
AWS DataSync logs	Compatível		

Tipo de log	CloudWatch Logs	Amazon S3	Firehose
Registros da Amazon ElastiCache para Redis	[Permissões v1] compatíveis		[Permissões v1] compatíveis
AWS Elastic Beanstalk logs	Compatível		
Logs do Amazon Elastic Container Service	Compatível		
Logs do ambiente de gerenciamento do Serviço Amazon Elastic Kubernetes	Compatível		
Amazon EventBridge Registro de tubulações	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
AWS Fargate logs	Compatível		
AWS Fault Injection Service registros de experimentos		[Permissões v1] compatíveis	
Amazon FinSpace	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
AWS Global Accelerator registros de fluxo		[Permissões v1] compatíveis	
AWS Glue registros de trabalho	Compatível		
Registros de erros do IAM Identity Center	[Permissões v2] compatíveis	[Permissões v2] compatíveis	[Permissões v2] compatíveis

Tipo de log	CloudWatch Logs	Amazon S3	Firehose
Logs de chat do Amazon Interactive Video Service	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
AWS IoT logs	Compatível		
AWS IoT FleetWise logs	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
AWS Lambda logs	Compatível		
Logs do Amazon Macie	Compatível		
AWS Mainframe Modernization	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
Logs do Amazon Managed Service for Prometheus	[Permissões v1] compatíveis		
Logs do agente do Amazon MSK	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
Logs do Amazon MSK Connect	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
Logs gerais e de auditoria do Amazon MQ	Compatível		
AWS Registros do Firewall de Rede	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis

Tipo de log	CloudWatch Logs	Amazon S3	Firehose
Logs de acesso do Network Load Balancer		[Permissões v1] compatíveis	
OpenSearch troncos	Compatível		
Registros OpenSearch de ingestão do Amazon Service	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
AWS OpsWorks logs	Compatível		
Registros ServicePostgre SQL do Amazon Relational Database	Compatível		
AWS RoboMaker troncos	Compatível		
Logs de consulta ao DNS público do Amazon Route 53	Compatível		
Logs de consulta do Amazon Route 53 Resolver	[Permissões v1] compatíveis	[Permissões v1] compatíveis	
SageMaker Eventos da Amazon	[Permissões v1] compatíveis		
Eventos para SageMaker trabalhadores da Amazon	[Permissões v1] compatíveis		
AWS Registros de VPN de site para site	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis

Tipo de log	CloudWatch Logs	Amazon S3	Firehose
Logs do Amazon Simple Notification Service	Compatível		
Logs da política de proteção de dados do Amazon Simple Notification Service	Compatível		
Arquivos do feed de dados da instância spot do EC2		[Permissões v1] compatíveis	
AWS Step Functions Registros de fluxo de trabalho expresso e fluxo de trabalho padrão	[Permissões v1] compatíveis		
Logs de auditoria e logs de integridade do Storage Gateway	[Permissões v1] compatíveis		
AWS Transfer Family logs	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
Acesso Verificado pela AWS logs	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
Logs de fluxo da Amazon Virtual Private Cloud	Compatível	[Permissões v1] compatíveis	[Permissões v1] compatíveis
Logs de acesso do Amazon VPC Lattice	[Permissões v1] compatíveis	[Permissões v1] compatíveis	[Permissões v1] compatíveis
AWS WAF logs	[Permissões v1] compatíveis	[Permissões v1] compatíveis	Compatível

Tipo de log	CloudWatch Logs	Amazon S3	Firehose
Amazon WorkMail troncos	[Permissões v2] compatíveis	[Permissões v2] compatíveis	[Permissões v2] compatíveis

Registro em log que requer permissões [v1] adicionais

Alguns AWS serviços usam uma infraestrutura comum para enviar seus CloudWatch registros para Logs, Amazon S3 ou Firehose. Para habilitar os serviços da AWS listados na tabela a seguir para enviar seus logs para esses destinos, você deve estar conectado como um usuário com determinadas permissões.

Além disso, é necessário conceder permissões AWS para permitir que os registros sejam enviados. AWS pode criar automaticamente essas permissões quando os registros são configurados, ou você mesmo pode criá-las antes de configurar o registro. Para a entrega entre contas, você mesmo deve criar manualmente as políticas de permissão.

Se você optar por configurar AWS automaticamente as permissões e as políticas de recursos necessárias quando você ou alguém em sua organização configura o envio de registros pela primeira vez, o usuário que está configurando o envio de registros deverá ter determinadas permissões, conforme explicado posteriormente nesta seção. Se preferir, você pode criar as políticas de recursos, e os usuários que configurarem o envio de logs não precisarão de tantas permissões.

A tabela a seguir resume a quais tipos de logs e a quais destinos de log as informações nesta seção se aplicam.

As seções a seguir fornecem mais detalhes sobre cada um desses destinos.

Registros enviados para CloudWatch Logs

> **Important**
>
> Quando você configura os tipos de registro na lista a seguir para serem enviados para o CloudWatch Logs, AWS cria ou altera as políticas de recursos associadas ao grupo de registros que recebe os registros, se necessário. Continue lendo esta seção para ver os detalhes.

Esta seção se aplica quando os tipos de registros listados na tabela da seção anterior são enviados para CloudWatch Logs:

Permissões de usuário

Para poder configurar o envio de qualquer um desses tipos de CloudWatch registros para o Logs pela primeira vez, você precisa estar conectado a uma conta com as seguintes permissões.

- `logs:CreateLogDelivery`
- `logs:PutResourcePolicy`
- `logs:DescribeResourcePolicies`
- `logs:DescribeLogGroups`

> **Note**
>
> Ao especificar a `logs:PutResourcePolicy` permissão`logs:DescribeLogGroups`, ou`logs:DescribeResourcePolicies`, certifique-se de definir o ARN de sua `Resource` linha para usar um `*` caractere curinga, em vez de especificar apenas um único nome de grupo de registros. Por exemplo, `"Resource": "arn:aws:logs:us-east-1:111122223333:log-group:*"`.

Se algum desses tipos de registros já estiver sendo enviado para um grupo de CloudWatch registros no Logs, para configurar o envio de outro desses tipos de registros para esse mesmo grupo de registros, você só precisará da `logs:CreateLogDelivery` permissão.

Política de recursos do grupo de logs

O grupo de logs para o qual os logs estão sendo enviados deve ter uma política de recursos que contenha determinadas permissões. Se o grupo de registros atualmente não tiver uma política de recursos e o usuário que configura o registro tiver as `logs:DescribeLogGroups` permissões`logs:PutResourcePolicy`,`logs:DescribeResourcePolicies`, e para o grupo de registros, AWS criará automaticamente a política a seguir quando você começar a enviar os CloudWatch registros para o 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:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
      }
    }
  ]
}
```

Se o grupo de logs tem uma política de recursos, mas essa política não contém a instrução exibida na política anterior, e o usuário configurando o log tem as permissões `logs:PutResourcePolicy`, `logs:DescribeResourcePolicies` e `logs:DescribeLogGroups` para o grupo de logs, essa instrução é anexada à política de recursos do grupo de logs.

Considerações sobre o limite do tamanho da política de recursos do grupo de logs

Esses serviços devem listar cada grupo de registros para o qual estão enviando registros na política de recursos, e as políticas de recursos de CloudWatch registros estão limitadas a 5120 caracteres. Um serviço que envia registros para um grande número de grupos de registros pode atingir esse limite.

Para mitigar isso, o CloudWatch Logs monitora o tamanho das políticas de recursos usadas pelo serviço que está enviando registros e, quando detecta que uma política se aproxima do limite de tamanho de 5120 caracteres, o CloudWatch Logs ativa `/aws/vendedlogs/*` automaticamente a

política de recursos desse serviço. Depois, você pode começar a usar grupos de logs com nomes que começam com `/aws/vendedlogs/` como destinos para os logs desses serviços.

Logs enviados ao Amazon S3

Quando você define os registros a serem enviados para o Amazon S3, AWS cria ou altera as políticas de recursos associadas ao bucket do S3 que está recebendo os registros, se necessário.

Os logs publicados diretamente no 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.

Quando você entrega logs a um bucket do Amazon S3 pela primeira vez, o serviço que entrega logs registra o proprietário do bucket para garantir que os logs sejam entregues somente a um bucket pertencente a essa conta. Consequentemente, para alterar o proprietário do bucket do Amazon S3, é necessário recriar ou atualizar a assinatura de log no serviço de origem.

> **Note**
>
> CloudFront usa um modelo de permissões diferente dos outros serviços que enviam registros vendidos para o S3. Para obter mais informações, consulte Permissões necessárias para configurar o registro padrão e acessar seus arquivos de log.
> Além disso, se você usar o mesmo bucket do S3 para registros de CloudFront acesso e outra fonte de log, habilitar a ACL no bucket CloudFront também concederá permissão a todas as outras fontes de log que usam esse bucket.

Permissões de usuário

Para poder configurar o envio de qualquer um desses tipos de logs ao Amazon S3 pela primeira vez, é necessário conectar-se a uma conta com as permissões a seguir.

- `logs:CreateLogDelivery`
- `S3:GetBucketPolicy`
- `S3:PutBucketPolicy`

Se algum 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 esse mesmo bucket, apenas a permissão `logs:CreateLogDelivery` será necessária.

Política de recursos do bucket do S3

O bucket do S3 para o qual os logs estão sendo enviados deve ter uma política de recursos que contenha determinadas permissões. Se o bucket atualmente não tiver uma política de recursos e o usuário que configura o registro tiver as `S3:PutBucketPolicy` permissões `S3:GetBucketPolicy` e para o bucket, criará AWS automaticamente a seguinte política para ele quando você começar a enviar os registros 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",
            "Condition": {
                "StringEquals": {
                "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
                }
            }
        },
        {
            "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",
                    "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
```

```
                    "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
                }
            }
        }
    ]
}
```

Na política anterior, para `aws:SourceAccount`, especifique a lista de IDS de conta para os quais os logs estão sendo entregues a esse bucket. Para `aws:SourceArn`, especifique a lista de ARNs do recurso que gera os logs, no formulário `arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Se o bucket tiver uma política de recursos, mas ela não contiver a instrução exibida na política anterior, e o usuário configurando o log tiver as permissões `S3:GetBucketPolicy` e `S3:PutBucketPolicy` para o bucket, essa instrução será anexada à política de recursos do bucket.

> **Note**
>
> Em alguns casos, você pode ver `AccessDenied` erros AWS CloudTrail se a `s3:ListBucket` permissão não tiver sido concedida`delivery.logs.amazonaws.com`. Para evitar esses erros em seus CloudTrail registros, você deve conceder a `s3:ListBucket` permissão `delivery.logs.amazonaws.com` e incluir `Condition` os parâmetros mostrados com o conjunto de `s3:GetBucketAcl` permissões na política de bucket anterior. Para simplificar isso, em vez de criar uma nova `Statement`, você pode atualizar `AWSLogDeliveryAclCheck` diretamente para `“Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]`

Criptografia no lado do servidor de bucket do Amazon S3

Você pode proteger os dados em seu bucket do Amazon S3 habilitando a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) ou a criptografia do lado do servidor com uma chave armazenada em (SSE-KMS). AWS KMS AWS Key Management Service Para obter mais informações, consulte Proteger dados usando a criptografia no lado do servidor.

Se você escolher SSE-S3, nenhuma configuração adicional será necessária. O Amazon S3 lida com a chave de criptografia.

> **Warning**
>
> Se você escolher o SSE-KMS, deverá usar uma chave gerenciada pelo cliente, pois o uso de uma chave AWS gerenciada não é suportado nesse cenário. Se você configurar a criptografia usando uma chave AWS gerenciada, os registros serão entregues em um formato ilegível.

Ao usar uma AWS KMS chave gerenciada pelo cliente, você pode especificar o Amazon Resource Name (ARN) da chave gerenciada pelo cliente ao ativar a criptografia do bucket. Você precisa adicionar o seguinte à política de chaves da chave gerenciada pelo cliente (não à política de bucket para o bucket do S3), para que a conta de entrega de log possa gravar no bucket do S3.

Se você escolher o SSE-KMS, deverá usar uma chave gerenciada pelo cliente, pois o uso de uma chave AWS gerenciada não é suportado nesse cenário. Ao usar uma AWS KMS chave gerenciada pelo cliente, você pode especificar o Amazon Resource Name (ARN) da chave gerenciada pelo cliente ao ativar a criptografia do bucket. Você precisa adicionar o seguinte à política de chaves da chave gerenciada pelo cliente (não à política de bucket para o bucket do S3), para que a conta de entrega de log possa gravar no bucket do S3.

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
```

```
        }
}
```

Para `aws:SourceAccount`, especifique a lista de IDS de conta para os quais os logs estão sendo entregues a esse bucket. Para `aws:SourceArn`, especifique a lista de ARNs do recurso que gera os logs, no formulário `arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Registros enviados para o Firehose

Esta seção se aplica quando os tipos de registros listados na tabela da seção anterior são enviados para o Firehose:

Permissões de usuário

Para poder configurar o envio de qualquer um desses tipos de registros para o Firehose pela primeira vez, você precisa estar conectado a uma conta com as seguintes permissões.

- `logs:CreateLogDelivery`
- `firehose:TagDeliveryStream`
- `iam:CreateServiceLinkedRole`

Se algum desses tipos de registros já estiver sendo enviado para o Firehose, para configurar o envio de outro desses tipos de registros para o Firehose, você precisará ter apenas as permissões e. `logs:CreateLogDelivery` `firehose:TagDeliveryStream`

Funções do IAM usadas para permissões

Como o Firehose não usa políticas de recursos, AWS usa funções do IAM ao configurar esses registros para serem enviados ao Firehose. 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": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

Essa função vinculada ao serviço concede permissão para todos os streams de entrega do Firehose que têm a tag definida como. `LogDeliveryEnabled` `true` AWS fornece essa tag ao stream de entrega de destino quando você configura o registro.

Essa função vinculada ao serviço também tem uma política de confiança que permite que a entidade de serviço `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"
    }
  ]
}
```

Registro em log que requer permissões [v2] adicionais

Alguns AWS serviços usam um novo método para enviar seus registros. Esse é um método flexível que permite configurar a entrega de registros desses serviços para um ou mais dos seguintes destinos: CloudWatch Logs, Amazon S3 ou Firehose.

A entrega de um registro de trabalho consiste em três elementos:

- A`DeliverySource`, que é um objeto lógico que representa os recursos que realmente enviam os registros.

- `ADeliveryDestination`, que é um objeto lógico que representa o destino real da entrega.
- `ADelivery`, que conecta uma fonte de entrega ao destino de entrega

Para configurar a entrega de registros entre um AWS serviço compatível e um destino, você deve fazer o seguinte:

- Crie uma fonte de entrega com PutDeliverySource.
- Crie um destino de entrega com PutDeliveryDestination.
- Se você estiver entregando registros entre contas, deverá usá-los PutDeliveryDestinationPolicyna conta de destino para atribuir uma IAM política ao destino. Essa política autoriza a criação de uma entrega da fonte de entrega na conta A até o destino da entrega na conta B. Para entrega entre contas, você mesmo deve criar manualmente as políticas de permissão.
- Crie uma entrega combinando exatamente uma fonte de entrega e um destino de entrega, usando CreateDelivery.

As seções a seguir fornecem os detalhes das permissões que você precisa ter ao fazer login para configurar a entrega de logs para cada tipo de destino, usando o processo v2. Essas permissões podem ser concedidas a um perfil do IAM com o qual você está conectado.

> **Important**
>
> É sua responsabilidade remover os recursos de entrega de registros após excluir o recurso gerador de registros. Para fazer isso, siga estas etapas.
>
> 1. Exclua o `Delivery` usando a DeleteDeliveryoperação.
> 2. Exclua o `DeliverySource` usando a DeleteDeliverySourceoperação.
> 3. Se o `DeliveryDestination` associado ao `DeliverySource` que você acabou de excluir for usado somente para esse específico`DeliverySource`, você poderá removê-lo usando a DeleteDeliveryDestinationsoperação.

Sumário

- Registros enviados para CloudWatch Logs
- Logs enviados ao Amazon S3
  - Criptografia no lado do servidor de bucket do Amazon S3

- Registros enviados para o Firehose
- Permissões específicas do serviço
- Permissões específicas do console

Registros enviados para CloudWatch Logs

Permissões de usuário

Para ativar o envio de CloudWatch registros para o Logs, você precisa estar conectado com as seguintes permissões.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
```

```
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyCWL",
            "Effect": "Allow",
            "Action": [
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:*"
            ]
        }
    ]
}
```

Política de recursos do grupo de logs

O grupo de logs para o qual os logs estão sendo enviados deve ter uma política de recursos que contenha determinadas permissões. Se o grupo de registros atualmente não tiver uma política de recursos e o usuário que configura o registro tiver as `logs:DescribeLogGroups` permissões`logs:PutResourcePolicy`,`logs:DescribeResourcePolicies`, e para o grupo de registros, AWS criará automaticamente a política a seguir quando você começar a enviar os CloudWatch registros para o 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:*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
          "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:*"]
        }
      }
    }
  ]
}
```

Considerações sobre o limite do tamanho da política de recursos do grupo de logs

Esses serviços devem listar cada grupo de registros para o qual estão enviando registros na política de recursos, e as políticas de recursos de CloudWatch registros estão limitadas a 5120 caracteres. Um serviço que envia logs a 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 registros e, quando detecta que uma política se aproxima do limite de tamanho de 5120 caracteres, o CloudWatch Logs ativa `/aws/vendedlogs/*` automaticamente a política de recursos desse serviço. Depois, você pode começar a usar grupos de logs com nomes que começam com `/aws/vendedlogs/` como destinos para os logs desses serviços.

Logs enviados ao Amazon S3

Permissões de usuário

Para permitir o envio de logs ao Amazon S3, é necessário fazer login com as permissões a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
```

```
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyS3",
            "Effect": "Allow",
            "Action": [
                "s3:PutBucketPolicy",
                "s3:GetBucketPolicy"
            ],
            "Resource": "arn:aws:s3:::bucket-name"
        }
    ]
}
```

O bucket do S3 para o qual os logs estão sendo enviados deve ter uma política de recursos que contenha determinadas permissões. Se o bucket atualmente não tiver uma política de recursos e o usuário que configura o registro tiver as `S3:PutBucketPolicy` permissões

`S3:GetBucketPolicy` e para o bucket, criará AWS automaticamente a seguinte política para ele quando você começar a enviar os registros 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",
            "Condition": {
                "StringEquals": {
                "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-source*"]
                }
            }
        },
        {
            "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",
                    "aws:SourceAccount": ["0123456789"]
                },
                "ArnLike": {
                    "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-
source:*"]
                }
            }
        }
```

```
    ]
}
```

Na política anterior, para `aws:SourceAccount`, especifique a lista de IDS de conta para os quais os logs estão sendo entregues a esse bucket. Para `aws:SourceArn`, especifique a lista de ARNs do recurso que gera os logs, no formulário `arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Se o bucket tiver uma política de recursos, mas ela não contiver a instrução exibida na política anterior, e o usuário configurando o log tiver as permissões `S3:GetBucketPolicy` e `S3:PutBucketPolicy` para o bucket, essa instrução será anexada à política de recursos do bucket.

> **Note**
>
> Em alguns casos, você pode ver `AccessDenied` erros AWS CloudTrail se a `s3:ListBucket` permissão não tiver sido concedida`delivery.logs.amazonaws.com`. Para evitar esses erros em seus CloudTrail registros, você deve conceder a `s3:ListBucket` permissão `delivery.logs.amazonaws.com` e incluir `Condition` os parâmetros mostrados com o conjunto de `s3:GetBucketAcl` permissões na política de bucket anterior. Para simplificar isso, em vez de criar uma nova `Statement`, você pode atualizar `AWSLogDeliveryAclCheck` diretamente para `“Action”: [“s3:GetBucketAcl”, “s3:ListBucket”]`

Criptografia no lado do servidor de bucket do Amazon S3

Você pode proteger os dados em seu bucket do Amazon S3 habilitando a criptografia do lado do servidor com chaves gerenciadas pelo Amazon S3 (SSE-S3) ou a criptografia do lado do servidor com uma chave armazenada em (SSE-KMS). AWS KMS AWS Key Management Service Para obter mais informações, consulte Proteger dados usando a criptografia no lado do servidor.

Se você escolher SSE-S3, nenhuma configuração adicional será necessária. O Amazon S3 lida com a chave de criptografia.

> **Warning**
>
> Se você escolher o SSE-KMS, deverá usar uma chave gerenciada pelo cliente, pois o uso de uma chave AWS gerenciada não é suportado nesse cenário. Se você configurar

a criptografia usando uma chave AWS gerenciada, os registros serão entregues em um formato ilegível.

Ao usar uma AWS KMS chave gerenciada pelo cliente, você pode especificar o Amazon Resource Name (ARN) da chave gerenciada pelo cliente ao ativar a criptografia do bucket. Você precisa adicionar o seguinte à política de chaves da chave gerenciada pelo cliente (não à política de bucket para o bucket do S3), para que a conta de entrega de log possa gravar no bucket do S3.

Se você escolher o SSE-KMS, deverá usar uma chave gerenciada pelo cliente, pois o uso de uma chave AWS gerenciada não é suportado nesse cenário. Ao usar uma AWS KMS chave gerenciada pelo cliente, você pode especificar o Amazon Resource Name (ARN) da chave gerenciada pelo cliente ao ativar a criptografia do bucket. Você precisa adicionar o seguinte à política de chaves da chave gerenciada pelo cliente (não à política de bucket para o bucket do S3), para que a conta de entrega de log possa gravar no bucket do S3.

```
{
    "Sid": "Allow Logs Delivery to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [ "delivery.logs.amazonaws.com" ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:SourceAccount": ["0123456789"]
        },
        "ArnLike": {
            "aws:SourceArn": ["arn:aws:logs:us-east-1:0123456789:delivery-source:*"]
        }
        }
}
```

Para `aws:SourceAccount`, especifique a lista de IDS de conta para os quais os logs estão sendo entregues a esse bucket. Para `aws:SourceArn`, especifique a lista de ARNs do recurso que gera os logs, no formulário `arn:aws:logs:`*`source-region`*`:`*`source-account-id`*`:*`.

Registros enviados para o Firehose

Permissões de usuário

Para ativar o envio de registros para o Firehose, você deve estar conectado com as seguintes permissões.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:GetDelivery",
                "logs:GetDeliverySource",
                "logs:PutDeliveryDestination",
                "logs:GetDeliveryDestinationPolicy",
                "logs:DeleteDeliverySource",
                "logs:PutDeliveryDestinationPolicy",
                "logs:CreateDelivery",
                "logs:GetDeliveryDestination",
                "logs:PutDeliverySource",
                "logs:DeleteDeliveryDestination",
                "logs:DeleteDeliveryDestinationPolicy",
                "logs:DeleteDelivery"
            ],
            "Resource": [
                "arn:aws:logs:region:account-id:delivery:*",
                "arn:aws:logs:region:account-id:delivery-source:*",
                "arn:aws:logs:region:account-id:delivery-destination:*"
            ]
        },
        {
            "Sid": "ListAccessForLogDeliveryActions",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeDeliveryDestinations",
                "logs:DescribeDeliverySources",
```

```
                "logs:DescribeDeliveries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowUpdatesToResourcePolicyFH",
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream"
            ],
            "Resource": [
                "arn:aws:firehose:region:account-id:deliverystream/*"
            ]
        },
        {
            "Sid": "CreateServiceLinkedRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::account-id:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery"
        }
    ]
}
```

Perfis do IAM usados para permissões de recursos

Como o Firehose não usa políticas de recursos, AWS usa funções do IAM ao configurar esses registros para serem enviados ao Firehose. 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": {
                    "aws:ResourceTag/LogDeliveryEnabled": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

Essa função vinculada ao serviço concede permissão para todos os streams de entrega do Firehose que têm a tag definida como. `LogDeliveryEnabled` `true` AWS fornece essa tag ao stream de entrega de destino quando você configura o registro.

Essa função vinculada ao serviço também tem uma política de confiança que permite que a entidade de serviço `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"
    }
  ]
}
```

Permissões específicas do serviço

Além das permissões específicas do destino listadas nas seções anteriores, alguns serviços exigem autorização explícita para que os clientes possam enviar registros de seus recursos, como uma camada adicional de segurança. Ele autoriza a `AllowVendedLogDeliveryForResource` ação de recursos que vendem registros dentro desse serviço. Para esses serviços, use a política a seguir e substitua o *`tipo de serviço e recurso`* pelos valores apropriados. Para obter os valores específicos do serviço para esses campos, consulte a página de documentação desses serviços para registros vendidos.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ServiceLevelAccessForLogDelivery",
            "Effect": "Allow",
            "Action": [
                "service:AllowVendedLogDeliveryForResource"
            ],
            "Resource": "arn:aws:service:region:account-id:resource-type/*"
        }
    ]
}
```

Permissões específicas do console

Além das permissões listadas nas seções anteriores, se você estiver configurando a entrega de registros usando o console em vez das APIs, também precisará das seguintes permissões adicionais:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLogDeliveryActionsConsoleCWL",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:*"
            ]
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleS3",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ]
```

```
        },
        {
            "Sid": "AllowLogDeliveryActionsConsoleFH",
            "Effect": "Allow",
            "Action": [
                "firehose:ListDeliveryStreams",
                "firehose:DescribeDeliveryStream"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Prevenção contra o ataque do “substituto confuso” em todos os serviços

“Confused deputy” é um problema de segurança no qual uma entidade sem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar em um problema confuso de delegado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição `aws:SourceArn`, `aws:SourceAccount`, `aws:SourceOrgID`, e `aws:SourceOrgPaths` global nas políticas de recursos para limitar as permissões que o CloudWatch Logs concede a outro serviço ao recurso. Use `aws:SourceArn` se quiser associar apenas um recurso ao acesso entre serviços. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços. Use `aws:SourceOrgID` se quiser permitir que qualquer recurso de qualquer conta de uma organização seja associado ao uso entre serviços. Use `aws:SourceOrgPaths` se quiser associar qualquer recurso das contas em um caminho do AWS Organizations seja associado ao uso entre serviços. Para obter mais informações sobre como usar e entender os caminhos, consulte Compreender o caminho da AWS Organizations entidade.

A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou especificar vários recursos, use a chave de condição de contexto global `aws:SourceArn` com caracteres curinga (*) para as partes desconhecidas do ARN. Por exemplo, `.arn:aws:servicename:*:123456789012:*`

Se o valor do `aws:SourceArn` não contiver o ID da conta, como um ARN de bucket do Amazon S3, você deverá usar ambos, a `aws:SourceAccount` e o `aws:SourceArn` para limitar as permissões.

Para se proteger do problema de "confused deputy" em grande escala, use a chave de contexto de condição global `aws:SourceOrgID` ou `aws:SourceOrgPaths` com o ID ou o caminho da organização do recurso nas políticas baseadas em recursos. As políticas que incluem a chave `aws:SourceOrgID` ou `aws:SourceOrgPaths` incluem automaticamente as contas corretas e você não tem que atualizar manualmente as políticas quando adiciona, remove ou move contas na organização.

As políticas nas seções anteriores desta página mostram como você pode usar as chaves de contexto de condição global `aws:SourceArn` e `aws:SourceAccount` para evitar o problema “confused deputy”.

CloudWatch Registra atualizações em políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do CloudWatch Logs desde que esse serviço começou a monitorar essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página Histórico de documentos de CloudWatch registros.

Alteração	Descrição	Data
AWSServiceRoleForLogDelivery política de função vinculada ao serviço — atualização de uma política existente	CloudWatch Os registros alteraram as permissões na política do IAM associada à função AWSServiceRoleForLogDeliveryvinculada ao serviço. Foi feita a seguinte alteração:	15 de julho de 2021

Alteração	Descrição	Data
	• A chave de condição `firehose:ResourceTag/LogDeliveryEnabled": "true"` foi alterada para `aws:ResourceTag/LogDeliveryEnabled": "true"` .	
CloudWatch Os registros começaram a rastrear as alterações	CloudWatch A Logs começou a rastrear as alterações em suas políticas AWS gerenciadas.	10 de junho de 2021

Exportar dados de log para o Amazon S3

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. É possível exportar para um bucket na mesma conta ou em uma conta diferente.

Você pode fazer o seguinte:

- Exporte dados de log para buckets do S3 que são criptografados pelo SSE-KMS em () AWS Key Management Service AWS KMS
- Exportar dados de log para buckets do S3 com o S3 Object Lock ativado com um período de retenção

> **Note**
>
> A exportação para o Amazon S3 é suportada somente para grupos de log na classe de log Standard. Para obter mais informações sobre classes de log, consulteClasses de log.

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 S3 para arquivar ou excluir arquivos exportados automaticamente.

Você pode exportar para buckets do S3 criptografados com AES-256 ou com SSE-KMS. Não há suporte para a exportação de buckets do S3 criptografados com DSSE-KMS.

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, é possível especificar um prefixo que será usado como o prefixo das chaves do Amazon S3 para todos os objetos exportados.

> **Note**
>
> A classificação baseada em tempo em pedaços de dados de log dentro de um arquivo exportado não é garantida. Você pode classificar os dados exportados do campo de log usando os utilitários do Linux. Por exemplo, o comando de utilitário a seguir classifica os eventos em todos os arquivos `.gz` em uma única pasta.

```
find . -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
```

O comando de utilitário a seguir classifica os arquivos .gz de várias subpastas.

```
find ./*/ -type f -exec zcat {} + | sed -r 's/^[0-9]+/\x0&/' | sort -z
```

Além disso, você pode usar outro comando `stdout` para canalizar a saída classificada para outro arquivo e salvá-la.

Pode levar até 12 horas para os dados de log se tornarem disponíveis para exportação. O tempo limite para exportar de dados de log é de 24 horas. Se suas tarefas de exportação estiverem atingindo o tempo limite, reduza o intervalo de tempo quando criar a tarefa de exportação.

Para obter informações sobre análise quase em tempo real de dados de log, consulte Análise de dados de registro com o CloudWatch Logs Insights ou Processamento em tempo real de dados de log com assinaturas.

Conteúdo

- Conceitos
- Exportação de dados de log para o Amazon S3 usando o console.
- Exporte dados de log para o Amazon S3 usando o AWS CLI
- Descrever tarefas de exportação
- Cancelar uma tarefa de exportação

Conceitos

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 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 registro no grupo de registros que foram ingeridos nesse período ou 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 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 Amazon S3 para todos os objetos exportados. Isso ajuda a criar uma organização do tipo pasta em seu bucket.

Exportação de dados de log para o Amazon S3 usando o console.

Nos exemplos a seguir, você usa o CloudWatch console da Amazon para exportar todos os dados de um grupo de CloudWatch logs do Amazon Logs chamado `my-log-group` para um bucket do Amazon S3 chamado. `my-exported-logs`

Há suporte para a exportação de dados de log para buckets do S3 criptografados por SSE-KMS. Não há suporte para a exportação de buckets do S3 criptografados com DSSE-KMS.

Os detalhes de como configurar a exportação dependem se o bucket do Amazon S3 para o qual você deseja exportar está na mesma conta que os logs que estão sendo exportados ou em uma conta diferente.

Tópicos

- Exportação para a mesma conta
- Exportação entre contas

Exportação para a mesma conta

Se o bucket do Amazon S3 estiver na mesma conta dos logs que estão sendo exportados, use as instruções nesta seção.

Tópicos

- Etapa 1: Crie um bucket do Amazon S3
- Etapa 2: configurar permissões de acesso
- Etapa 3: Definir permissões em um bucket do S3.
- (Opcional) Etapa 4: Exportar para um bucket criptografado com o SSE-KMS
- Etapa 5: Criar uma tarefa de exportação

Etapa 1: Crie 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 S3 deve residir na mesma região dos dados de log a serem exportados. CloudWatch O Logs não oferece suporte à exportação de dados para buckets do S3 em uma região diferente.

Para criar um bucket do S3

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Se necessário, altere a região da . Na barra de navegação, escolha a região em que seus CloudWatch registros residem.
3. Escolha Criar bucket.
4. Para Bucket Name (Nome do bucket), digite um nome para o bucket.
5. Em Região, selecione a região em que seus dados de CloudWatch registros residem.
6. Escolha Criar.

Etapa 2: configurar permissões de acesso

Para criar a tarefa de exportação na etapa 5, você precisará ter feio login com o perfil do IAM `AmazonS3ReadOnlyAccess` e com as seguintes permissões:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criar um perfil para um provedor de identidades de terceiros (federação) no Guia do usuário do IAM.
- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
  - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Etapa 3: Definir permissões em um bucket do S3.

Por padrão, todos os buckets e objetos do S3 são privados. Somente o proprietário do recurso, o Conta da AWS que criou o bucket pode acessá-lo e quaisquer objetos que ele contenha. 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.

> **Important**
>
> Para tornar as exportações para buckets S3 mais seguras, agora exigimos que você especifique a lista de contas de origem que têm permissão para exportar dados de log para seu bucket S3.
> No exemplo a seguir, a lista de IDs de conta na `aws:SourceAccount` chave seriam as contas das quais um usuário pode exportar dados de log para seu bucket do S3. A chave `aws:SourceArn` seria o recurso para o qual a ação está sendo realizada. Você pode restringir isso a um grupo de logs específico ou usar um curinga, como mostrado neste exemplo.
> Recomendamos que você inclua também o ID da conta na qual o bucket do S3 foi criado para permitir a exportação dentro da mesma conta.

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 política do bucket), adicione a política a seguir. Altere `my-exported-logs` para o nome do bucket do S3. Certifique-se de especificar o endpoint correto da região como `us-west-1` para a Entidade principal.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
```

```
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                    ]
            }
          }
      }
    ]
}
```

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 seu bucket do 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 instruções de acesso de CloudWatch registros a essa política ou políticas. Recomendamos avaliar o conjunto resultante de permissões para ter certeza de que elas são apropriadas para os usuários que acessarão o bucket.

(Opcional) Etapa 4: Exportar para um bucket criptografado com o SSE-KMS

Essa etapa é necessária somente se você estiver exportando para um bucket do S3 que usa criptografia do lado do servidor com. AWS KMS keys Essa criptografia é conhecida como SSE-KMS.

Para exportar para um bucket criptografado com SSE-KMS

1. Abra o AWS KMS console em https://console.aws.amazon.com/kms.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. Na barra de navegação esquerdo, escolha Customer managed keys (Chaves gerenciadas pelo cliente).

   Escolha Create Key (Criar chave).
4. Para Key type (Tipo de chave), escolha Symmetric (Simétrica).
5. Em Key usage (Uso da chave), escolha Encrypt and decrypt (Criptografar e descriptografar) e, em seguida, escolha Next (Avançar).
6. Em Add labels (Adicionar rótulos), insira um alias para a chave e, opcionalmente, adicione uma descrição ou tags. Em seguida, escolha Próximo.
7. Em Key administrators (Administradores de chaves), selecione quem pode administrar essa chave e escolha Next (Avançar).
8. Em Define key usage permissions (Definir permissões de uso da chave), não faça alterações e escolha Next (Avançar).
9. Revise as configurações e escolha Finish (Concluir).
10. De volta à página Customer managed keys (Chaves gerenciadas pelo cliente), escolha o nome da chave que você acabou de criar.
11. Na guia Key Policy (Política de chaves), selecione Switch to policy view (Alternar para visualização de política).

12. Na seção Key policy (Política de chaves), escolha Edit (Editar).
13. Adicione a declaração a seguir à lista de declarações de política de chaves. Ao fazer isso, substitua *Região* pela Região dos seus registros e substitua o *account-ARN (ARN da conta)* pelo ARN da conta que possui a chave KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

14. Escolha Salvar alterações.
15. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.

16. Encontre o bucket que você criou no Etapa 1: criar um bucket do S3 e escolha o nome do bucket.
17. Escolha a guia Properties (Propriedades). Em seguida, em Default encryption (Criptografia padrão), escolha Edit (Editar).
18. Em Server-side encryption (Criptografia no lado do servidor), escolha Enable (Habilitar).
19. Em Encryption type (Tipo de criptografia), selecione AWS Key Management Service key (SSE-KMS) (Chave do SSE-KMS)).
20. Escolha Escolher entre suas AWS KMS chaves e encontre a chave que você criou.
21. Para Bucket key (Chave do bucket), escolha Enable (Habilitar).
22. Escolha Salvar alterações.

Etapa 5: 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 CloudWatch

1. Faça login com permissões suficientes, conforme documentado em Etapa 2: configurar permissões de acesso.
2. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação, escolha Grupos de logs.
4. Na tela Grupos de logs, escolha o nome do grupo de logs.
5. Escolha Actions (Ações), Export to Amazon S3 (Exportar para o Amazon S3).
6. Na tela Export data to Amazon S3 (Exportar dados para o Amazon S3), em Define data export (Definir exportação de dados), defina o período dos dados a serem exportados usando From (De) e To (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 Escolher bucket do S3, escolha a conta associada ao bucket do S3.
9. Em Nome do bucket do S3, escolha um bucket do S3.
10. Em Prefixo do bucket do S3, insira a string gerada aleatoriamente que você especificou na política do bucket.
11. Escolha Export (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), View all exports to Amazon S3 (Visualizar todas as exportações para o Amazon S3).

Exportação entre contas

Se o bucket do Amazon S3 estiver em uma conta diferente da conta dos logs que estão sendo exportados, use as instruções nesta seção.

Tópicos

- Etapa 1: Crie um bucket do Amazon S3
- Etapa 2: configurar permissões de acesso
- Etapa 3: Definir permissões em um bucket do S3.
- (Opcional) Etapa 4: Exportar para um bucket criptografado com o SSE-KMS
- Etapa 5: Criar uma tarefa de exportação

Etapa 1: Crie 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 S3 deve residir na mesma região dos dados de log a serem exportados. CloudWatch O Logs não oferece suporte à exportação de dados para buckets do S3 em uma região diferente.

Para criar um bucket do S3

1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Se necessário, altere a região da . Na barra de navegação, escolha a região em que seus CloudWatch registros residem.
3. Escolha Criar bucket.
4. Para Bucket Name (Nome do bucket), digite um nome para o bucket.
5. Em Região, selecione a região em que seus dados de CloudWatch registros residem.
6. Escolha Criar.

Etapa 2: configurar permissões de acesso

Primeiro, você deve criar uma nova política do IAM para permitir que CloudWatch os Logs tenham a `s3:PutObject` permissão para o bucket Amazon S3 de destino na conta de destino.

A política que você cria depende se o bucket de destino usa AWS KMS criptografia.

Para criar uma política do IAM para exportar registros para um bucket do Amazon S3

1. Abra o console IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Políticas.
3. Escolha Criar política.
4. Na seção Editor de políticas, escolha JSON.
5. Se o bucket de destino não usar AWS KMS criptografia, cole a política a seguir no editor.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

Se o bucket de destino usar AWS KMS criptografia, cole a política a seguir no editor.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-exported-logs/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
```

```
      ],
      "Resource": "ARN_OF_KMS_KEY"
    }
  ]
}
```

6. Escolha Próximo.
7. Insira um nome de política. Você usará esse nome para vincular a política ao seu perfil do IAM.
8. Escolha Criar política para salvar a nova política.

Para criar a tarefa de exportação na etapa 5, será necessário ter feito login com o perfil do IAM `AmazonS3ReadOnlyAccess`. Você deverá estar conectado com a política do IAM que acabou de criar e com as seguintes permissões:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criar um perfil para um provedor de identidades de terceiros (federação) no Guia do usuário do IAM.
- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
  - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Etapa 3: Definir permissões em um bucket do S3.

Por padrão, todos os buckets e objetos do S3 são privados. Somente o proprietário do recurso, o Conta da AWS que criou o bucket pode acessá-lo e quaisquer objetos que ele contenha. 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.

> **Important**
>
> Para tornar as exportações para buckets S3 mais seguras, agora exigimos que você especifique a lista de contas de origem que têm permissão para exportar dados de log para seu bucket S3.
> No exemplo a seguir, a lista de IDs de conta na `aws:SourceAccount` chave seriam as contas das quais um usuário pode exportar dados de log para seu bucket do S3. A chave `aws:SourceArn` seria o recurso para o qual a ação está sendo realizada. Você pode restringir isso a um grupo de logs específico ou usar um curinga, como mostrado neste exemplo.
> Recomendamos que você inclua também o ID da conta na qual o bucket do S3 foi criado para permitir a exportação dentro da mesma conta.

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 política do bucket), adicione a política a seguir. Altere `my-exported-logs` para o nome do bucket do S3. Certifique-se de especificar o endpoint correto da região como `us-west-1` para a Entidade principal.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
```

```
            "Resource": "arn:aws:s3:::my-exported-logs",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                       ]
              }
            }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                      ]
              }
            }
        },
        {
            "Effect": "Allow",
```

```
            "Principal": {
              "AWS": "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control"
              }
            }
        }
    ]
}
```

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 seu bucket do 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 instruções de acesso de CloudWatch registros a essa política ou políticas. Recomendamos avaliar o conjunto resultante de permissões para ter certeza de que elas são apropriadas para os usuários que acessarão o bucket.

(Opcional) Etapa 4: Exportar para um bucket criptografado com o SSE-KMS

Essa etapa é necessária somente se você estiver exportando para um bucket do S3 que usa criptografia do lado do servidor com. AWS KMS keys Essa criptografia é conhecida como SSE-KMS.

Para exportar para um bucket criptografado com SSE-KMS

1. Abra o AWS KMS console em https://console.aws.amazon.com/kms.
2. Para alterar o Região da AWS, use o seletor de região no canto superior direito da página.
3. Na barra de navegação esquerdo, escolha Customer managed keys (Chaves gerenciadas pelo cliente).

   Escolha Create Key (Criar chave).
4. Para Key type (Tipo de chave), escolha Symmetric (Simétrica).

5. Em Key usage (Uso da chave), escolha Encrypt and decrypt (Criptografar e descriptografar) e, em seguida, escolha Next (Avançar).
6. Em Add labels (Adicionar rótulos), insira um alias para a chave e, opcionalmente, adicione uma descrição ou tags. Em seguida, escolha Próximo.
7. Em Key administrators (Administradores de chaves), selecione quem pode administrar essa chave e escolha Next (Avançar).
8. Em Define key usage permissions (Definir permissões de uso da chave), não faça alterações e escolha Next (Avançar).
9. Revise as configurações e escolha Finish (Concluir).
10. De volta à página Customer managed keys (Chaves gerenciadas pelo cliente), escolha o nome da chave que você acabou de criar.
11. Na guia Key Policy (Política de chaves), selecione Switch to policy view (Alternar para visualização de política).
12. Na seção Key policy (Política de chaves), escolha Edit (Editar).
13. Adicione a declaração a seguir à lista de declarações de política de chaves. Ao fazer isso, substitua *Região* pela Região dos seus registros e substitua o *account-ARN (ARN da conta)* pelo ARN da conta que possui a chave KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
```

```
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS":
 "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

14. Escolha Salvar alterações.
15. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
16. Encontre o bucket que você criou no Etapa 1: criar um bucket do S3 e escolha o nome do bucket.
17. Escolha a guia Properties (Propriedades). Em seguida, em Default encryption (Criptografia padrão), escolha Edit (Editar).
18. Em Server-side encryption (Criptografia no lado do servidor), escolha Enable (Habilitar).
19. Em Encryption type (Tipo de criptografia), selecione AWS Key Management Service key (SSE-KMS) (Chave do SSE-KMS)).
20. Escolha Escolher entre suas AWS KMS chaves e encontre a chave que você criou.
21. Para Bucket key (Chave do bucket), escolha Enable (Habilitar).
22. Escolha Salvar alterações.

Etapa 5: 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 CloudWatch

1. Faça login com permissões suficientes, conforme documentado em Etapa 2: configurar permissões de acesso.
2. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
3. No painel de navegação, escolha Grupos de logs.
4. Na tela Grupos de logs, escolha o nome do grupo de logs.
5. Escolha Actions (Ações), Export to Amazon S3 (Exportar para o Amazon S3).
6. Na tela Export data to Amazon S3 (Exportar dados para o Amazon S3), em Define data export (Definir exportação de dados), defina o período dos dados a serem exportados usando From (De) e To (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 Escolher bucket do S3, escolha a conta associada ao bucket do S3.
9. Em Nome do bucket do S3, escolha um bucket do S3.
10. Em Prefixo do bucket do S3, insira a string gerada aleatoriamente que você especificou na política do bucket.
11. Escolha Export (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), View all exports to Amazon S3 (Visualizar todas as exportações para o Amazon S3).

Exporte dados de log para o Amazon S3 usando o AWS CLI

No exemplo a seguir, você usa uma tarefa de exportação para exportar todos os dados de um grupo de CloudWatch logs de registros chamado `my-log-group` para um bucket do Amazon S3 chamado. `my-exported-logs` Este exemplo pressupõe que você já tenha criado um grupo de logs denominado `my-log-group`.

Há suporte para a exportação de dados de log para buckets do S3 que são criptografados pelo AWS KMS . Não há suporte para a exportação de buckets do S3 criptografados com DSSE-KMS.

Os detalhes de como configurar a exportação dependem se o bucket do Amazon S3 para o qual você deseja exportar está na mesma conta que os logs que estão sendo exportados ou em uma conta diferente.

Tópicos

- Exportação para a mesma conta
- Exportação entre contas

Exportação para a mesma conta

Se o bucket do Amazon S3 estiver na mesma conta dos logs que estão sendo exportados, use as instruções nesta seção.

Tópicos

- Etapa 1: criar um bucket do S3
- Etapa 2: configurar permissões de acesso
- Etapa 3: Definir permissões em um bucket do S3.
- (Opcional) Etapa 4: Exportar para um bucket criptografado com o SSE-KMS
- Etapa 5: Criar uma tarefa de exportação

Etapa 1: criar um bucket do 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 S3 deve residir na mesma região dos dados de log a serem exportados. CloudWatch O Logs não oferece suporte à exportação de dados para buckets do S3 em uma região diferente.

Para criar um bucket S3 usando o 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
```

A seguir, um exemplo de saída.

```
{
    "Location": "/my-exported-logs"
}
```

Etapa 2: configurar permissões de acesso

Para criar a tarefa de exportação na etapa 5, você precisará ter feio login com o perfil do IAM `AmazonS3ReadOnlyAccess` e com as seguintes permissões:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criar um perfil para um provedor de identidades de terceiros (federação) no Guia do usuário do IAM.
- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
  - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Etapa 3: Definir permissões em um bucket do S3.

Por padrão, todos os buckets e objetos do 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.

> **Important**
>
> Para tornar as exportações para buckets S3 mais seguras, agora exigimos que você especifique a lista de contas de origem que têm permissão para exportar dados de log para seu bucket S3.
> No exemplo a seguir, a lista de IDs de conta na `aws:SourceAccount` chave seriam as contas das quais um usuário pode exportar dados de log para seu bucket do S3. A chave `aws:SourceArn` seria o recurso para o qual a ação está sendo realizada. Você pode restringir isso a um grupo de logs específico ou usar um curinga, como mostrado neste exemplo.
> Recomendamos que você inclua também o ID da conta na qual o bucket do S3 foi criado para permitir a exportação dentro da mesma conta.

Para definir permissões em um bucket do S3.

1. Crie um arquivo denominado `policy.json` e adicione a seguinte política de acesso, alterando `my-exported-logs` para o nome do bucket do S3 e `Principal` para o endpoint da região na qual você está exportando dados de log, como, por exemplo, `us-west-1`. Use um editor de texto para criar este arquivo de política. Não use o console do IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "aws:SourceAccount": [
                    "AccountId1",
```

```
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                     ]
            }
          }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/*",
          "Principal": { "Service": "logs.Region.amazonaws.com" },
          "Condition": {
            "StringEquals": {
                "s3:x-amz-acl": "bucket-owner-full-control",
                "aws:SourceAccount": [
                    "AccountId1",
                    "AccountId2",
                    ...
                ]
            },
            "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:Region:AccountId1:log-group:*",
                        "arn:aws:logs:Region:AccountId2:log-group:*",
                        ...
                    ]
            }
          }
      }
    ]
}
```

2. Defina a política que você acabou de adicionar como política de acesso no seu bucket usando o put-bucket-policycomando. Essa política permite que o CloudWatch Logs exporte dados de log para seu bucket do 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 instruções de acesso de CloudWatch registros a essa política ou políticas. Recomendamos avaliar o conjunto resultante de permissões para ter certeza de que elas são apropriadas para os usuários que acessarão o bucket.

(Opcional) Etapa 4: Exportar para um bucket criptografado com o SSE-KMS

Essa etapa é necessária somente se você estiver exportando para um bucket do S3 que usa criptografia do lado do servidor com. AWS KMS keys Essa criptografia é conhecida como SSE-KMS.

Para exportar para um bucket criptografado com SSE-KMS

1. Use um editor de texto para criar um arquivo chamado `key_policy.json` e adicione a seguinte política de acesso. Ao adicionar a política, faça as seguintes alterações:

   - Substitua *`Region`* (Região) pela região dos seus logs.
   - Substitua *`account-ARN (conta da ARN)`* pelo ARN da conta proprietária da chave do KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

2. Digite o comando :

```
aws kms create-key --policy file://key_policy.json
```

A seguir está um exemplo de saída deste comando:

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account_id:key/key_id",
        "CreationDate": "time",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
```

```
        "MultiRegion": false
    }
```

3. Use um editor de texto para criar um arquivo chamado `bucketencryption.json` com o seguinte conteúdo:

```
{
  "Rules": [
    {
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. Digite o seguinte comando, substituindo *bucket-name* pelo nome do bucket para o qual você está exportando logs.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

   Se o comando não retornar um erro, o processo foi bem-sucedido.

Etapa 5: Criar uma tarefa de exportação

Use o comando a seguir para criar a tarefa de exportação. Depois de criada, a tarefa de exportação pode levar de alguns segundos a algumas horas, dependendo do tamanho dos dados a serem exportados.

Para exportar dados para o Amazon S3 usando o AWS CLI

1. Faça login com permissões suficientes, conforme documentado em Etapa 2: configurar permissões de acesso.
2. Em um prompt de comando, use o create-export-taskcomando a seguir 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"
```

A seguir, um exemplo de saída.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Exportação entre contas

Se o bucket do Amazon S3 estiver em uma conta diferente da conta dos logs que estão sendo exportados, use as instruções nesta seção.

Tópicos

- Etapa 1: criar um bucket do S3
- Etapa 2: configurar permissões de acesso
- Etapa 3: Definir permissões em um bucket do S3.
- (Opcional) Etapa 4: Exportar para um bucket criptografado com o SSE-KMS
- Etapa 5: Criar uma tarefa de exportação

Etapa 1: criar um bucket do 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 S3 deve residir na mesma região dos dados de log a serem exportados. CloudWatch O Logs não oferece suporte à exportação de dados para buckets do S3 em uma região diferente.

Para criar um bucket S3 usando o 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
```

A seguir, um exemplo de saída.

```
{
    "Location": "/my-exported-logs"
}
```

Etapa 2: configurar permissões de acesso

Primeiro, você deve criar uma nova política do IAM para permitir que CloudWatch os Logs tenham a `s3:PutObject` permissão para o bucket Amazon S3 de destino.

Para criar a tarefa de exportação na etapa 5, será necessário ter feito login com o perfil do IAM `AmazonS3ReadOnlyAccess` e com algumas outras permissões. Você pode criar uma política que contenha algumas dessas outras permissões necessárias.

A política que você cria depende se o bucket de destino usa AWS KMS criptografia. Se não usar AWS KMS criptografia, crie uma política com o conteúdo a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        }
    ]
}
```

Se o bucket de destino usar AWS KMS criptografia, crie uma política com o conteúdo a seguir.

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

```
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

Para criar a tarefa de exportação na etapa 5, é necessário estar conectado com o perfil do IAM `AmazonS3ReadOnlyAccess`, a política do IAM que você acabou de criar e também com as seguintes permissões:

- `logs:CreateExportTask`
- `logs:CancelExportTask`
- `logs:DescribeExportTasks`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criar um perfil para um provedor de identidades de terceiros (federação) no Guia do usuário do IAM.
- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.

- (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Etapa 3: Definir permissões em um bucket do S3.

Por padrão, todos os buckets e objetos do 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.

> **Important**
>
> Para tornar as exportações para buckets S3 mais seguras, agora exigimos que você especifique a lista de contas de origem que têm permissão para exportar dados de log para seu bucket S3.
> No exemplo a seguir, a lista de IDs de conta na `aws:SourceAccount` chave seriam as contas das quais um usuário pode exportar dados de log para seu bucket do S3. A chave `aws:SourceArn` seria o recurso para o qual a ação está sendo realizada. Você pode restringir isso a um grupo de logs específico ou usar um curinga, como mostrado neste exemplo.
> Recomendamos que você inclua também o ID da conta na qual o bucket do S3 foi criado para permitir a exportação dentro da mesma conta.

Para definir permissões em um bucket do S3.

1. Crie um arquivo denominado `policy.json` e adicione a seguinte política de acesso, alterando `my-exported-logs` para o nome do bucket do S3 e `Principal` para o endpoint da região na qual você está exportando dados de log, como, por exemplo, `us-west-1`. Use um editor de texto para criar este arquivo de política. Não use o console do IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
```

```
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                       ]
              }
            }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Principal": { "Service": "logs.Region.amazonaws.com" },
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control",
                  "aws:SourceAccount": [
                      "AccountId1",
                      "AccountId2",
                      ...
                  ]
              },
              "ArnLike": {
                      "aws:SourceArn": [
                          "arn:aws:logs:Region:AccountId1:log-group:*",
                          "arn:aws:logs:Region:AccountId2:log-group:*",
                          ...
                      ]
              }
            }
        },
        {
```

```
            "Effect": "Allow",
            "Principal": {
              "AWS": "arn:aws:iam::create_export_task_caller_account:role/role_name"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Condition": {
              "StringEquals": {
                  "s3:x-amz-acl": "bucket-owner-full-control"
              }
            }
        }
    ]
}
```

2. Defina a política que você acabou de adicionar como política de acesso no seu bucket usando o put-bucket-policycomando. Essa política permite que o CloudWatch Logs exporte dados de log para seu bucket do 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 instruções de acesso de CloudWatch registros a essa política ou políticas. Recomendamos avaliar o conjunto resultante de permissões para ter certeza de que elas são apropriadas para os usuários que acessarão o bucket.

(Opcional) Etapa 4: Exportar para um bucket criptografado com o SSE-KMS

Essa etapa é necessária somente se você estiver exportando para um bucket do S3 que usa criptografia do lado do servidor com. AWS KMS keys Essa criptografia é conhecida como SSE-KMS.

Para exportar para um bucket criptografado com SSE-KMS

1. Use um editor de texto para criar um arquivo chamado `key_policy.json` e adicione a seguinte política de acesso. Ao adicionar a política, faça as seguintes alterações:

- Substitua *Region* (Região) pela região dos seus logs.
- Substitua *account-ARN (conta da ARN)* pelo ARN da conta proprietária da chave do KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow CWL Service Principal usage",
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.Region.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "account-ARN"
            },
            "Action": [
                "kms:GetKeyPolicy*",
                "kms:PutKeyPolicy*",
                "kms:DescribeKey*",
                "kms:CreateAlias*",
                "kms:ScheduleKeyDeletion*",
                "kms:Decrypt"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM Role Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS":
 "arn:aws:iam::create_export_task_caller_account:role/role_name"
```

```
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "ARN_OF_KMS_KEY"
        }
    ]
}
```

2. Digite o comando :

```
aws kms create-key --policy file://key_policy.json
```

A seguir está um exemplo de saída deste comando:

```
{
    "KeyMetadata": {
        "AWSAccountId": "account_id",
        "KeyId": "key_id",
        "Arn": "arn:aws:kms:us-east-2:account_id:key/key_id",
        "CreationDate": "time",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": false
    }
```

3. Use um editor de texto para criar um arquivo chamado `bucketencryption.json` com o seguinte conteúdo:

```
{
  "Rules": [
    {
```

```
      "ApplyServerSideEncryptionByDefault": {
        "SSEAlgorithm": "aws:kms",
        "KMSMasterKeyID": "{KMS Key ARN}"
      },
      "BucketKeyEnabled": true
    }
  ]
}
```

4. Digite o seguinte comando, substituindo *bucket-name* pelo nome do bucket para o qual você está exportando logs.

```
aws s3api put-bucket-encryption --bucket bucket-name --server-side-encryption-
configuration file://bucketencryption.json
```

Se o comando não retornar um erro, o processo foi bem-sucedido.

Etapa 5: Criar uma tarefa de exportação

Use o comando a seguir para criar a tarefa de exportação. Depois de criada, a tarefa de exportação pode levar de alguns segundos a algumas horas, dependendo do tamanho dos dados a serem exportados.

Para exportar dados para o Amazon S3 usando o AWS CLI

1. Faça login com permissões suficientes, conforme documentado em Etapa 2: configurar permissões de acesso.
2. Em um prompt de comando, use o create-export-taskcomando a seguir 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"
```

A seguir, um exemplo de saída.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
```

```
}
```

Descrever 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 o AWS CLI

Em um prompt de comando, use o describe-export-taskscomando a seguir.

```
aws logs --profile CWLExportUser describe-export-tasks --task-id
 "cda45419-90ea-4db5-9833-aade86253e66"
```

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 - lista todas as suas tarefas de exportação, na ordem inversa de criação.
- Filtrar com base no ID de tarefa - lista a tarefa de exportação, se houver, com o ID especificado.
- Filtrar com base no status da tarefa - lista as tarefas de exportação com o status 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"
```

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
   }]
}
```

Cancelar uma tarefa de exportação

Você pode cancelar uma tarefa de exportação se estiver no estado `PENDING` ou `RUNNING`.

Para cancelar uma tarefa de exportação usando o AWS CLI

Em um prompt de comando, use o seguinte cancel-export-taskcomando:

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

Você pode usar o describe-export-taskscomando para verificar se a tarefa foi cancelada com êxito.

Streaming de dados de CloudWatch registros para o Amazon OpenSearch Service

Você pode configurar um grupo de CloudWatch registros do Logs para transmitir os dados que ele recebe para o seu cluster do Amazon OpenSearch Service quase em tempo real por meio de uma assinatura do CloudWatch Logs. Para ter mais informações, consulte Processamento em tempo real de dados de log com assinaturas.

> **Note**
>
> O streaming para o OpenSearch serviço é suportado somente para grupos de registros na classe de log Standard. Para obter mais informações sobre classes de log, consulteClasses de log.

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 Escalabilidade de funções do Lambda.

> **Note**
>
> Transmitir grandes quantidades de dados de CloudWatch registros para o OpenSearch Serviço pode resultar em altas taxas de uso. Recomendamos que você crie um orçamento no AWS Billing and Cost Management console. Para obter mais informações, consulte Gerenciar seus custos com o AWS Budgets.

Pré-requisitos

Antes de começar, crie um domínio OpenSearch de serviço. O domínio pode ter acesso público ou acesso à VPC, mas você não poderá modificar o tipo de acesso depois que o domínio for criado. Talvez você queira revisar as configurações do domínio de OpenSearch serviço posteriormente e modificar a configuração do cluster com base na quantidade de dados que o cluster processará. Para obter instruções sobre como criar um domínio, consulte Criação OpenSearch de domínios de serviço.

Para obter mais informações sobre o OpenSearch serviço, consulte o Amazon OpenSearch Service Developer Guide.

Inscrever um grupo de registros no OpenSearch Serviço

Você pode usar o CloudWatch console para inscrever um grupo de registros no OpenSearch Serviço.

Para inscrever um grupo de registros no OpenSearch Serviço

1. Abra o CloudWatch console em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Grupos de logs.
3. Selecione o nome do grupo de logs.
4. Escolha Ações, Filtros de assinatura, Criar filtro OpenSearch de assinatura do Amazon Service.
5. Escolha se deseja fazer streaming para um cluster nessa conta ou em outra conta.
   - Se você escolheu essa conta, selecione o domínio que criou na etapa anterior.
   - Caso tenha escolhido outra conta, forneça o ARN do domínio e o endpoint.
6. Para a função de execução do Lambda IAM, escolha a função do IAM que o Lambda deve usar ao executar chamadas. OpenSearch

   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 OpenSearch de serviço de destino usar acesso à VPC, a função deverá ter a AWSLambdaVPCAccessExecutionRolepolítica anexada. Essa política gerenciada pela Amazon concede ao Lambda acesso à VPC do cliente, permitindo que a Lambda grave no endpoint na VPC. OpenSearch

7. Em Log format, escolha um formato de log.
8. Em Subscription filter pattern (Padrão de filtro de assinatura), digite os termos ou o padrão a ser localizado nos eventos de log. Isso garante que você envie somente os dados de seu interesse para o seu OpenSearch cluster. Para ter mais informações, consulte Criar métricas de eventos de log usando filtros.
9. (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.
10. Selecione Start streaming (Iniciar transmissão).

Exemplos de código para CloudWatch registros usando AWS SDKs

Os exemplos de código a seguir mostram como usar o CloudWatch Logs com um kit AWS de desenvolvimento de software (SDK).

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar funções de serviço específicas, é possível ver as ações contextualizadas em seus devidos cenários e exemplos entre serviços.

Cenários são exemplos de código que mostram como realizar uma tarefa específica chamando várias funções dentro do mesmo serviço.

Exemplos entre serviços são amostras de aplicações que funcionam em vários Serviços da AWS.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- Ações para CloudWatch registros usando AWS SDKs
  - Use AssociateKmsKey com um AWS SDK ou CLI
  - Use CancelExportTask com um AWS SDK ou CLI
  - Use CreateExportTask com um AWS SDK ou CLI
  - Use CreateLogGroup com um AWS SDK ou CLI
  - Use CreateLogStream com um AWS SDK ou CLI
  - Use DeleteLogGroup com um AWS SDK ou CLI
  - Use DeleteSubscriptionFilter com um AWS SDK ou CLI
  - Use DescribeExportTasks com um AWS SDK ou CLI
  - Use DescribeLogGroups com um AWS SDK ou CLI
  - Use DescribeSubscriptionFilters com um AWS SDK ou CLI
  - Use GetQueryResults com um AWS SDK ou CLI
  - Use PutSubscriptionFilter com um AWS SDK ou CLI
  - Use StartLiveTail com um AWS SDK ou CLI

- Use StartQuery com um AWS SDK ou CLI
- Cenários para CloudWatch registros usando AWS SDKs
  - Use CloudWatch registros para executar uma consulta grande
- Exemplos de vários serviços para CloudWatch registros usando AWS SDKs
  - Usar eventos programados para chamar uma função do Lambda

Ações para CloudWatch registros usando AWS SDKs

Os exemplos de código a seguir demonstram como realizar ações individuais do CloudWatch Logs com AWS SDKs. Esses trechos chamam a API CloudWatch Logs e são trechos de código de programas maiores que devem ser executados em contexto. Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Para obter uma lista completa, consulte a Referência da API Amazon CloudWatch Logs.

Exemplos

- Use AssociateKmsKey com um AWS SDK ou CLI
- Use CancelExportTask com um AWS SDK ou CLI
- Use CreateExportTask com um AWS SDK ou CLI
- Use CreateLogGroup com um AWS SDK ou CLI
- Use CreateLogStream com um AWS SDK ou CLI
- Use DeleteLogGroup com um AWS SDK ou CLI
- Use DeleteSubscriptionFilter com um AWS SDK ou CLI
- Use DescribeExportTasks com um AWS SDK ou CLI
- Use DescribeLogGroups com um AWS SDK ou CLI
- Use DescribeSubscriptionFilters com um AWS SDK ou CLI
- Use GetQueryResults com um AWS SDK ou CLI
- Use PutSubscriptionFilter com um AWS SDK ou CLI
- Use StartLiveTail com um AWS SDK ou CLI
- Use StartQuery com um AWS SDK ou CLI

Use **`AssociateKmsKey`** com um AWS SDK ou CLI

O código de exemplo a seguir mostra como usar `AssociateKmsKey`.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to associate an AWS Key Management Service (AWS KMS) key with
    /// an Amazon CloudWatch Logs log group.
    /// </summary>
    public class AssociateKmsKey
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();

            string kmsKeyId = "arn:aws:kms:us-west-2:<account-
number>:key/7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";
            string groupName = "cloudwatchlogs-example-loggroup";

            var request = new AssociateKmsKeyRequest
            {
                KmsKeyId = kmsKeyId,
                LogGroupName = groupName,
            };
```

```
            var response = await client.AssociateKmsKeyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully associated KMS key ID:
{kmsKeyId} with log group: {groupName}.");
            }
            else
            {
                Console.WriteLine("Could not make the association between:
{kmsKeyId} and {groupName}.");
            }
        }
    }
```

- Para obter detalhes da API, consulte AssociateKmsKeya Referência AWS SDK for .NET da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`CancelExportTask`** com um AWS SDK ou CLI

O código de exemplo a seguir mostra como usar `CancelExportTask`.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
    using System;
```

```
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to cancel an Amazon CloudWatch Logs export task.
/// </summary>
public class CancelExportTask
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();
        string taskId = "exampleTaskId";

        var request = new CancelExportTaskRequest
        {
            TaskId = taskId,
        };

        var response = await client.CancelExportTaskAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"{taskId} successfully canceled.");
        }
        else
        {
            Console.WriteLine($"{taskId} could not be canceled.");
        }
    }
}
```

- Para obter detalhes da API, consulte CancelExportTaska Referência AWS SDK for .NET da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`CreateExportTask`** com um AWS SDK ou CLI

O código de exemplo a seguir mostra como usar `CreateExportTask`.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to create an Export Task to export the contents of the Amazon
    /// CloudWatch Logs to the specified Amazon Simple Storage Service (Amazon
S3)
    /// bucket.
    /// </summary>
    public class CreateExportTask
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();
            string taskName = "export-task-example";
            string logGroupName = "cloudwatchlogs-example-loggroup";
            string destination = "doc-example-bucket";
            var fromTime = 1437584472382;
```

```
            var toTime = 1437584472833;

            var request = new CreateExportTaskRequest
            {
                From = fromTime,
                To = toTime,
                TaskName = taskName,
                LogGroupName = logGroupName,
                Destination = destination,
            };

            var response = await client.CreateExportTaskAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"The task, {taskName} with ID: " +
                                  $"{response.TaskId} has been created
successfully.");
            }
        }
    }
```

- Para obter detalhes da API, consulte CreateExportTaska Referência AWS SDK for .NET da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`CreateLogGroup`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `CreateLogGroup`.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to create an Amazon CloudWatch Logs log group.
/// </summary>
public class CreateLogGroup
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();

        string logGroupName = "cloudwatchlogs-example-loggroup";

        var request = new CreateLogGroupRequest
        {
            LogGroupName = logGroupName,
        };

        var response = await client.CreateLogGroupAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"Successfully create log group with ID:
{logGroupName}.");
        }
```

```
            else
            {
                Console.WriteLine("Could not create log group.");
            }
        }
    }
```

- Para obter detalhes da API, consulte CreateLogGroupa Referência AWS SDK for .NET da API.

CLI

AWS CLI

O seguinte comando cria um grupo de logs chamado `my-logs`:

```
aws logs create-log-group --log-group-name my-logs
```

- Para obter detalhes da API, consulte CreateLogGroupna Referência de AWS CLI Comandos.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import { CreateLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new CreateLogGroupCommand({
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
```

```
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Para obter detalhes da API, consulte CreateLogGroupa Referência AWS SDK for JavaScript da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`CreateLogStream`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `CreateLogStream`.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Shows how to create an Amazon CloudWatch Logs stream for a CloudWatch
    /// log group.
```

```
    /// </summary>
    public class CreateLogStream
    {
        public static async Task Main()
        {
            // This client object will be associated with the same AWS Region
            // as the default user on this system. If you need to use a
            // different AWS Region, pass it as a parameter to the client
            // constructor.
            var client = new AmazonCloudWatchLogsClient();
            string logGroupName = "cloudwatchlogs-example-loggroup";
            string logStreamName = "cloudwatchlogs-example-logstream";

            var request = new CreateLogStreamRequest
            {
                LogGroupName = logGroupName,
                LogStreamName = logStreamName,
            };

            var response = await client.CreateLogStreamAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"{logStreamName} successfully created for
{logGroupName}.");
            }
            else
            {
                Console.WriteLine("Could not create stream.");
            }
        }
    }
```

- Para obter detalhes da API, consulte CreateLogStreama Referência AWS SDK for .NET da API.

CLI

AWS CLI

O seguinte comando cria um fluxo de logs `20150601` no grupo de logs `my-logs`:

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- Para obter detalhes da API, consulte CreateLogStreamna Referência de AWS CLI Comandos.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DeleteLogGroup`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `DeleteLogGroup`.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Uses the Amazon CloudWatch Logs Service to delete an existing
/// CloudWatch Logs log group.
/// </summary>
public class DeleteLogGroup
{
    public static async Task Main()
    {
        var client = new AmazonCloudWatchLogsClient();
        string logGroupName = "cloudwatchlogs-example-loggroup";

        var request = new DeleteLogGroupRequest
```

```
            {
                LogGroupName = logGroupName,
            };

            var response = await client.DeleteLogGroupAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully deleted CloudWatch log group,
{logGroupName}.");
            }
        }
    }
```

- Para obter detalhes da API, consulte DeleteLogGroupa Referência AWS SDK for .NET da API.

CLI

AWS CLI

O seguinte comando exclui um grupo de logs chamado `my-logs`:

```
aws logs delete-log-group --log-group-name my-logs
```

- Para obter detalhes da API, consulte DeleteLogGroupna Referência de AWS CLI Comandos.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import { DeleteLogGroupCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteLogGroupCommand({
    // The name of the log group.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Para obter detalhes da API, consulte DeleteLogGroupa Referência AWS SDK for JavaScript da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DeleteSubscriptionFilter`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `DeleteSubscriptionFilter`.

C++

SDK para C++

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Inclua os arquivos necessários.

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DeleteSubscriptionFilterRequest.h>
#include <iostream>
```

Excluir o filtro de assinatura.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DeleteSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetLogGroupName(log_group);

        auto outcome = cwl.DeleteSubscriptionFilter(request);
        if (!outcome.IsSuccess()) {
            std::cout << "Failed to delete CloudWatch log subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        } else {
            std::cout << "Successfully deleted CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- Para obter detalhes da API, consulte DeleteSubscriptionFiltera Referência AWS SDK for C++ da API.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DeleteSubscriptionFilterRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <logGroup>

                Where:
                  filter - The name of the subscription filter (for example,
 MyFilter).
                  logGroup - The name of the log group. (for example, testgroup).
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String filter = args[0];
        String logGroup = args[1];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .build();

        deleteSubFilter(logs, filter, logGroup);
        logs.close();
    }

    public static void deleteSubFilter(CloudWatchLogsClient logs, String filter,
 String logGroup) {
```

```
        try {
            DeleteSubscriptionFilterRequest request =
 DeleteSubscriptionFilterRequest.builder()
                    .filterName(filter)
                    .logGroupName(logGroup)
                    .build();

            logs.deleteSubscriptionFilter(request);
            System.out.printf("Successfully deleted CloudWatch logs subscription
 filter %s", filter);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte DeleteSubscriptionFiltera Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import { DeleteSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteSubscriptionFilterCommand({
    // The name of the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,
    // The name of the log group.
```

```
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Para obter detalhes da API, consulte DeleteSubscriptionFiltera Referência AWS SDK for JavaScript da API.

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  filterName: "FILTER",
  logGroupName: "LOG_GROUP",
};

cwl.deleteSubscriptionFilter(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
```

```
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte DeleteSubscriptionFiltera Referência AWS SDK for JavaScript da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun deleteSubFilter(
    filter: String?,
    logGroup: String?,
) {
    val request =
        DeleteSubscriptionFilterRequest {
            filterName = filter
            logGroupName = logGroup
        }

    CloudWatchLogsClient { region = "us-west-2" }.use { logs ->
        logs.deleteSubscriptionFilter(request)
        println("Successfully deleted CloudWatch logs subscription filter named
 $filter")
    }
}
```

- Para obter detalhes da API, consulte a DeleteSubscriptionFilterreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DescribeExportTasks`** com um AWS SDK ou CLI

O código de exemplo a seguir mostra como usar `DescribeExportTasks`.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
using System;
using System.Threading.Tasks;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;

/// <summary>
/// Shows how to retrieve a list of information about Amazon CloudWatch
/// Logs export tasks.
/// </summary>
public class DescribeExportTasks
{
    public static async Task Main()
    {
        // This client object will be associated with the same AWS Region
        // as the default user on this system. If you need to use a
        // different AWS Region, pass it as a parameter to the client
        // constructor.
        var client = new AmazonCloudWatchLogsClient();

        var request = new DescribeExportTasksRequest
        {
            Limit = 5,
        };
```

```
            var response = new DescribeExportTasksResponse();

            do
            {
                response = await client.DescribeExportTasksAsync(request);
                response.ExportTasks.ForEach(t =>
                {
                    Console.WriteLine($"{t.TaskName} with ID: {t.TaskId} has
status: {t.Status}");
                });
            }
            while (response.NextToken is not null);
        }
    }
```

- Para obter detalhes da API, consulte DescribeExportTasksa Referência AWS SDK for .NET da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use `DescribeLogGroups` com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `DescribeLogGroups`.

.NET

AWS SDK for .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.CloudWatchLogs;
```

```
    using Amazon.CloudWatchLogs.Model;

    /// <summary>
    /// Retrieves information about existing Amazon CloudWatch Logs log groups
    /// and displays the information on the console.
    /// </summary>
    public class DescribeLogGroups
    {
        public static async Task Main()
        {
            // Creates a CloudWatch Logs client using the default
            // user. If you need to work with resources in another
            // AWS Region than the one defined for the default user,
            // pass the AWS Region as a parameter to the client constructor.
            var client = new AmazonCloudWatchLogsClient();

            bool done = false;
            string newToken = null;

            var request = new DescribeLogGroupsRequest
            {
                Limit = 5,
            };

            DescribeLogGroupsResponse response;

            do
            {
                if (newToken is not null)
                {
                    request.NextToken = newToken;
                }

                response = await client.DescribeLogGroupsAsync(request);

                response.LogGroups.ForEach(lg =>
                {
                    Console.WriteLine($"{lg.LogGroupName} is associated with the
key: {lg.KmsKeyId}.");
                    Console.WriteLine($"Created on:
{lg.CreationTime.Date.Date}");
                    Console.WriteLine($"Date for this group will be stored for:
{lg.RetentionInDays} days.\n");
                });
```

```
                if (response.NextToken is null)
                {
                    done = true;
                }
                else
                {
                    newToken = response.NextToken;
                }
            }
            while (!done);
        }
    }
```

- Para obter detalhes da API, consulte DescribeLogGroupsa Referência AWS SDK for .NET da API.

CLI

AWS CLI

O seguinte comando descreve um grupo de logs chamado `my-logs`:

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

Saída:

```
{
    "logGroups": [
        {
            "storedBytes": 0,
            "metricFilterCount": 0,
            "creationTime": 1433189500783,
            "logGroupName": "my-logs",
            "retentionInDays": 5,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
        }
    ]
}
```

- Para obter detalhes da API, consulte DescribeLogGroupsna Referência de AWS CLI Comandos.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import {
  paginateDescribeLogGroups,
  CloudWatchLogsClient,
} from "@aws-sdk/client-cloudwatch-logs";

const client = new CloudWatchLogsClient({});

export const main = async () => {
  const paginatedLogGroups = paginateDescribeLogGroups({ client }, {});
  const logGroups = [];

  for await (const page of paginatedLogGroups) {
    if (page.logGroups && page.logGroups.every((lg) => !!lg)) {
      logGroups.push(...page.logGroups);
    }
  }

  console.log(logGroups);
  return logGroups;
};
```

- Para obter detalhes da API, consulte DescribeLogGroupsa Referência AWS SDK for JavaScript da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DescribeSubscriptionFilters`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `DescribeSubscriptionFilters`.

C++

SDK para C++

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Inclua os arquivos necessários.

```
#include <aws/core/Aws.h>
#include <aws/core/utils/Outcome.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/DescribeSubscriptionFiltersRequest.h>
#include <aws/logs/model/DescribeSubscriptionFiltersResult.h>
#include <iostream>
#include <iomanip>
```

Liste os filtros de assinatura.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::DescribeSubscriptionFiltersRequest request;
        request.SetLogGroupName(log_group);
        request.SetLimit(1);

        bool done = false;
        bool header = false;
        while (!done) {
            auto outcome = cwl.DescribeSubscriptionFilters(
                    request);
```

```
            if (!outcome.IsSuccess()) {
                std::cout << "Failed to describe CloudWatch subscription filters
"
                    << "for log group " << log_group << ": " <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header) {
                std::cout << std::left << std::setw(32) << "Name" <<
                    std::setw(64) << "FilterPattern" << std::setw(64) <<
                    "DestinationArn" << std::endl;
                header = true;
            }

            const auto &filters = outcome.GetResult().GetSubscriptionFilters();
            for (const auto &filter : filters) {
                std::cout << std::left << std::setw(32) <<
                    filter.GetFilterName() << std::setw(64) <<
                    filter.GetFilterPattern() << std::setw(64) <<
                    filter.GetDestinationArn() << std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Para obter detalhes da API, consulte DescribeSubscriptionFilters a Referência AWS SDK for C++ da API.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.DescribeSubscriptionFiltersResponse
import software.amazon.awssdk.services.cloudwatchlogs.model.SubscriptionFilter;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeSubscriptionFilters {
    public static void main(String[] args) {

        final String usage = """

                Usage:
                  <logGroup>

                Where:
                  logGroup - A log group name (for example, myloggroup).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String logGroup = args[0];
        CloudWatchLogsClient logs = CloudWatchLogsClient.builder()
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        describeFilters(logs, logGroup);
        logs.close();
    }
```

```
   public static void describeFilters(CloudWatchLogsClient logs, String
logGroup) {
       try {
           boolean done = false;
           String newToken = null;

           while (!done) {
               DescribeSubscriptionFiltersResponse response;
               if (newToken == null) {
                   DescribeSubscriptionFiltersRequest request =
DescribeSubscriptionFiltersRequest.builder()
                           .logGroupName(logGroup)
                           .limit(1).build();

                   response = logs.describeSubscriptionFilters(request);
               } else {
                   DescribeSubscriptionFiltersRequest request =
DescribeSubscriptionFiltersRequest.builder()
                           .nextToken(newToken)
                           .logGroupName(logGroup)
                           .limit(1).build();
                   response = logs.describeSubscriptionFilters(request);
               }

               for (SubscriptionFilter filter : response.subscriptionFilters())
{
                   System.out.printf("Retrieved filter with name %s, " +
"pattern %s " + "and destination arn %s",
                           filter.filterName(),
                           filter.filterPattern(),
                           filter.destinationArn());
               }

               if (response.nextToken() == null) {
                   done = true;
               } else {
                   newToken = response.nextToken();
               }
           }

       } catch (CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
```

```
        }
        System.out.printf("Done");
    }
}
```

- Para obter detalhes da API, consulte DescribeSubscriptionFilters a Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import { DescribeSubscriptionFiltersCommand } from "@aws-sdk/client-cloudwatch-
logs";
import { client } from "../libs/client.js";

const run = async () => {
  // This will return a list of all subscription filters in your account
  // matching the log group name.
  const command = new DescribeSubscriptionFiltersCommand({
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
    limit: 1,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Para obter detalhes da API, consulte DescribeSubscriptionFilters a Referência AWS SDK for JavaScript da API.

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  logGroupName: "GROUP_NAME",
  limit: 5,
};

cwl.describeSubscriptionFilters(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.subscriptionFilters);
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte DescribeSubscriptionFilters a Referência AWS SDK for JavaScript da API.

Kotlin

SDK para Kotlin

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
suspend fun describeFilters(logGroup: String) {
    val request =
        DescribeSubscriptionFiltersRequest {
            logGroupName = logGroup
            limit = 1
        }

    CloudWatchLogsClient { region = "us-west-2" }.use { cwlClient ->
        val response = cwlClient.describeSubscriptionFilters(request)
        response.subscriptionFilters?.forEach { filter ->
            println("Retrieved filter with name  ${filter.filterName} pattern
 ${filter.filterPattern} and destination ${filter.destinationArn}")
        }
    }
}
```

- Para obter detalhes da API, consulte a DescribeSubscriptionFiltersreferência da API AWS SDK for Kotlin.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`GetQueryResults`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `GetQueryResults`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Executar uma consulta grande

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
/**
 * Simple wrapper for the GetQueryResultsCommand.
 * @param {string} queryId
 */
_getQueryResults(queryId) {
  return this.client.send(new GetQueryResultsCommand({ queryId }));
}
```

- Para obter detalhes da API, consulte GetQueryResultsa Referência AWS SDK for JavaScript da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
```

```
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
                return results.get("results", [])
```

- Para obter detalhes da API, consulte a GetQueryResultsReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`PutSubscriptionFilter`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `PutSubscriptionFilter`.

C++

SDK para C++

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

Inclua os arquivos necessários.

```
#include <aws/core/Aws.h>
#include <aws/logs/CloudWatchLogsClient.h>
#include <aws/logs/model/PutSubscriptionFilterRequest.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Crie o filtro de assinatura.

```
        Aws::CloudWatchLogs::CloudWatchLogsClient cwl;
        Aws::CloudWatchLogs::Model::PutSubscriptionFilterRequest request;
        request.SetFilterName(filter_name);
        request.SetFilterPattern(filter_pattern);
        request.SetLogGroupName(log_group);
        request.SetDestinationArn(dest_arn);
        auto outcome = cwl.PutSubscriptionFilter(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch logs subscription filter "
                << filter_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch logs subscription " <<
                "filter " << filter_name << std::endl;
        }
```

- Para obter detalhes da API, consulte PutSubscriptionFiltera Referência AWS SDK for C++ da API.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.PutSubscriptionFilterRequest;

/**
 * Before running this code example, you need to grant permission to CloudWatch
 * Logs the right to execute your Lambda function.
 * To perform this task, you can use this CLI command:
 *
 * aws lambda add-permission --function-name "lamda1" --statement-id "lamda1"
 * --principal "logs.us-west-2.amazonaws.com" --action "lambda:InvokeFunction"
 * --source-arn "arn:aws:logs:us-west-2:111111111111:log-group:testgroup:*"
 * --source-account "111111111111"
 *
 * Make sure you replace the function name with your function name and replace
 * '111111111111' with your account details.
 * For more information, see "Subscription Filters with AWS Lambda" in the
 * Amazon CloudWatch Logs Guide.
 *
 *
 * Also, before running this Java V2 code example,set up your development
 * environment,including your credentials.
 *
 * For more information,see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 */

public class PutSubscriptionFilter {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <filter> <pattern> <logGroup> <functionArn>\s

                Where:
                  filter - A filter name (for example, myfilter).
                  pattern - A filter pattern (for example, ERROR).
```

```
                logGroup - A log group name (testgroup).
                functionArn - An AWS Lambda function ARN (for example,
arn:aws:lambda:us-west-2:111111111111:function:lambda1) .
               """;

       if (args.length != 4) {
           System.out.println(usage);
           System.exit(1);
       }

       String filter = args[0];
       String pattern = args[1];
       String logGroup = args[2];
       String functionArn = args[3];
       Region region = Region.US_WEST_2;
       CloudWatchLogsClient cwl = CloudWatchLogsClient.builder()
               .region(region)
               .build();

       putSubFilters(cwl, filter, pattern, logGroup, functionArn);
       cwl.close();
   }

   public static void putSubFilters(CloudWatchLogsClient cwl,
           String filter,
           String pattern,
           String logGroup,
           String functionArn) {

       try {
           PutSubscriptionFilterRequest request =
PutSubscriptionFilterRequest.builder()
                   .filterName(filter)
                   .filterPattern(pattern)
                   .logGroupName(logGroup)
                   .destinationArn(functionArn)
                   .build();

           cwl.putSubscriptionFilter(request);
           System.out.printf(
                   "Successfully created CloudWatch logs subscription filter
%s",
                   filter);
```

```
        } catch (CloudWatchLogsException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte PutSubscriptionFiltera Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
import { PutSubscriptionFilterCommand } from "@aws-sdk/client-cloudwatch-logs";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new PutSubscriptionFilterCommand({
    // An ARN of a same-account Kinesis stream, Kinesis Firehose
    // delivery stream, or Lambda function.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
SubscriptionFilters.html
    destinationArn: process.env.CLOUDWATCH_LOGS_DESTINATION_ARN,

    // A name for the filter.
    filterName: process.env.CLOUDWATCH_LOGS_FILTER_NAME,

    // A filter pattern for subscribing to a filtered stream of log events.
    // https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/
FilterAndPatternSyntax.html
    filterPattern: process.env.CLOUDWATCH_LOGS_FILTER_PATTERN,
```

```
    // The name of the log group. Messages in this group matching the filter
 pattern
    // will be sent to the destination ARN.
    logGroupName: process.env.CLOUDWATCH_LOGS_LOG_GROUP,
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

- Para obter detalhes da API, consulte PutSubscriptionFiltera Referência AWS SDK for JavaScript da API.

SDK para JavaScript (v2)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no Repositório de exemplos de código da AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create the CloudWatchLogs service object
var cwl = new AWS.CloudWatchLogs({ apiVersion: "2014-03-28" });

var params = {
  destinationArn: "LAMBDA_FUNCTION_ARN",
  filterName: "FILTER_NAME",
  filterPattern: "ERROR",
  logGroupName: "LOG_GROUP",
};

cwl.putSubscriptionFilter(params, function (err, data) {
```

```
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Para obter mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.
- Para obter detalhes da API, consulte PutSubscriptionFiltera Referência AWS SDK for JavaScript da API.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`StartLiveTail`** com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `StartLiveTail`.

.NET

AWS SDK for .NET

Inclua os arquivos necessários.

```
using Amazon;
using Amazon.CloudWatchLogs;
using Amazon.CloudWatchLogs.Model;
```

Inicie a sessão do Live Tail.

```
            var client = new AmazonCloudWatchLogsClient();
            var request = new StartLiveTailRequest
            {
                LogGroupIdentifiers = logGroupIdentifiers,
                LogStreamNames = logStreamNames,
                LogEventFilterPattern = filterPattern,
```

```
            };

            var response = await client.StartLiveTailAsync(request);

            // Catch if request fails
            if (response.HttpStatusCode != System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine("Failed to start live tail session");
                return;
            }
```

Você pode lidar com os eventos da sessão do Live Tail de duas maneiras:

```
            /* Method 1
            * 1). Asynchronously loop through the event stream
            * 2). Set a timer to dispose the stream and stop the Live Tail
session at the end.
            */
            var eventStream = response.ResponseStream;
            var task = Task.Run(() =>
            {
                foreach (var item in eventStream)
                {
                    if (item is LiveTailSessionUpdate liveTailSessionUpdate)
                    {
                        foreach (var sessionResult in
liveTailSessionUpdate.SessionResults)
                        {
                            Console.WriteLine("Message : {0}",
sessionResult.Message);
                        }
                    }
                    if (item is LiveTailSessionStart)
                    {
                        Console.WriteLine("Live Tail session started");
                    }
                    // On-stream exceptions are processed here
                    if (item is CloudWatchLogsEventStreamException)
                    {
                        Console.WriteLine($"ERROR: {item}");
                    }
                }
```

```
            });
            // Close the stream to stop the session after a timeout
            if (!task.Wait(TimeSpan.FromSeconds(10))){
                eventStream.Dispose();
                Console.WriteLine("End of line");
            }
```

```
            /* Method 2
            * 1). Add event handlers to each event variable
            * 2). Start processing the stream and wait for a timeout using
AutoResetEvent
            */
            AutoResetEvent endEvent = new AutoResetEvent(false);
            var eventStream = response.ResponseStream;
            using (eventStream) // automatically disposes the stream to stop the
session after execution finishes
            {
                eventStream.SessionStartReceived += (sender, e) =>
                {
                    Console.WriteLine("LiveTail session started");
                };
                eventStream.SessionUpdateReceived += (sender, e) =>
                {
                    foreach (LiveTailSessionLogEvent logEvent in
e.EventStreamEvent.SessionResults){
                        Console.WriteLine("Message: {0}", logEvent.Message);
                    }
                };
                // On-stream exceptions are captured here
                eventStream.ExceptionReceived += (sender, e) =>
                {
                    Console.WriteLine($"ERROR:
{e.EventStreamException.Message}");
                };

                eventStream.StartProcessing();
                // Stream events for this amount of time.
                endEvent.WaitOne(TimeSpan.FromSeconds(10));
                Console.WriteLine("End of line");
            }
```

- Para obter detalhes da API, consulte StartLiveTaila Referência AWS SDK for .NET da API.

Go

SDK para Go V2

Inclua os arquivos necessários.

```
import (
 "context"
 "log"
 "time"

 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs"
 "github.com/aws/aws-sdk-go-v2/service/cloudwatchlogs/types"
)
```

Gerencie os eventos da sessão do Live Tail.

```
func handleEventStreamAsync(stream *cloudwatchlogs.StartLiveTailEventStream) {
 eventsChan := stream.Events()
 for {
  event := <-eventsChan
  switch e := event.(type) {
  case *types.StartLiveTailResponseStreamMemberSessionStart:
   log.Println("Received SessionStart event")
  case *types.StartLiveTailResponseStreamMemberSessionUpdate:
   for _, logEvent := range e.Value.SessionResults {
    log.Println(*logEvent.Message)
   }
  default:
   // Handle on-stream exceptions
   if err := stream.Err(); err != nil {
    log.Fatalf("Error occured during streaming: %v", err)
   } else if event == nil {
    log.Println("Stream is Closed")
    return
   } else {
    log.Fatalf("Unknown event type: %T", e)
   }
```

```
  }
 }
}
```

Inicie a sessão do Live Tail.

```
cfg, err := config.LoadDefaultConfig(context.TODO())
if err != nil {
 panic("configuration error, " + err.Error())
}
client := cloudwatchlogs.NewFromConfig(cfg)

request := &cloudwatchlogs.StartLiveTailInput{
 LogGroupIdentifiers:   logGroupIdentifiers,
 LogStreamNames:        logStreamNames,
 LogEventFilterPattern: logEventFilterPattern,
}

response, err := client.StartLiveTail(context.TODO(), request)
// Handle pre-stream Exceptions
if err != nil {
 log.Fatalf("Failed to start streaming: %v", err)
}

// Start a Goroutine to handle events over stream
stream := response.GetStream()
go handleEventStreamAsync(stream)
```

Interrompa a sessão do Live Tail após um período decorrido.

```
// Close the stream (which ends the session) after a timeout
time.Sleep(10 * time.Second)
stream.Close()
log.Println("Event stream closed")
```

- Para obter detalhes da API, consulte StartLiveTaila Referência AWS SDK for Go da API.

Java

SDK para Java 2.x

Inclua os arquivos necessários.

```
import io.reactivex.FlowableSubscriber;
import io.reactivex.annotations.NonNull;
import org.reactivestreams.Subscription;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsAsyncClient;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionStart;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.LiveTailSessionUpdate;
import software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailRequest;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseHandler;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.CloudWatchLogsException;
import
 software.amazon.awssdk.services.cloudwatchlogs.model.StartLiveTailResponseStream;

import java.util.Date;
import java.util.List;
import java.util.concurrent.atomic.AtomicReference;
```

Gerencie os eventos da sessão do Live Tail.

```
    private static StartLiveTailResponseHandler
 getStartLiveTailResponseStreamHandler(
            AtomicReference<Subscription> subscriptionAtomicReference) {
        return StartLiveTailResponseHandler.builder()
            .onResponse(r -> System.out.println("Received initial response"))
            .onError(throwable -> {
                CloudWatchLogsException e = (CloudWatchLogsException)
 throwable.getCause();
                System.err.println(e.awsErrorDetails().errorMessage());
                System.exit(1);
            })
            .subscriber(() -> new FlowableSubscriber<>() {
```

```
                @Override
                public void onSubscribe(@NonNull Subscription s) {
                    subscriptionAtomicReference.set(s);
                    s.request(Long.MAX_VALUE);
                }

                @Override
                public void onNext(StartLiveTailResponseStream event) {
                    if (event instanceof LiveTailSessionStart) {
                        LiveTailSessionStart sessionStart =
(LiveTailSessionStart) event;
                        System.out.println(sessionStart);
                    } else if (event instanceof LiveTailSessionUpdate) {
                        LiveTailSessionUpdate sessionUpdate =
(LiveTailSessionUpdate) event;
                        List<LiveTailSessionLogEvent> logEvents =
sessionUpdate.sessionResults();
                        logEvents.forEach(e -> {
                            long timestamp = e.timestamp();
                            Date date = new Date(timestamp);
                            System.out.println("[" + date + "] " + e.message());
                        });
                    } else {
                        throw CloudWatchLogsException.builder().message("Unknown
event type").build();
                    }
                }

                @Override
                public void onError(Throwable throwable) {
                    System.out.println(throwable.getMessage());
                    System.exit(1);
                }

                @Override
                public void onComplete() {
                    System.out.println("Completed Streaming Session");
                }
            })
            .build();
    }
```

Inicie a sessão do Live Tail.

```
        CloudWatchLogsAsyncClient cloudWatchLogsAsyncClient =
                CloudWatchLogsAsyncClient.builder()
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();

        StartLiveTailRequest request =
                StartLiveTailRequest.builder()
                    .logGroupIdentifiers(logGroupIdentifiers)
                    .logStreamNames(logStreamNames)
                    .logEventFilterPattern(logEventFilterPattern)
                    .build();

        /* Create a reference to store the subscription */
        final AtomicReference<Subscription> subscriptionAtomicReference = new
AtomicReference<>(null);

        cloudWatchLogsAsyncClient.startLiveTail(request,
getStartLiveTailResponseStreamHandler(subscriptionAtomicReference));
```

Interrompa a sessão do Live Tail após um período decorrido.

```
        /* Set a timeout for the session and cancel the subscription. This will:
         * 1). Close the stream
         * 2). Stop the Live Tail session
         */
        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        if (subscriptionAtomicReference.get() != null) {
            subscriptionAtomicReference.get().cancel();
            System.out.println("Subscription to stream closed");
        }
```

- Para obter detalhes da API, consulte StartLiveTaila Referência AWS SDK for Java 2.x da API.

JavaScript

SDK para JavaScript (v3)

Inclua os arquivos necessários.

```
import { CloudWatchLogsClient, StartLiveTailCommand } from "@aws-sdk/client-
cloudwatch-logs";
```

Gerencie os eventos da sessão do Live Tail.

```
async function handleResponseAsync(response) {
    try {
      for await (const event of response.responseStream) {
        if (event.sessionStart !== undefined) {
          console.log(event.sessionStart);
        } else if (event.sessionUpdate !== undefined) {
          for (const logEvent of event.sessionUpdate.sessionResults) {
            const timestamp = logEvent.timestamp;
            const date = new Date(timestamp);
            console.log("[" + date + "] " + logEvent.message);
          }
        } else {
            console.error("Unknown event type");
        }
      }
    } catch (err) {
        // On-stream exceptions are captured here
        console.error(err)
    }
}
```

Inicie a sessão do Live Tail.

```
    const client = new CloudWatchLogsClient();

    const command = new StartLiveTailCommand({
        logGroupIdentifiers: logGroupIdentifiers,
        logStreamNames: logStreamNames,
        logEventFilterPattern: filterPattern
    });
```

```
    try{
        const response = await client.send(command);
        handleResponseAsync(response);
    } catch (err){
        // Pre-stream exceptions are captured here
        console.log(err);
    }
```

Interrompa a sessão do Live Tail após um período decorrido.

```
    /* Set a timeout to close the client. This will stop the Live Tail session.
*/
    setTimeout(function() {
        console.log("Client timeout");
        client.destroy();
      }, 10000);
```

- Para obter detalhes da API, consulte StartLiveTaila Referência AWS SDK for JavaScript da API.

Kotlin

SDK para Kotlin

Inclua os arquivos necessários.

```
import aws.sdk.kotlin.services.cloudwatchlogs.CloudWatchLogsClient
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailRequest
import aws.sdk.kotlin.services.cloudwatchlogs.model.StartLiveTailResponseStream
import kotlinx.coroutines.flow.takeWhile
```

Inicie a sessão do Live Tail.

```
    val client = CloudWatchLogsClient.fromEnvironment()

    val request = StartLiveTailRequest {
        logGroupIdentifiers = logGroupIdentifiersVal
        logStreamNames = logStreamNamesVal
```

```
        logEventFilterPattern = logEventFilterPatternVal
    }

    val startTime = System.currentTimeMillis()

    try {
        client.startLiveTail(request) { response ->
            val stream = response.responseStream
            if (stream != null) {
                /* Set a timeout to unsubcribe from the flow. This will:
                * 1). Close the stream
                * 2). Stop the Live Tail session
                */
                stream.takeWhile { System.currentTimeMillis() - startTime <
10000 }.collect { value ->
                    if (value is StartLiveTailResponseStream.SessionStart) {
                        println(value.asSessionStart())
                    } else if (value is
StartLiveTailResponseStream.SessionUpdate) {
                        for (e in value.asSessionUpdate().sessionResults!!) {
                            println(e)
                        }
                    } else {
                        throw IllegalArgumentException("Unknown event type")
                    }
                }
            } else {
                throw IllegalArgumentException("No response stream")
            }
        }
    } catch (e: Exception) {
        println("Exception occurred during StartLiveTail: $e")
        System.exit(1)
    }
```

- Para obter detalhes da API, consulte a StartLiveTailreferência da API AWS SDK for Kotlin.

Python

SDK para Python (Boto3)

Inclua os arquivos necessários.

```
import boto3
import time
from datetime import datetime
```

Inicie a sessão do Live Tail.

```
    # Initialize the client
    client = boto3.client('logs')

    start_time = time.time()

    try:
        response = client.start_live_tail(
            logGroupIdentifiers=log_group_identifiers,
            logStreamNames=log_streams,
            logEventFilterPattern=filter_pattern
        )
        event_stream = response['responseStream']
        # Handle the events streamed back in the response
        for event in event_stream:
            # Set a timeout to close the stream.
            # This will end the Live Tail session.
            if (time.time() - start_time >= 10):
                event_stream.close()
                break
            # Handle when session is started
            if 'sessionStart' in event:
                session_start_event = event['sessionStart']
                print(session_start_event)
            # Handle when log event is given in a session update
            elif 'sessionUpdate' in event:
                log_events = event['sessionUpdate']['sessionResults']
                for log_event in log_events:
                    print('[{date}]
 {log}'.format(date=datetime.fromtimestamp(log_event['timestamp']/1000),log=log_event['me
            else:
                # On-stream exceptions are captured here
                raise RuntimeError(str(event))
    except Exception as e:
        print(e)
```

- Para obter detalhes da API, consulte a StartLiveTailReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use `StartQuery` com um AWS SDK ou CLI

Os exemplos de códigos a seguir mostram como usar `StartQuery`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação no contexto no seguinte exemplo de código:

- Executar uma consulta grande

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
/**
 * Wrapper for the StartQueryCommand. Uses a static query string
 * for consistency.
 * @param {[Date, Date]} dateRange
 * @param {number} maxLogs
 * @returns {Promise<{ queryId: string }>}
 */
async _startQuery([startDate, endDate], maxLogs = 10000) {
  try {
    return await this.client.send(
      new StartQueryCommand({
        logGroupNames: this.logGroupNames,
        queryString: "fields @timestamp, @message | sort @timestamp asc",
        startTime: startDate.valueOf(),
```

```
          endTime: endDate.valueOf(),
          limit: maxLogs,
        }),
      );
    } catch (err) {
      /** @type {string} */
      const message = err.message;
      if (message.startsWith("Query's end date and time")) {
        // This error indicates that the query's start or end date occur
        // before the log group was created.
        throw new DateOutOfBoundsError(message);
      }

      throw err;
    }
  }
```

- Para obter detalhes da API, consulte StartQuerya Referência AWS SDK for JavaScript da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

```
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
```

```
        client = boto3.client("logs")
        try:
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
                response = client.start_query(
                    logGroupName=self.log_groups,
                    startTime=start_time,
                    endTime=end_time,
                    queryString="fields @timestamp, @message | sort @timestamp
asc",
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
```

```
        :type max_logs: int
        :return: The query ID as a string.
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
                logGroupName=self.log_groups,
                startTime=start_time,
                endTime=end_time,
                queryString="fields @timestamp, @message | sort @timestamp asc",
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")
```

- Para obter detalhes da API, consulte a StartQueryReferência da API AWS SDK for Python (Boto3).

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cenários para CloudWatch registros usando AWS SDKs

Os exemplos de código a seguir mostram como implementar cenários comuns em CloudWatch Logs com AWS SDKs. Esses cenários mostram como realizar tarefas específicas chamando várias funções no CloudWatch Logs. Cada cenário inclui um link para GitHub, onde você pode encontrar instruções sobre como configurar e executar o código.

Exemplos

- Use CloudWatch registros para executar uma consulta grande

Use CloudWatch registros para executar uma consulta grande

Os exemplos de código a seguir mostram como usar o CloudWatch Logs para consultar mais de 10.000 registros.

JavaScript

SDK para JavaScript (v3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Esse é o ponto de entrada.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { CloudWatchLogsClient } from "@aws-sdk/client-cloudwatch-logs";
import { CloudWatchQuery } from "./cloud-watch-query.js";

console.log("Starting a recursive query...");

if (!process.env.QUERY_START_DATE || !process.env.QUERY_END_DATE) {
  throw new Error(
    "QUERY_START_DATE and QUERY_END_DATE environment variables are required.",
  );
}

const cloudWatchQuery = new CloudWatchQuery(new CloudWatchLogsClient({}), {
  logGroupNames: ["/workflows/cloudwatch-logs/large-query"],
  dateRange: [
    new Date(parseInt(process.env.QUERY_START_DATE)),
    new Date(parseInt(process.env.QUERY_END_DATE)),
  ],
});

await cloudWatchQuery.run();
```

```
console.log(
  `Queries finished in ${cloudWatchQuery.secondsElapsed} seconds.\nTotal logs
 found: ${cloudWatchQuery.results.length}`,
);
```

Essa é uma classe que divide as consultas em várias etapas, se necessário.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import {
  StartQueryCommand,
  GetQueryResultsCommand,
} from "@aws-sdk/client-cloudwatch-logs";
import { splitDateRange } from "@aws-doc-sdk-examples/lib/utils/util-date.js";
import { retry } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";

class DateOutOfBoundsError extends Error {}

export class CloudWatchQuery {
  /**
   * Run a query for all CloudWatch Logs within a certain date range.
   * CloudWatch logs return a max of 10,000 results. This class
   * performs a binary search across all of the logs in the provided
   * date range if a query returns the maximum number of results.
   *
   * @param {import('@aws-sdk/client-cloudwatch-logs').CloudWatchLogsClient}
 client
   * @param {{ logGroupNames: string[], dateRange: [Date, Date], queryConfig:
 { limit: number } }} config
   */
  constructor(client, { logGroupNames, dateRange, queryConfig }) {
    this.client = client;
    /**
     * All log groups are queried.
     */
    this.logGroupNames = logGroupNames;

    /**
     * The inclusive date range that is queried.
     */
    this.dateRange = dateRange;
```

```
    /**
     * CloudWatch Logs never returns more than 10,000 logs.
     */
    this.limit = queryConfig?.limit ?? 10000;

    /**
     * @type {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]}
     */
    this.results = [];
  }

  /**
   * Run the query.
   */
  async run() {
    this.secondsElapsed = 0;
    const start = new Date();
    this.results = await this._largeQuery(this.dateRange);
    const end = new Date();
    this.secondsElapsed = (end - start) / 1000;
    return this.results;
  }

  /**
   * Recursively query for logs.
   * @param {[Date, Date]} dateRange
   * @returns {Promise<import("@aws-sdk/client-cloudwatch-logs").ResultField[]
[]>}
   */
  async _largeQuery(dateRange) {
    const logs = await this._query(dateRange, this.limit);

    console.log(
      `Query date range: ${dateRange
        .map((d) => d.toISOString())
        .join(" to ")}. Found ${logs.length} logs.`,
    );

    if (logs.length < this.limit) {
      return logs;
    }

    const lastLogDate = this._getLastLogDate(logs);
```

```
    const offsetLastLogDate = new Date(lastLogDate);
    offsetLastLogDate.setMilliseconds(lastLogDate.getMilliseconds() + 1);
    const subDateRange = [offsetLastLogDate, dateRange[1]];
    const [r1, r2] = splitDateRange(subDateRange);
    const results = await Promise.all([
      this._largeQuery(r1),
      this._largeQuery(r2),
    ]);
    return [logs, ...results].flat();
  }

  /**
   * Find the most recent log in a list of logs.
   * @param {import("@aws-sdk/client-cloudwatch-logs").ResultField[][]} logs
   */
  _getLastLogDate(logs) {
    const timestamps = logs
      .map(
        (log) =>
          log.find((fieldMeta) => fieldMeta.field === "@timestamp")?.value,
      )
      .filter((t) => !!t)
      .map((t) => `${t}Z`)
      .sort();

    if (!timestamps.length) {
      throw new Error("No timestamp found in logs.");
    }

    return new Date(timestamps[timestamps.length - 1]);
  }

  // snippet-start:[javascript.v3.cloudwatch-logs.actions.GetQueryResults]
  /**
   * Simple wrapper for the GetQueryResultsCommand.
   * @param {string} queryId
   */
  _getQueryResults(queryId) {
    return this.client.send(new GetQueryResultsCommand({ queryId }));
  }
  // snippet-end:[javascript.v3.cloudwatch-logs.actions.GetQueryResults]

  /**
   * Starts a query and waits for it to complete.
```

```
   * @param {[Date, Date]} dateRange
   * @param {number} maxLogs
   */
  async _query(dateRange, maxLogs) {
    try {
      const { queryId } = await this._startQuery(dateRange, maxLogs);
      const { results } = await this._waitUntilQueryDone(queryId);
      return results ?? [];
    } catch (err) {
      /**
       * This error is thrown when StartQuery returns an error indicating
       * that the query's start or end date occur before the log group was
       * created.
       */
      if (err instanceof DateOutOfBoundsError) {
        return [];
      } else {
        throw err;
      }
    }
  }

  // snippet-start:[javascript.v3.cloudwatch-logs.actions.StartQuery]
  /**
   * Wrapper for the StartQueryCommand. Uses a static query string
   * for consistency.
   * @param {[Date, Date]} dateRange
   * @param {number} maxLogs
   * @returns {Promise<{ queryId: string }>}
   */
  async _startQuery([startDate, endDate], maxLogs = 10000) {
    try {
      return await this.client.send(
        new StartQueryCommand({
          logGroupNames: this.logGroupNames,
          queryString: "fields @timestamp, @message | sort @timestamp asc",
          startTime: startDate.valueOf(),
          endTime: endDate.valueOf(),
          limit: maxLogs,
        }),
      );
    } catch (err) {
      /** @type {string} */
      const message = err.message;
```

```
      if (message.startsWith("Query's end date and time")) {
        // This error indicates that the query's start or end date occur
        // before the log group was created.
        throw new DateOutOfBoundsError(message);
      }

      throw err;
    }
  }
  // snippet-end:[javascript.v3.cloudwatch-logs.actions.StartQuery]

  /**
   * Call GetQueryResultsCommand until the query is done.
   * @param {string} queryId
   */
  _waitUntilQueryDone(queryId) {
    const getResults = async () => {
      const results = await this._getQueryResults(queryId);
      const queryDone = [
        "Complete",
        "Failed",
        "Cancelled",
        "Timeout",
        "Unknown",
      ].includes(results.status);

      return { queryDone, results };
    };

    return retry(
      { intervalInMs: 1000, maxRetries: 60, quiet: true },
      async () => {
        const { queryDone, results } = await getResults();
        if (!queryDone) {
          throw new Error("Query not done.");
        }

        return results;
      },
    );
  }
}
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK for JavaScript .
  - GetQueryResults
  - StartQuery

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e veja como configurar e executar no AWS Code Examples Repository.

Esse arquivo invoca um módulo de exemplo para gerenciar CloudWatch consultas com mais de 10.000 resultados.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import logging
import os
import sys

import boto3
from botocore.config import Config

from cloudwatch_query import CloudWatchQuery
from date_utilities import DateUtilities

# Configure logging at the module level.
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s - %(levelname)s - %(filename)s:%(lineno)d - %(message)s",
)


class CloudWatchLogsQueryRunner:
    def __init__(self):
        """
```

```
        Initializes the CloudWatchLogsQueryRunner class by setting up date
utilities
        and creating a CloudWatch Logs client with retry configuration.
        """
        self.date_utilities = DateUtilities()
        self.cloudwatch_logs_client = self.create_cloudwatch_logs_client()

    def create_cloudwatch_logs_client(self):
        """
        Creates and returns a CloudWatch Logs client with a specified retry
configuration.

        :return: A CloudWatch Logs client instance.
        :rtype: boto3.client
        """
        try:
            return boto3.client("logs", config=Config(retries={"max_attempts":
10}))
        except Exception as e:
            logging.error(f"Failed to create CloudWatch Logs client: {e}")
            sys.exit(1)

    def fetch_environment_variables(self):
        """
        Fetches and validates required environment variables for query start and
end dates.

        :return: Tuple of query start date and end date as integers.
        :rtype: tuple
        :raises SystemExit: If required environment variables are missing or
invalid.
        """
        try:
            query_start_date = int(os.environ["QUERY_START_DATE"])
            query_end_date = int(os.environ["QUERY_END_DATE"])
        except KeyError:
            logging.error(
                "Both QUERY_START_DATE and QUERY_END_DATE environment variables
are required."
            )
            sys.exit(1)
        except ValueError as e:
            logging.error(f"Error parsing date environment variables: {e}")
            sys.exit(1)
```

```
        return query_start_date, query_end_date

    def convert_dates_to_iso8601(self, start_date, end_date):
        """
        Converts UNIX timestamp dates to ISO 8601 format using DateUtilities.

        :param start_date: The start date in UNIX timestamp.
        :type start_date: int
        :param end_date: The end date in UNIX timestamp.
        :type end_date: int
        :return: Start and end dates in ISO 8601 format.
        :rtype: tuple
        """
        start_date_iso8601 =
 self.date_utilities.convert_unix_timestamp_to_iso8601(
            start_date
        )
        end_date_iso8601 = self.date_utilities.convert_unix_timestamp_to_iso8601(
            end_date
        )
        return start_date_iso8601, end_date_iso8601

    def execute_query(
        self,
        start_date_iso8601,
        end_date_iso8601,
        log_group="/workflows/cloudwatch-logs/large-query",
    ):
        """
        Creates a CloudWatchQuery instance and executes the query with provided
 date range.

        :param start_date_iso8601: The start date in ISO 8601 format.
        :type start_date_iso8601: str
        :param end_date_iso8601: The end date in ISO 8601 format.
        :type end_date_iso8601: str
        :param log_group: Log group to search: "/workflows/cloudwatch-logs/large-
query"
        :type log_group: str
        """
        cloudwatch_query = CloudWatchQuery(
            [start_date_iso8601, end_date_iso8601],
        )
```

```
        cloudwatch_query.query_logs((start_date_iso8601, end_date_iso8601))
        logging.info("Query executed successfully.")
        logging.info(
            f"Queries completed in {cloudwatch_query.query_duration} seconds.
 Total logs found: {len(cloudwatch_query.query_results)}"
        )


def main():
    """
    Main function to start a recursive CloudWatch logs query.
    Fetches required environment variables, converts dates, and executes the
 query.
    """
    logging.info("Starting a recursive CloudWatch logs query...")
    runner = CloudWatchLogsQueryRunner()
    query_start_date, query_end_date = runner.fetch_environment_variables()
    start_date_iso8601 = DateUtilities.convert_unix_timestamp_to_iso8601(
        query_start_date
    )
    end_date_iso8601 =
 DateUtilities.convert_unix_timestamp_to_iso8601(query_end_date)
    runner.execute_query(start_date_iso8601, end_date_iso8601)


if __name__ == "__main__":
    main()
```

Este módulo processa CloudWatch consultas com mais de 10.000 resultados.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import logging
import time
from datetime import datetime
import threading
import boto3

from date_utilities import DateUtilities


class DateOutOfBoundsError(Exception):
```

```
    """Exception raised when the date range for a query is out of bounds."""

    pass


class CloudWatchQuery:
    """
    A class to query AWS CloudWatch logs within a specified date range.

    :ivar date_range: Start and end datetime for the query.
    :vartype date_range: tuple
    :ivar limit: Maximum number of log entries to return.
    :vartype limit: int
    """

    def __init__(self, date_range):
        self.lock = threading.Lock()
        self.log_groups = "/workflows/cloudwatch-logs/large-query"
        self.query_results = []
        self.date_range = date_range
        self.query_duration = None
        self.datetime_format = "%Y-%m-%d %H:%M:%S.%f"
        self.date_utilities = DateUtilities()
        self.limit = 10000

    def query_logs(self, date_range):
        """
        Executes a CloudWatch logs query for a specified date range and
 calculates the execution time of the query.

        :return: A batch of logs retrieved from the CloudWatch logs query.
        :rtype: list
        """
        start_time = datetime.now()

        start_date, end_date = self.date_utilities.normalize_date_range_format(
            date_range, from_format="unix_timestamp", to_format="datetime"
        )

        logging.info(
            f"Original query:"
            f"\n       START:    {start_date}"
            f"\n       END:      {end_date}"
        )
```

```
        self.recursive_query((start_date, end_date))
        end_time = datetime.now()
        self.query_duration = (end_time - start_time).total_seconds()

    def recursive_query(self, date_range):
        """
        Processes logs within a given date range, fetching batches of logs
recursively if necessary.

        :param date_range: The date range to fetch logs for, specified as a tuple
(start_timestamp, end_timestamp).
        :type date_range: tuple
        :return: None if the recursive fetching is continued or stops when the
final batch of logs is processed.
                 Although it doesn't explicitly return the query results, this
method accumulates all fetched logs
                 in the `self.query_results` attribute.
        :rtype: None
        """
        batch_of_logs = self.perform_query(date_range)
        # Add the batch to the accumulated logs
        with self.lock:
            self.query_results.extend(batch_of_logs)
        if len(batch_of_logs) == self.limit:
            logging.info(f"Fetched {self.limit}, checking for more...")
            most_recent_log = self.find_most_recent_log(batch_of_logs)
            most_recent_log_timestamp = next(
                item["value"]
                for item in most_recent_log
                if item["field"] == "@timestamp"
            )
            new_range = (most_recent_log_timestamp, date_range[1])
            midpoint = self.date_utilities.find_middle_time(new_range)

            first_half_thread = threading.Thread(
                target=self.recursive_query,
                args=((most_recent_log_timestamp, midpoint),),
            )
            second_half_thread = threading.Thread(
                target=self.recursive_query, args=((midpoint, date_range[1]),)
            )

            first_half_thread.start()
            second_half_thread.start()
```

```
            first_half_thread.join()
            second_half_thread.join()

    def find_most_recent_log(self, logs):
        """
        Search a list of log items and return most recent log entry.
        :param logs: A list of logs to analyze.
        :return: log
        :type :return List containing log item details
        """
        most_recent_log = None
        most_recent_date = "1970-01-01 00:00:00.000"

        for log in logs:
            for item in log:
                if item["field"] == "@timestamp":
                    logging.debug(f"Compared: {item['value']} to
{most_recent_date}")
                    if (
                        self.date_utilities.compare_dates(
                            item["value"], most_recent_date
                        )
                        == item["value"]
                    ):
                        logging.debug(f"New most recent: {item['value']}")
                        most_recent_date = item["value"]
                        most_recent_log = log
        logging.info(f"Most recent log date of batch: {most_recent_date}")
        return most_recent_log

    # snippet-start:[python.example_code.cloudwatch_logs.start_query]
    def perform_query(self, date_range):
        """
        Performs the actual CloudWatch log query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :return: A list containing the query results.
        :rtype: list
        """
        client = boto3.client("logs")
        try:
```

```
            try:
                start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
                )
                end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
                )
                response = client.start_query(
                    logGroupName=self.log_groups,
                    startTime=start_time,
                    endTime=end_time,
                    queryString="fields @timestamp, @message | sort @timestamp
asc",
                    limit=self.limit,
                )
                query_id = response["queryId"]
            except client.exceptions.ResourceNotFoundException as e:
                raise DateOutOfBoundsError(f"Resource not found: {e}")
            while True:
                time.sleep(1)
                results = client.get_query_results(queryId=query_id)
                if results["status"] in [
                    "Complete",
                    "Failed",
                    "Cancelled",
                    "Timeout",
                    "Unknown",
                ]:
                    return results.get("results", [])
        except DateOutOfBoundsError:
            return []

    def _initiate_query(self, client, date_range, max_logs):
        """
        Initiates the CloudWatch logs query.

        :param date_range: A tuple representing the start and end datetime for
the query.
        :type date_range: tuple
        :param max_logs: The maximum number of logs to retrieve.
        :type max_logs: int
        :return: The query ID as a string.
```

```
        :rtype: str
        """
        try:
            start_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[0])
            )
            end_time = round(

self.date_utilities.convert_iso8601_to_unix_timestamp(date_range[1])
            )
            response = client.start_query(
                logGroupName=self.log_groups,
                startTime=start_time,
                endTime=end_time,
                queryString="fields @timestamp, @message | sort @timestamp asc",
                limit=max_logs,
            )
            return response["queryId"]
        except client.exceptions.ResourceNotFoundException as e:
            raise DateOutOfBoundsError(f"Resource not found: {e}")

    # snippet-end:[python.example_code.cloudwatch_logs.start_query]

    # snippet-start:[python.example_code.cloudwatch_logs.get_query_results]
    def _wait_for_query_results(self, client, query_id):
        """
        Waits for the query to complete and retrieves the results.

        :param query_id: The ID of the initiated query.
        :type query_id: str
        :return: A list containing the results of the query.
        :rtype: list
        """
        while True:
            time.sleep(1)
            results = client.get_query_results(queryId=query_id)
            if results["status"] in [
                "Complete",
                "Failed",
                "Cancelled",
                "Timeout",
                "Unknown",
            ]:
```

```
                return results.get("results", [])

    # snippet-end:[python.example_code.cloudwatch_logs.get_query_results]
```

- Para obter detalhes da API, consulte os tópicos a seguir na Referência da API AWS SDK para Python (Boto3).
  - GetQueryResults
  - StartQuery

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de vários serviços para CloudWatch registros usando AWS SDKs

Os exemplos de aplicativos a seguir usam AWS SDKs para combinar CloudWatch registros com outros Serviços da AWS. Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções sobre como configurar e executar o aplicativo.

Exemplos

- Usar eventos programados para chamar uma função do Lambda

Usar eventos programados para chamar uma função do Lambda

Os exemplos de código a seguir mostram como criar uma AWS Lambda função invocada por um evento EventBridge agendado pela Amazon.

Python

SDK para Python (Boto3)

Este exemplo mostra como registrar uma AWS Lambda função como alvo de um EventBridge evento programado da Amazon. O manipulador do Lambda grava uma mensagem amigável e os dados completos do evento no Amazon CloudWatch Logs para recuperação posterior.

- Implanta uma função do Lambda.

- Cria um evento EventBridge agendado e torna a função Lambda o alvo.
- Concede permissão para permitir a EventBridge invocação da função Lambda.
- Imprime os dados mais recentes do CloudWatch Logs para mostrar o resultado das invocações programadas.
- Limpa todos os recursos criados durante a demonstração.

Este exemplo é melhor visualizado em GitHub. Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Serviços utilizados neste exemplo

- CloudWatch Registros
- EventBridge
- Lambda

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando CloudWatch registros com um AWS SDK. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Segurança no Amazon CloudWatch Logs

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

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

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade aplicáveis WorkSpaces, consulte AWS Serviços no escopo do programa de conformidade AWS .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço 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

Essa documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon CloudWatch Logs. Ele mostra como configurar o Amazon CloudWatch Logs para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos do CloudWatch Logs.

Conteúdo

- Proteção de dados no Amazon CloudWatch Logs
- Gerenciamento de identidade e acesso para Amazon CloudWatch Logs
- Validação de conformidade para Amazon CloudWatch Logs
- Resiliência no Amazon CloudWatch Logs
- Segurança da infraestrutura no Amazon CloudWatch Logs
- Usando CloudWatch registros com endpoints VPC de interface

Proteção de dados no Amazon CloudWatch Logs

> **Note**
>
> Além das informações a seguir sobre proteção geral de dados no AWS, o CloudWatch Logs também permite que você proteja dados confidenciais em eventos de log mascarando-os. Para ter mais informações, consulte Ajude a proteger dados de log confidenciais com mascaramento.

O modelo de responsabilidade AWS compartilhada de se aplica à proteção de dados no Amazon CloudWatch Logs. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para ter mais informações sobre a proteção de dados na Europa, consulte a AWS postagem do blog Shared Responsibility Model and GDPR no AWS Blog de segurança da.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para ter mais

informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de email dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso inclui quando você trabalha com CloudWatch Logs ou outros Serviços da AWS usando o console, a API ou AWS os SDKs. AWS CLI Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendemos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Criptografia em repouso

CloudWatch O Logs protege os dados em repouso usando criptografia. Todos os grupos de logs são criptografados. Por padrão, o serviço CloudWatch Logs gerencia as chaves de criptografia do lado do servidor.

Se você quiser gerenciar as chaves usadas para criptografar e descriptografar seus registros, use chaves. AWS KMS Para ter mais informações, consulte Criptografe dados de registro no CloudWatch Logs usando AWS Key Management Service.

Criptografia em trânsito

CloudWatch O Logs usa end-to-end criptografia de dados em trânsito. O serviço CloudWatch Logs gerencia as chaves de criptografia do lado do servidor.

Gerenciamento de identidade e acesso para Amazon CloudWatch Logs

O acesso ao Amazon CloudWatch Logs requer credenciais que AWS possam ser usadas para autenticar suas solicitações. Essas credenciais devem ter permissões para acessar AWS recursos, como para recuperar dados de CloudWatch registros sobre seus recursos de nuvem. As seções a seguir fornecem detalhes sobre como você pode usar o AWS Identity and Access Management (IAM) e CloudWatch os registros para ajudar a proteger seus recursos controlando quem pode acessá-los:

- Autenticação
- Controle de acesso

Autenticação

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criando um perfil para um provedor de identidades de terceiros (federação) no Guia do Usuário do IAM.
- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
  - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Controle de acesso

Você pode ter credenciais válidas para autenticar suas solicitações, mas, a menos que tenha permissões, não é possível criar ou acessar 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
- Usando políticas baseadas em identidade (políticas do IAM) para registros CloudWatch
- CloudWatch Referência de permissões de registros

Visão geral do gerenciamento de permissões de acesso aos seus recursos do CloudWatch Logs

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em AWS IAM Identity Center:

  Crie um conjunto de permissões. Siga as instruções em Criação de um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
- Usuários gerenciados no IAM com provedor de identidades:

  Crie um perfil para a federação de identidades. Siga as instruções em Criando um perfil para um provedor de identidades de terceiros (federação) no Guia do Usuário do IAM.
- Usuários do IAM:
  - Crie um perfil que seu usuário possa assumir. Siga as instruções em Criação de um perfil para um usuário do IAM no Guia do usuário do IAM.
  - (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em Adição de permissões a um usuário (console) no Guia do usuário do IAM.

Tópicos

- CloudWatch Registra recursos e operações
- Entendendo a propriedade de recursos
- Gerenciamento de acesso aos recursos
- Especificar elementos da política: ações, efeitos e entidades principais
- Especificar condições em uma política

CloudWatch Registra recursos e operações

Em CloudWatch Logs, os recursos principais são grupos de registros, fluxos de registros e destinos. CloudWatch O Logs não oferece suporte a sub-recursos (outros recursos para uso com o recurso primário).

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 ARN
Grupo de logs	Ambos os itens a seguir são usados. O segundo, com o `:*` no final, é o que é

Tipo de recurso	Formato ARN
	retornado pelo comando `describe-log-groups` CLI e pela DescribeLogGroupsAPI. arn:aws:logs:*region*:*account-id* :log-group:*log_group_name* arn:aws:logs:*region*:*account-id* :log-group:*log_group_name* :* Use a primeira versão, sem o final `:*`, nas seguintes situações: • No campo de `logGroupIdentifier` entrada em muitas CloudWatch Logs APIs. • No `resourceArn` campo em APIs de marcação • Nas IAM políticas, ao especificar permissões para TagResourceUntagResource, e. ListTagsForResource Use a segunda versão, com o final `:*`, para se referir ao ARN ao especificar permissões nas políticas do IAM para todas as outras ações da API.
Stream de log	*arn:aws:logs: region: account-id: log-group: log_group_name:log-stream: log-stream-name*
Destination (Destino)	arn:aws:logs:*region*:*account-id* :destination:*destination_name*

Para obter mais informações sobre ARNs, consulte ARNs no Manual do usuário do IAM. Para obter informações sobre ARNs de CloudWatch registros, consulte Amazon Resource Names (ARNs) em. Referência geral da Amazon Web Services Para ver um exemplo de uma política que abrange

CloudWatch registros, consulteUsando políticas baseadas em identidade (políticas do IAM) para registros CloudWatch .

CloudWatch O Logs fornece um conjunto de operações para trabalhar com os recursos do CloudWatch Logs. Para ver uma lista das operações disponíveis, consulte CloudWatch Referência de permissões de registros.

Entendendo a propriedade de recursos

A AWS conta é proprietária dos recursos criados na conta, independentemente de quem criou os recursos. Especificamente, o proprietário do recurso é a AWS conta da entidade principal (ou seja, a conta raiz, um usuário ou uma função do IAM) que autentica a solicitação de criação do recurso. Os seguintes exemplos mostram como isso funciona:

- Se você usar as credenciais da conta raiz da sua AWS conta para criar um grupo de registros, sua AWS conta é a proprietária do recurso CloudWatch Registros.
- Se você criar um usuário em sua AWS conta e conceder permissões para criar recursos de CloudWatch registros a esse usuário, o usuário poderá criar recursos de CloudWatch registros. No entanto, sua AWS conta, à qual o usuário pertence, é proprietária dos recursos do CloudWatch Logs.
- Se você criar uma função do IAM em sua AWS conta com permissões para criar recursos de CloudWatch registros, qualquer pessoa que possa assumir a função poderá criar recursos de CloudWatch registros. Sua AWS conta, à qual a função pertence, é proprietária dos recursos do CloudWatch Logs.

Gerenciamento de acesso aos recursos

A política de permissões descreve quem tem acesso a quê. A seção a seguir explica as opções disponíveis para a criação das políticas de permissões.

> **Note**
>
> Esta seção discute o uso do IAM no contexto dos CloudWatch registros. Não são fornecidas informações detalhadas sobre o serviço IAM. Para ver a documentação completa do IAM, consulte What is IAM? no IAM User Guide. Para obter informações sobre a sintaxe e as descrições da política do IAM, consulte a IAM policy reference no IAM User Guide.

As políticas anexadas a uma identidade do IAM são chamadas de políticas baseadas em identidade (políticas do IAM) e as políticas anexadas a um recurso são chamadas de políticas baseadas em recursos. CloudWatch O Logs oferece suporte a políticas baseadas em identidade e políticas baseadas em recursos para destinos, que são usadas para permitir assinaturas entre contas. Para ter mais informações, consulte Assinaturas entre contas e regiões.

Tópicos

- Permissões de grupo de logs e Contributor Insights
- Políticas baseadas em recursos

Permissões de grupo de logs e Contributor Insights

O Contributor Insights é um recurso CloudWatch que permite analisar dados de grupos de registros e criar séries temporais que exibem dados do colaborador. 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 você concede a um usuário as `cloudwatch:GetInsightRuleReport` permissões `cloudwatch:PutInsightRule` e, esse usuário pode criar uma regra que avalia qualquer grupo de CloudWatch registros no Logs e depois ver os resultados. Os resultados podem conter dados de colaborador para esses grupos de log. Certifique-se de conceder essas permissões somente aos usuários que possam visualizar esses dados.

Políticas baseadas em recursos

CloudWatch O Logs oferece suporte a políticas baseadas em recursos para destinos, que você pode usar para habilitar assinaturas entre contas. Para ter mais informações, consulte Etapa 1: criar um destino. Os destinos podem ser criados usando a PutDestinationAPI, e você pode adicionar uma política de recursos ao destino usando a PutDestinationPolicyAPI. O exemplo a seguir permite que outra AWS conta com o ID de conta 111122223333 inscreva seus grupos de registros 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: ações, efeitos e entidades principais

Para cada recurso do CloudWatch Logs, o serviço define um conjunto de operações de API. Para conceder permissões para essas operações de API, o CloudWatch Logs define um conjunto de ações que você pode especificar 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 CloudWatch Registra recursos e operações e CloudWatch Referência de permissões de registros.

Estes são os elementos de política básicos:

- Recurso: use um nome de recurso da Amazon (ARN) para identificar o recurso ao qual a política se aplica. Para obter mais informações, consulte CloudWatch Registra recursos e operações.
- Ação: você usa palavras-chave de ação para identificar operações de recursos que 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 (permitir) explicitamente acesso a um recurso, o acesso estará implicitamente negado. Você também pode negar explicitamente o acesso a um recurso, para ter certeza de que um usuário não consiga acessá-lo, mesmo que uma política diferente conceda acesso.
- Entidade principal: em políticas baseadas em identidade (políticas do IAM), o usuário ao qual a política é anexada é a entidade principal implícita. Para políticas baseadas em recursos, você especifica o usuário, a conta, o serviço ou outra entidade que deseja receber permissões (aplica-se somente às políticas baseadas em recursos). CloudWatch O Logs oferece suporte a políticas baseadas em recursos para destinos.

Para saber mais sobre a sintaxe e as descrições da política do IAM, consulte a Referência de política do AWS IAM no Guia do usuário do IAM.

Para ver uma tabela que mostra todas as ações da API CloudWatch Logs e os recursos aos quais elas se aplicam, consulteCloudWatch Referência de permissões de registros.

Especificar condições em uma política

Ao conceder permissões, você pode usar a linguagem da política de acesso para especificar as condições quando uma política deve entrar em vigor. Por exemplo, é recomendável aplicar uma política somente após uma data específica. Para obter mais informações sobre como especificar condições em uma linguagem de política, consulte Condition no Guia do usuário do IAM.

Para expressar condições, você usa chaves de condição predefinidas. Para obter uma lista de chaves de contexto suportadas por cada AWS serviço e uma lista de chaves AWS de política abrangentes, consulte Ações, recursos e chaves de condição para AWS serviços e chaves de contexto de condição AWS globais.

> **Note**
>
> Você pode usar tags para controlar o acesso aos recursos do CloudWatch Logs, incluindo grupos e destinos de registros. O acesso aos fluxos de log é controlado no nível do grupo de log, devido à relação hierárquica entre grupos de log e fluxos de log. Para obter mais informações sobre como usar etiquetas para controlar o acesso, consulte Controlar acesso a recursos da Amazon Web Services usando etiquetas de recursos.

Usando políticas baseadas em identidade (políticas do IAM) para registros CloudWatch

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 que você primeiro analise os tópicos introdutórios que explicam os conceitos básicos e as opções disponíveis para gerenciar o acesso aos seus recursos do CloudWatch Logs. Para ter mais informações, consulte Visão geral do gerenciamento de permissões de acesso aos seus recursos do CloudWatch Logs.

Este tópico abrange o seguinte:

- Permissões necessárias para usar o CloudWatch console
- AWS políticas gerenciadas (predefinidas) para registros CloudWatch
- Exemplos de política gerenciada pelo cliente

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 Manual do usuário do IAM. Para ver uma lista que mostra todas as ações do CloudWatch Logs, consulteCloudWatch Referência de permissões de registros.

Permissões necessárias para usar o CloudWatch console

Para que um usuário trabalhe com CloudWatch Logs no CloudWatch console, ele precisa ter um conjunto mínimo de permissões que permita ao usuário descrever outros AWS recursos em sua AWS conta. Para usar o CloudWatch Logs no CloudWatch console, você precisa ter permissões dos seguintes serviços:

- CloudWatch
- CloudWatch Registros
- OpenSearch Serviço
- IAM
- Kinesis
- Lambda
- Amazon S3

Se você criar uma política do IAM que seja mais restritiva que as permissões mínimas necessárias, o console do não funcionará como pretendido para os usuários com essa política do IAM. Para garantir que esses usuários ainda possam usar o CloudWatch console, anexe também a política `CloudWatchReadOnlyAccess` gerenciada ao usuário, conforme descrito emAWS políticas gerenciadas (predefinidas) para registros CloudWatch .

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API CloudWatch Logs AWS CLI ou para a Logs.

O conjunto completo de permissões necessárias para trabalhar com o CloudWatch console para um usuário que não está usando o console para gerenciar assinaturas de log é:

- cloudwatch: GetMetricData
- cloudwatch: ListMetrics
- troncos: CancelExportTask
- troncos: CreateExportTask
- troncos: CreateLogGroup
- troncos: CreateLogStream
- troncos: DeleteLogGroup

- troncos: DeleteLogStream
- troncos: DeleteMetricFilter
- troncos: DeleteQueryDefinition
- troncos: DeleteRetentionPolicy
- troncos: DeleteSubscriptionFilter
- troncos: DescribeExportTasks
- troncos: DescribeLogGroups
- troncos: DescribeLogStreams
- troncos: DescribeMetricFilters
- troncos: DescribeQueryDefinitions
- troncos: DescribeQueries
- troncos: DescribeSubscriptionFilters
- troncos: FilterLogEvents
- troncos: GetLogEvents
- troncos: GetLogGroupFields
- troncos: GetLogRecord
- troncos: GetQueryResults
- troncos: PutMetricFilter
- troncos: PutQueryDefinition
- troncos: PutRetentionPolicy
- troncos: StartQuery
- troncos: StopQuery
- troncos: PutSubscriptionFilter
- troncos: TestMetricFilter

Para um usuário que também usará o console para gerenciar assinaturas de log, as seguintes permissões também são necessárias:

- Sim: DescribeElasticsearchDomain
- Sim: ListDomainNames

- objetivo: AttachRolePolicy
- objetivo: CreateRole
- objetivo: GetPolicy
- objetivo: GetPolicyVersion
- objetivo: GetRole
- objetivo: ListAttachedRolePolicies
- objetivo: ListRoles
- cinesia: DescribeStreams
- cinesia: ListStreams
- lambda: AddPermission
- lambda: CreateFunction
- lambda: GetFunctionConfiguration
- lambda: ListAliases
- lambda: ListFunctions
- lambda: ListVersionsByFunction
- lambda: RemovePermission
- s3: ListBuckets

AWS políticas gerenciadas (predefinidas) para registros CloudWatch

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 AWS gerenciadas, que você pode associar aos usuários e funções na sua conta, são específicas do CloudWatch Logs:

- CloudWatchLogsFullAccess— Concede acesso total aos CloudWatch registros.
- CloudWatchLogsReadOnlyAccess— Concede acesso somente para CloudWatch leitura aos registros.

CloudWatchLogsFullAccess

A CloudWatchLogsFullAccesspolítica concede acesso total aos CloudWatch registros. A política inclui a `cloudwatch:GenerateQuery` permissão para que os usuários com essa política possam gerar uma string de consulta do CloudWatch Logs Insights a partir de um prompt em linguagem natural. Contém o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*",
                "cloudwatch:GenerateQuery"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

CloudWatchLogsReadOnlyAccess

A CloudWatchLogsReadOnlyAccesspolítica concede acesso somente para CloudWatch leitura aos registros. Ela inclui a `cloudwatch:GenerateQuery` permissão para que os usuários com essa política possam gerar uma string de consulta do CloudWatch Logs Insights a partir de um prompt em linguagem natural. Contém o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
```

```
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudWatchLogsCrossAccountSharingConfiguration

A CloudWatchLogsCrossAccountSharingConfigurationpolítica concede acesso para criar, gerenciar e visualizar links do Observability Access Manager para compartilhar recursos de CloudWatch registros entre contas. Para obter mais informações, consulte CloudWatch observabilidade entre contas.

Contém o seguinte:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:Link",
                "oam:ListLinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
```

```
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        }
    ]
}
```

CloudWatch Registra atualizações em políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do CloudWatch Logs desde que esse serviço começou a monitorar essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página Histórico de documentos de CloudWatch registros.

Alteração	Descrição	Data
CloudWatchLogsFullAccess - Atualizar para uma política existente	CloudWatch Os registros adicionaram uma permissão ao CloudWatchLogsFull Access. A `cloudwatch:GenerateQuery` permissão foi adicionada para que os usuários com essa política possam gerar uma string de consulta do CloudWatch Logs Insights a partir de um prompt em linguagem natural.	27 de novembro de 2023
CloudWatchLogsReadOnlyAccess: atualizar para uma política existente.	CloudWatch adicionou uma permissão para CloudWatchLogsReadOnlyAccess. A `cloudwatch:GenerateQuery` permissão foi	27 de novembro de 2023

Alteração	Descrição	Data
	adicionada para que os usuários com essa política possam gerar uma string de consulta do CloudWatch Logs Insights a partir de um prompt em linguagem natural.	
CloudWatchLogsRead OnlyAccess: atualizar para uma política existente	CloudWatch Os registros adicionaram permissões ao CloudWatchLogsRead OnlyAccess. As `logs:StopLiveTail` permissões `logs:StartLiveTail` e foram adicionadas para que os usuários com essa política possam usar o console para iniciar e interromper CloudWatch as sessões de live tail do Logs. Para obter mais informações, consulte Usar o Live Tail para visualizar registros quase em tempo real.	6 de junho de 2023

Alteração	Descrição	Data
CloudWatchLogsCrossAccountSharingConfiguration – Nova política	CloudWatch O Logs adicionou uma nova política para permitir que você gerencie links de observabilidade CloudWatch entre contas que compartilham grupos de CloudWatch registros do Logs. Para obter mais informações, consulte CloudWatch observabilidade entre contas	27 de novembro de 2022
CloudWatchLogsReadOnlyAccess: atualizar para uma política existente	CloudWatch Os registros adicionaram permissões ao CloudWatchLogsReadOnlyAccess. As `oam:ListAttachedLinks` permissões `oam:ListSinks` e foram adicionadas para que os usuários com essa política possam usar o console para visualizar dados compartilhados das contas de origem na CloudWatch observabilidade entre contas.	27 de novembro de 2022

Exemplos de política gerenciada pelo cliente

Você pode criar suas próprias políticas personalizadas do IAM para permitir permissões para ações e recursos do CloudWatch Logs. Você pode anexar essas políticas personalizadas a usuários ou grupos do que exijam essas permissões.

Nesta seção, você pode encontrar exemplos de políticas de usuário que concedem permissões para várias ações do CloudWatch Logs. Essas políticas funcionam quando você usa a API CloudWatch Logs, AWS os SDKs ou o. AWS CLI

Exemplos

- Exemplo 1: Permitir acesso total aos CloudWatch registros
- Exemplo 2: Permitir acesso somente de leitura aos registros CloudWatch
- Exemplo 3: Permitir acesso a um grupo de logs

Exemplo 1: Permitir acesso total aos CloudWatch registros

A política a seguir 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 somente de leitura aos registros CloudWatch

AWS fornece uma CloudWatchLogsReadOnlyAccesspolítica que permite acesso somente para 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",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "cloudwatch:GenerateQuery"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Exemplo 3: Permitir acesso a um grupo de logs

A política a seguir permite que um usuário leia e grave eventos de log em um grupo de logs especificado.

> **Important**
>
> O `:*` no final do nome do grupo de logs na linha de `Resource` é necessário para indicar que a política se aplica a todos os fluxos de logs nesse grupo de logs. Se você omitir o `:*`, a política não será aplicada.

```
{
   "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 etiquetas 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, etiquete seus grupos de logs e use políticas do IAM que fazem referência a essas etiquetas. Para aplicar tags a um grupo de logs, você precisa ter a permissão `logs:TagResource` ou `logs:TagLogGroup`. Isso se aplica se você estiver atribuindo tags ao grupo de logs ao criá-lo. ou atribuí-los mais tarde.

Para obter mais informações sobre como marcar grupos de logs, consulte Marque grupos de registros no Amazon CloudWatch Logs.

Ao etiquetar 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 etiqueta. 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": {
                    "aws:ResourceTag/Team": "Green"
                }
            }
        }
    ]
}
```

As operações StopQuerye a StopLiveTailAPI não interagem com AWS os recursos no sentido tradicional. Elas não retornam dados, não colocam dados nem modificam recursos. Em vez disso, eles operam somente em uma determinada sessão final ao vivo ou em uma determinada consulta do CloudWatch Logs Insights, que não são categorizadas como recursos. Como resultado, ao especificar o campo `Resource` nas políticas do IAM para essas operações, será necessário definir o valor do campo `Resource` como *, como no exemplo a seguir.

```
{
```

```
    "Version": "2012-10-17",
    "Statement":
        [ {
            "Effect": "Allow",
            "Action": [
                "logs:StopQuery",
                "logs:StopLiveTail"
            ],
            "Resource": "*"
            }
        ]
}
```

Para obter mais informações sobre como usar instruções de política do IAM, consulte Controlar o acesso usando políticas no Manual do usuário do IAM.

CloudWatch Referência de permissões de registros

Ao configurar o Controle de acesso e escrever políticas de permissões que podem ser anexadas a uma identidade do IAM (políticas baseadas em identidade), você pode usar a tabela a seguir como referência. A tabela lista cada operação da API CloudWatch Logs e as ações correspondentes para as quais você pode conceder permissões para realizar a ação. Você especifica as ações no campo `Action` das políticas. Para o `Resource` campo, você pode especificar o ARN de um grupo de registros ou stream de registros, ou especificar `*` para representar todos os recursos de CloudWatch registros.

Você pode usar chaves de condição AWS-wide em suas políticas de CloudWatch registros para expressar condições. Para obter uma lista completa AWS de chaves gerais, consulte Chaves de contexto de condição AWS globais e 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 todas as ações do CloudWatch Logs).

CloudWatch Registra as operações da API e as permissões necessárias para ações

CloudWatch Registra as operações da API	Permissões obrigatórias (ações de 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 Registra as operações da API	Permissões obrigatórias (ações de API)
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Obrigatório para excluir uma definição de consulta salva no CloudWatch Logs Insights.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Obrigatório para excluir uma política de recursos de CloudWatch registros.
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.

CloudWatch Registra as operações da API	Permissões obrigatórias (ações de API)
DescribeMetricFilters	`logs:DescribeMetricFilters` Necessária para visualizar todas as métricas associadas a um grupo de logs.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Obrigatório para ver a lista de definições de consulta salvas no CloudWatch Logs Insights.
DescribeQueries	`logs:DescribeQueries` Obrigatório para ver a lista de consultas do CloudWatch Logs Insights que estão programadas, em execução ou que foram executadas recentemente.
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Obrigatório para ver 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.

CloudWatch Registra as operações da API	Permissões obrigatórias (ações de API)
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` Necessário 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 logs de destino (como um fluxo do Kinesis).
PutDestinationPolicy	`logs:PutDestinationPolicy` Necessária para criar ou atualizar uma política de acesso associada a um destino de log existente.
PutLogEvents	`logs:PutLogEvents` Necessária para carregar um lote de eventos de log para um stream de log.

CloudWatch Registra as operações da API	Permissões obrigatórias (ações de API)
PutMetricFilter	`logs:PutMetricFilter` Necessária para criar ou atualizar um filtro de métrica e associá-lo a um grupo de logs.
PutQueryDefinition	`logs:PutQueryDefinition` Obrigatório para salvar uma consulta no CloudWatch Logs Insights.
PutResourcePolicy	`logs:PutResourcePolicy` Necessário para criar uma política de recursos de CloudWatch registros.
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` Obrigatório para interromper uma consulta do CloudWatch Logs Insights que está em andamento.

CloudWatch Registra as operações da API	Permissões obrigatórias (ações de API)
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.

Usando funções vinculadas a serviços para Logs CloudWatch

O Amazon CloudWatch Logs usa AWS Identity and Access Management funções vinculadas a serviços (IAM). Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente aos CloudWatch Logs. Os papéis vinculados ao serviço são predefinidos pelo CloudWatch Logs e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Uma função vinculada ao serviço torna a configuração de CloudWatch registros mais eficiente porque você não precisa adicionar manualmente as permissões necessárias. CloudWatch O Logs define as permissões de seus papéis vinculados ao serviço e, a menos que seja definido de outra forma, somente o CloudWatch Logs pode assumir esses papéis. As permissões definidas incluem a política de confiança e a política de permissões. Essa política de permissões não pode ser anexada a nenhuma outra entidade do IAM.

Para obter informações sobre outros produtos que oferecem suporte a funções vinculadas a serviços, consulte Serviços da AWS que funcionam com o IAM. Procure os serviços que têm Sim na coluna Função vinculada ao serviço. Escolha um Sim com um link para visualizar a documentação da função vinculada a esse serviço.

Permissões de função vinculadas ao serviço para Logs CloudWatch

CloudWatch O Logs usa a função vinculada ao serviço chamada. AWSServiceRoleForLogDelivery CloudWatch O Logs usa essa função vinculada ao serviço para gravar registros diretamente no Firehose. Para ter mais informações, consulte Habilitar o registro a partir de AWS serviços.

A função vinculada ao serviço AWSServiceRoleForLogDelivery confia nos seguintes serviços para assumir a função:

- `logs.amazonaws.com`

A política de permissões de função permite que o CloudWatch Logs conclua as seguintes ações nos recursos especificados:

- Ação: `firehose:PutRecord` e `firehose:PutRecordBatch` em todos os streams do Firehose que têm uma tag com uma `LogDeliveryEnabled` chave com um valor de. `True` Essa tag é anexada automaticamente a um stream do Firehose quando você cria uma assinatura para entregar os registros ao Firehose.

Você deve configurar permissões para que uma entidade do IAM crie, edite ou exclua uma função vinculada ao serviço. Essa entidade pode ser um usuário, um grupo ou uma função. Para mais informações, consulte Permissões de perfil vinculado ao serviço no Guia do usuário do IAM.

Criação de uma função vinculada ao serviço para o Logs CloudWatch

Não é necessário criar manualmente uma função vinculada ao serviço. Quando você configura registros para serem enviados diretamente para um stream do Firehose na, na ou na AWS API AWS Management Console AWS CLI, o CloudWatch Logs cria a função vinculada ao serviço para você.

Se excluir essa função vinculada ao serviço e precisar criá-la novamente, você poderá usar esse mesmo processo para recriar a função em sua conta. Quando você configura novamente os registros para serem enviados diretamente para um stream do Firehose, o CloudWatch Logs cria novamente a função vinculada ao serviço para você.

Editando uma função vinculada ao serviço para Logs CloudWatch

CloudWatch O Logs não permite que você edite AWSServiceRoleForLogDelivery, nem qualquer outra função vinculada ao serviço, depois de criá-la. Não é possível alterar o nome da função, pois várias entidades podem fazer referência a ela. No entanto, será possível editar a descrição da função usando o IAM. Para obter mais informações, consulte Editar uma função vinculada a serviço no Guia do usuário do IAM.

Excluindo uma função vinculada ao serviço para Logs CloudWatch

Se você não precisar mais usar um atributo ou serviço que requer uma função vinculada a serviço, é recomendável excluí-la. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de sua função vinculada ao serviço antes de excluí-la manualmente.

> **Note**
>
> Se o serviço CloudWatch Logs estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir CloudWatch os recursos do Logs usados pela função vinculada ao AWSServiceRoleForLogDeliveryserviço

- Pare de enviar registros diretamente para os streams do Firehose.

Como excluir manualmente a função vinculada a serviço usando o IAM

Use o console do IAM AWS CLI, o ou a AWS API para excluir a função AWSServiceRoleForLogDeliveryvinculada ao serviço. Para obter mais informações, consulte Excluir uma função vinculada ao serviço.

Regiões suportadas para funções vinculadas ao serviço CloudWatch Logs

CloudWatch O Logs oferece suporte ao uso de funções vinculadas ao serviço em todas as AWS regiões em que o serviço está disponível. Para obter mais informações, consulte CloudWatch Regiões e endpoints de registros.

Validação de conformidade para Amazon CloudWatch Logs

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

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

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar o Amazon CloudWatch Logs é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade da sua empresa e pelas leis e regulamentos aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido de segurança e compatibilidade: estes guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em compatibilidade e segurança na AWS.
- Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos compatíveis com a HIPAA.
- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- Avaliação de recursos com regras no Guia doAWS Config Desenvolvedor — AWS Configavalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes do setor e os regulamentos.
- AWS Security Hub— Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno, AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Resiliência no Amazon 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 aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte Infraestrutura global da AWS.

Segurança da infraestrutura no Amazon CloudWatch Logs

Como um serviço gerenciado, o Amazon CloudWatch Logs é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar CloudWatch os registros pela rede. Os clientes precisam oferecer suporte para:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com sigilo de encaminhamento perfeito (perfect forward secrecy, ou PFS) como DHE (Ephemeral Diffie-Hellman, ou Efêmero Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman, ou Curva elíptica efêmera Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas utilizando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou você pode usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Usando CloudWatch registros com endpoints VPC de interface

Se você usa a Amazon Virtual Private Cloud (Amazon VPC) para hospedar seus AWS recursos, você pode estabelecer uma conexão privada entre sua VPC e Logs. CloudWatch Você pode usar essa conexão para enviar registros para o CloudWatch Logs sem enviá-los pela Internet.

O Amazon VPC é um AWS serviço que você pode usar para lançar AWS recursos em uma rede virtual que você define. 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 uma interface VPC endpoint para Logs. CloudWatch Esse tipo de endpoint permite que você conecte a VPC aos serviços da AWS . O endpoint fornece conectividade confiável e escalável aos CloudWatch registros sem exigir um gateway de internet, instância de tradução de endereços de rede (NAT) ou conexão VPN. Para obter mais informações, consulte O que é a Amazon VPC? no Manual do usuário da Amazon VPC.

Os endpoints VPC da Interface são alimentados por AWS PrivateLink uma AWS tecnologia que permite a comunicação privada entre AWS serviços usando uma interface de rede elástica com endereços IP privados. Para obter mais informações, consulte Novo — AWS PrivateLink para AWS serviços.

As etapas a seguir são para usuários da Amazon VPC. Para obter mais informações, consulte Conceitos básicos no Guia do usuário da Amazon VPC.

Disponibilidade

CloudWatch Atualmente, o Logs oferece suporte a VPC endpoints em todas as AWS regiões, incluindo as regiões. AWS GovCloud (US)

Criação de um VPC endpoint para registros CloudWatch

Para começar a usar o CloudWatch Logs com sua VPC, crie uma interface VPC endpoint para Logs. CloudWatch O serviço a ser escolhido é com.amazonaws.***Região***.logs. Você não precisa alterar nenhuma configuração dos CloudWatch registros. Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

Testando a conexão entre sua VPC e Logs CloudWatch

Depois de criar o endpoint, você pode testar a conexão.

Para testar a conexão entre sua VPC e seu CloudWatch endpoint de registros

1. Conecte-se a uma instância do Amazon EC2 que reside na VPC. Para obter mais informações sobre a conexão, consulte Conecte-se à sua instância do Linux ou Conexão com sua instância Windows na documentação do Amazon EC2.
2. Na instância, use o AWS CLI para criar uma entrada de registro em um dos seus grupos de registros 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.

Controle do acesso ao seu endpoint CloudWatch VPC do Logs

Uma política de endpoint da 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 as políticas do IAM nem as políticas específicas do serviço. É uma política separada para controlar o acesso do endpoint ao serviço especificado.

Políticas de endpoint devem ser gravadas em formato JSON.

Para obter mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Manual do usuário da Amazon VPC.

Veja a seguir um exemplo de uma política de endpoint para CloudWatch Logs. Essa política permite que os usuários se conectem ao CloudWatch Logs por meio da VPC para criar fluxos de registros e enviar CloudWatch registros para o Logs, além de impedir que eles realizem outras CloudWatch ações do Logs.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
```

```
    }
  ]
}
```

Para modificar a política de VPC endpoint para Logs CloudWatch

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc/.
2. No painel de navegação, escolha Endpoints.
3. Se você ainda não criou o endpoint para CloudWatch Logs, escolha Create Endpoint. Em seguida, selecione com.amazonaws.***Região***.logs e selecione Create endpoint (Criar endpoint).
4. Selecione o endpoint com.amazonaws.***Região***.logs e selecione a guia Policy (Política) na metade inferior da tela.
5. Selecione Edit policy (Editar política) e faça as alterações na política.

Compatibilidade com chaves de contexto da VPC

CloudWatch O Logs oferece suporte às chaves de `aws:SourceVpce` contexto `aws:SourceVpc` e de contexto que podem limitar o acesso a VPCs específicas ou a endpoints de VPC específicos. Essas chaves funcionam somente quando o usuário está usando VPC endpoints. Para obter mais informações, consulte Chaves disponíveis para alguns serviços no Manual do usuário do IAM.

Operações da API CloudWatch Logging Logs e do console em AWS CloudTrail

O Amazon CloudWatch Logs é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço no CloudWatch Logs. CloudTrail captura chamadas de API feitas por ou em nome da sua AWS conta. As chamadas capturadas incluem chamadas do CloudWatch console e chamadas de código para as operações da API CloudWatch Logs. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para CloudWatch Logs. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao 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 CloudTrail, inclusive como configurá-lo e ativá-lo, consulte o Guia AWS CloudTrail do usuário.

Tópicos

- CloudWatch Registra as informações em CloudTrail
- Informações de geração de consultas em CloudTrail
- Noções básicas sobre entradas de arquivos de log do

CloudWatch Registra as informações em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando uma atividade de evento compatível ocorre nos CloudWatch registros, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos do CloudWatch Logs, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, a trilha se aplica a todas as AWS regiões. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para

analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte as informações a seguir.

- Visão Geral para Criar uma Trilha
- CloudTrail Serviços e integrações compatíveis
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e recebendo arquivos de CloudTrail log de várias contas

CloudWatch O Logs permite registrar as seguintes ações como eventos em arquivos de CloudTrail log:

- CancelExportTask
- CreateExportTask
- CreateLogGroup
- CreateLogStream
- DeleteDestination
- DeleteLogGroup
- DeleteLogStream
- DeleteMetricFilter
- DeleteRetentionPolicy
- DeleteSubscriptionFilter
- PutDestination
- PutDestinationPolicy
- PutMetricFilter
- PutResourcePolicy
- PutRetentionPolicy
- PutSubscriptionFilter
- StartQuery
- StopQuery
- TestMetricFilter

Somente elementos de solicitação são conectados CloudTrail para essas ações da API CloudWatch Logs:

- DescribeDestinations
- DescribeExportTasks
- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries
- DescribeResourcePolicies
- DescribeSubscriptionFilters
- FilterLogEvents
- GetLogEvents
- GetLogGroupFields
- GetLogRecord
- GetQueryResults

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias para um perfil ou usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte o elemento CloudTrail UserIdentity.

Informações de geração de consultas em CloudTrail

CloudTrail O registro de eventos do console do gerador de consultas também é suportado. Atualmente, o gerador de consultas é compatível com CloudWatch Logs Insights e CloudWatch Metric Insights. Nesses CloudTrail eventos, o `eventSource` é `monitoring.amazonaws.com`.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a GenerateQueryação no CloudWatch Logs Insights.


```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
                "accountId": "123456789012",
                "userName": "SAMPLE_NAME"
            },
            "attributes": {
                "creationDate": "2020-04-08T21:43:24Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "monitoring.amazonaws.com",
    "eventName": "GenerateQuery",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "exampleUserAgent",
    "requestParameters": {
        "query_ask": "***",
        "query_type": "LogsInsights",
        "logs_insights": {
            "fields": "***",
            "log_group_names": ["yourloggroup"]
        },
        "include_description": true
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
```

```
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Noções básicas sobre entradas de arquivos de log do

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

A entrada do arquivo de registro a seguir mostra que um usuário chamou a CreateExportTaskação CloudWatch Registros.

```
{
        "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"
}
```

CloudWatch Referência do agente de registros

> **Important**
>
> Essa referência é para o antigo agente do Logs obsoleto. CloudWatch Se você usa o Instance Metadata Service Version 2 (IMDSv2), deve usar o novo agente unificado. CloudWatch Mesmo que você não esteja usando o IMDSv2, é altamente recomendável usar o CloudWatch agente unificado mais novo em vez do agente de registros mais antigo. Para obter mais informações sobre o novo agente unificado, consulte Coleta de métricas e registros da instância do Amazon EC2 e de servidores locais com o agente. CloudWatch Para obter informações sobre a migração do agente CloudWatch Logs antigo para o agente unificado, consulte Criar o arquivo de configuração do CloudWatch agente com o assistente.

O agente CloudWatch Logs fornece uma forma automatizada de enviar dados de log para CloudWatch Logs a partir de instâncias do Amazon EC2. O agente inclui os seguintes componentes:

- Um plug-in para o AWS CLI que envia dados de registro para CloudWatch o Logs.
- Um script (daemon) que inicia o processo de envio de dados para o Logs. CloudWatch
- 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 as informações necessárias para o 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 do 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), ativa a codificação de conteúdo gzip http para enviar cargas comprimidas para o Logs. CloudWatch Isso diminui o uso da CPU, diminui

NetworkOut e diminui a latência de colocação. Para desativar esse recurso, adicione use_gzip_http_content_encoding = false à seção [general] do arquivo de configuração do agente do Logs e reinicie o CloudWatch 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 de 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 século preenchido como um número decimal preenchido com zeros. 00, 01, ..., 99

%Y: ano com século como número decimal.1970, 1988 2001, 2013

%b: mês como nome abreviado do local. Jan, Fev, ..., Dez (en_US);

%B: mês como nome completo do local. 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 (formato de 24 horas) como um número decimal preenchido com zeros. 00, 01, ..., 23

%I: hora (formato de 12 horas) como um número decimal preenchido com zeros. 01, 02, ..., 12

%p: equivalente de local de AM ou PM.

%M: minuto como um número decimal preenchido com zeros. 00, 01, ..., 59

%S: segundo como um número decimal preenchido com zeros. 00, 01, ..., 59

%f: microssegundo como um número decimal preenchido com zeros à esquerda. 000000, ..., 999999

%z: deslocamento de UTC na forma+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 para o CloudWatch Logs com base no horário 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 para o 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.

Usando o agente CloudWatch Logs com proxies HTTP

Você pode usar o agente CloudWatch Logs com proxies HTTP.

> **Note**
>
> Os proxies HTTP são compatíveis com awslogs-agent-setup .py versão 1.3.8 ou posterior.

Para usar o agente CloudWatch Logs com proxies HTTP

1. Execute um destes procedimentos:

   a. Para uma nova instalação do agente 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 da instância e dados do usuário no Guia do usuário do Amazon EC2.

Nos valores de `http-proxy` e `https-proxy`, você especifica a URL completa.

b. Para uma instalação existente do agente 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
```

Compartimentalizando os arquivos de configuração do agente CloudWatch Logs

Se você estiver usando awslogs-agent-setup o.py versão 1.3.8 ou posterior com awscli-cwlogs 1.3.3 ou posterior, você pode importar diferentes configurações de stream para vários componentes, independentemente um do outro, criando arquivos de configuração adicionais no diretório /var/awslogs/etc/config/. Quando o agente CloudWatch Logs é iniciado, ele inclui todas as configurações de stream 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
```

CloudWatch Perguntas frequentes sobre o agente de registros

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 usou um script de configuração para instalar o agente do CloudWatch Logs, pode usar o /var/awslogs/bin/awslogs-version.sh para verificar qual versão do agente está usando. Ele

imprime a versão do agente e suas principais dependências. Se você usou o yum para instalar o agente CloudWatch Logs, você pode usar “yum info awslogs” e “yum info aws-cli-plugin-cloudwatch -logs” para verificar a versão do agente e do plug-in do Logs. CloudWatch

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/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 CloudWatch Logs grava um aviso em seu registro 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 de CloudWatch registros pode começar de onde parou e continuar enviando os dados de registro.

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 agente CloudWatch Logs exige as `PutLogEvents` operações `CreateLogGroup` `CreateLogStreamDescribeLogStreams`,, e. Se você estiver usando o agente mais recente, o `DescribeLogStreams` não será necessário. Consulte o exemplo de política 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 de CloudWatch registros crie grupos ou fluxos de registros automaticamente. Como posso impedir que o agente recrie grupos e fluxos de log?

Em sua política do IAM, é possível 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 com CloudWatch métricas

CloudWatch O Logs envia métricas para a Amazon a CloudWatch cada minuto.

CloudWatch Métricas de registros

O namespace `AWS/Logs` inclui as métricas a seguir.

Métrica	Descrição
`CallCount`	O número de operações de API especificadas executadas em sua conta. `CallCount` é uma métrica de uso do serviço de CloudWatch registros . Para ter mais informações, consulte CloudWatch Registra métricas de uso do serviço. Dimensões válidas: classe, recurso, serviço, tipo Estatística válida: soma Unidades: nenhuma
`DeliveryErrors`	O número de eventos de registro para CloudWatch os quais o Logs recebeu um erro ao encaminhar dados para o destino da assinatura. Se o serviço de destino retornar um erro que pode ser repetido, como uma exceção de limitação ou uma exceção de serviço que pode ser repetida (HTTP 5xx, por exemplo), o CloudWatch Logs continuará tentando entregar novamente por até 24 horas. CloudWatch Os registros não tentarão ser entregues novamente se o erro for um erro que não possa ser repetido, como ou. `AccessDeniedException` `ResourceNotFoundException` Dimensões válidas: LogGroupName, DestinationType, FilterName, PolicyLevel Estatística válida: soma Unidades: nenhuma

Métrica	Descrição
`DeliveryThrottling`	O número de eventos de registro para CloudWatch os quais o Logs foi limitado ao encaminhar dados para o destino da assinatura. Se o serviço de destino retornar um erro que pode ser repetido, como uma exceção de limitação ou uma exceção de serviço que pode ser repetida (HTTP 5xx, por exemplo), o CloudWatch Logs continuará tentando entregar novamente por até 24 horas. CloudWatch Os registros não tentarão ser entregues novamente se o erro for um erro que não possa ser repetido, como ou. `AccessDeniedException` `ResourceNotFoundException` Dimensões válidas: LogGroupName, DestinationType, FilterName, PolicyLevel Estatística válida: soma Unidades: nenhuma
`EMFParsingErrors`	O número de erros de análise encontrados durante o processamento de logs de formato métrico integrado. Esses erros ocorrem quando os logs são identificados como formato de métrica incorporada, mas não seguem o formato correto. Para obter mais informações sobre o formato de métrica incorporada, consulte Especificação: formato de métrica integrado. Dimensões válidas: `LogGroupName` Estatística válida: soma Unidades: nenhuma

Métrica	Descrição
`EMFValidationErrors`	O número de erros de validação encontrados durante o processamento de logs de formato de métrica integrados. Esses erros ocorrem quando as definições de métrica nos logs de formato de métrica integrados não aderem ao formato de métrica integrado e às especificações `MetricDatum` . Para obter informações sobre o formato métrico CloudWatch incorporado, consulte Especificação: formato métrico incorporado. Para obter informações sobre o tipo de dados`MetricDatum` , consulte MetricDatuma Amazon CloudWatch API Reference. **Note** Certos erros de validação podem fazer com que várias métricas em um log de EMF não sejam publicadas. Por exemplo, todas as métricas definidas com um namespace inválido serão descartadas. Dimensões válidas: `LogGroupName` Estatística válida: soma Unidades: nenhuma
`ErrorCount`	O número de operações de API executadas em sua conta que resultaram em erros. `ErrorCount` é uma métrica de uso do serviço de CloudWatch registros . Para ter mais informações, consulte CloudWatch Registra métricas de uso do serviço. Dimensões válidas: classe, recurso, serviço, tipo Estatística válida: soma Unidades: nenhuma

Métrica	Descrição
`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, PolicyLevel Estatística válida: soma Unidades: nenhuma
`IncomingBytes`	O volume de eventos de log em bytes não compactados enviados para CloudWatch o 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 registro enviados para o 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

Métrica	Descrição
`LogEvents WithFindings`	O número de eventos de registro que correspondem a uma sequência de dados que você está auditando usando o recurso de proteção de dados de CloudWatch registros. Para ter mais informações, consulte Ajude a proteger dados de log confidenciais com mascaramento. Dimensões válidas: nenhuma Estatística válida: soma Unidades: nenhuma
`ThrottleCount`	O número de operações de API executadas em sua conta que foram limitadas por causa de cotas de uso. `ThrottleCount` é uma métrica de uso do serviço de CloudWatch registros. Para ter mais informações, consulte CloudWatch Registra métricas de uso do serviço. Dimensões válidas: classe, recurso, serviço, tipo Estatística válida: soma Unidades: nenhuma

Dimensões para métricas CloudWatch de registros

As dimensões que você pode usar com CloudWatch as métricas do Logs estão listadas na tabela a seguir.

Dimensão	Descrição
`LogGroupName`	O nome do grupo de CloudWatch registros de registros para o qual exibir métricas.

Dimensão	Descrição
`DestinationType`	O destino da assinatura para os dados do CloudWatch Logs, que pode ser AWS Lambda Amazon Kinesis Data Streams ou Amazon Data Firehose.
`FilterName`	O nome do filtro de inscrição que está encaminhando dados do grupo de logs para o destino. O nome do filtro de assinatura é automaticamente convertido CloudWatch em ASCII e qualquer caractere não suportado é substituído por um ponto de interroga ção (?).

As dimensões das métricas relacionadas aos filtros de assinatura no nível da conta estão listadas na tabela a seguir.

Dimensão	Descrição
`PolicyLevel`	O nível em que a política se aplica. Atualmente, o único valor válido para essa dimensão é `AccountPolicy`
`DestinationType`	O destino da assinatura para os dados do CloudWatch Logs, que pode ser AWS Lambda Amazon Kinesis Data Streams ou Amazon Data Firehose.
`FilterName`	O nome do filtro de inscrição que está encaminhando dados do grupo de logs para o destino. O nome do filtro de assinatura é automaticamente convertido CloudWatch em ASCII e qualquer caractere não suportado é substituído por um ponto de interroga ção (?).

CloudWatch Registra métricas de uso do serviço

CloudWatch O Logs envia métricas CloudWatch que rastreiam o uso das operações da API CloudWatch Logs. Essas métricas correspondem às cotas AWS de serviço. O rastreamento dessas métricas pode ajudar a gerenciar as cotas proativamente. Para obter mais informações, consulte Integração do Service Quotas e métricas de uso.

Por exemplo, é possível rastrear a métrica `ThrottleCount` ou definir um alarme nessa métrica. Se o valor dessa métrica aumentar, você deverá considerar solicitar um aumento de cota para a operação de API que está sendo limitada. Para obter mais informações sobre CloudWatch as cotas do serviço de registros, consulte[CloudWatch Registra cotas](#).

CloudWatch O Logs publica métricas de uso da cota de serviço a cada minuto nos namespaces `AWS/Usage` e`AWS/Logs`.

A tabela a seguir lista as métricas de uso do serviço publicadas pelo CloudWatch Logs. Essas métricas não têm uma unidade especificada. A estatística mais útil para essas métricas é `SUM`, que representa a contagem total de operações para o período de um minuto.

Cada uma dessas métricas é publicada com valores para todas as dimensões `Service`, `Class`, `Type` e `Resource`. Também são publicadas com uma única dimensão chamada `Account Metrics`. Use a dimensão `Account Metrics` para ver a soma das métricas de todas as operações de API em sua conta. Use as outras dimensões e especifique o nome de uma operação de API para a dimensão `Resource` localizar as métricas para essa API específica.

Métricas

Métrica	Descrição
`CallCount`	O número de operações especificadas executadas em sua conta. `CallCount` é publicado nos namespaces `AWS/Usage` e `AWS/Logs`.
`ErrorCount`	O número de operações de API executadas em sua conta que resultaram em erros. `ErrorCount` é publicado apenas no `AWS/Logs`.
`ThrottleCount`	O número de operações de API executadas em sua conta que foram limitadas por causa de cotas de uso. `ThrottleCount` é publicado apenas no `AWS/Logs`.

Dimensões

Dimensão	Descrição
`Account metrics`	Use essa dimensão para obter uma soma da métrica em todas as CloudWatch APIs do Logs. Se quiser ver as métricas de uma API específica, use as outras dimensões listadas nesta tabela e especifique o nome da API como o valor de `Resource`.
`Service`	O nome do AWS serviço que contém o recurso. Para métricas de uso do CloudWatch Logs, o valor dessa dimensão é`Logs`.
`Class`	A classe do recurso que está sendo rastreada. CloudWatch As métricas de uso da API Logs usam essa dimensão com um valor de`None`.
`Type`	O tipo de recurso que está sendo acompanhado. No momento, quando a dimensão `Service` é `Logs`, o único valor válido para `Type` é `API`.
`Resource`	O nome da operação da API. Os valores válidos incluem todos os nomes de operação da API listados em Actions (Ações). Por exemplo, `PutLogEvents`

CloudWatch Registra cotas

As tabelas a seguir fornecem as cotas de serviço padrão, também chamadas de limites, para CloudWatch registros de uma AWS conta. A maioria dessas cotas de serviço, mas não todas, estão listadas no namespace Amazon CloudWatch Logs no console Service Quotas. Para solicitar um aumento de cotas para essas cotas, consulte o procedimento mais adiante nesta seção.

Recurso	Cota padrão
Políticas em nível de conta	Uma política de filtro de assinatura em nível de conta por conta. Uma política de proteção de dados em nível de conta por conta. Essas cotas não podem ser alteradas.
Detectores de anomalias	10 detectores de anomalias por conta. Não é possível alterar esta cota.
Tamanho do lote	O tamanho máximo de um batch é 1.048.576 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. Não é possível alterar esta cota.
Arquivamento de dados	Até 5 GB de arquivamento de dados gratuito. Não é possível alterar esta cota.
CreateLogGroup	10 transações por segundo (TPS/conta/região), após as quais as transações são limitadas. É possível solicitar um aumento da cota.
CreateLogStream	50 transações por segundo (TPS/conta/região), após as quais as transações são limitadas. É possível solicitar um aumento da cota.

Recurso	Cota padrão
Identificadores de dados personalizados	Cada política de proteção de dados pode incluir até 10 identificadores de dados personalizados. É possível solicitar um aumento da cota. Cada expressão regular que define um identificador de dados personalizado pode incluir até 200 caracteres. Não é possível alterar esta cota.
DeleteLogGroup	10 transações por segundo (TPS/conta/região), após as quais as transações são limitadas. É possível solicitar um aumento da cota.
DeleteLogStream	15 transações por segundo (TPS/conta/região), após as quais as transações são limitadas. É possível solicitar um aumento da cota.
DescribeLogGroups	10 transações por segundo (TPS/conta/região). É possível solicitar um aumento da cota.
DescribeLogStreams	25 transações por segundo (TPS/conta/região). É possível solicitar um aumento da cota.
Campos de log descobertos	CloudWatch O Logs Insights pode descobrir no máximo 1.000 campos de eventos de registro em um grupo de registros. Não é possível alterar esta cota. Para ter mais informações, consulte Logs compatíveis e campos descobertos.
Campos de log extraídos em logs JSON	CloudWatch O Logs Insights pode extrair no máximo 200 campos de eventos de log de um log JSON. Não é possível alterar esta cota. Para ter mais informações, consulte Logs compatíveis e campos descobertos.

Recurso	Cota padrão
Tarefa de exportação	Uma tarefa de exportação ativa (em execução ou pendente) por vez, por conta. Não é possível alterar esta cota.
FilterLogEvents	25 solicitações por segundo no Leste dos EUA (N. da Virgínia). 5 solicitações por segundo nas seguintes regiões: • Ásia-Pacífico (Jacarta) • Ásia-Pacífico (Osaka) • Europa (Frankfurt) • Oeste do Canadá (Calgary) • Israel (Tel Aviv) 10 solicitações por segundo em outras regiões. Não é possível alterar esta cota.

Recurso	Cota padrão
GetLogEvents	30 solicitações por segundo na Europa (Paris). 10 solicitações por segundo nas seguintes regiões: • Oeste dos EUA (Oregon) • Ásia-Pacífico (Jacarta) • Ásia-Pacífico (Osaka) • Oeste do Canadá (Calgary) • Europa (Irlanda) • Europa (Frankfurt) • Israel (Tel Aviv) 25 solicitações por segundo em todas as outras regiões. Não é possível alterar esta cota. Se você estiver processando novos dados continuam ente, 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.
Sessões simultâneas do Live Tail.	15 de sessões simultâneas. É possível solicitar um aumento da cota.
Live Tail: grupos de logs pesquisad os em uma única sessão.	Máximo de 10 grupos de logs verificados em uma única sessão do Live Tail. Não é possível alterar esta cota.
Tamanho de evento de logs	256 KB (máximo). Não é possível alterar esta cota.

Recurso	Cota padrão
Grupos de logs	Um milhão de grupos de logs 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 logs e 30 dimensões por métrica. Para obter mais informações sobre o formato métrico incorporado, consulte Especificação: Formato métrico incorporado no Guia CloudWatch do usuário da Amazon.
PutLogEvents	O tamanho máximo do lote de uma PutLogEvents solicitação é de 1 MB. Esse tamanho é calculado como a soma de todas as mensagens de evento em UTF-8, mais 26 bytes para cada evento de log. 5000 transações por segundo por conta por região Você pode solicitar um aumento na cota de limitação por segundo usando o serviço. Service Quotas
Tempo limite de execução da consulta	As consultas no CloudWatch Logs Insights expiram após 60 minutos. Esse limite de tempo não pode ser alterado.
Grupos de logs consultados	No máximo 50 grupos de registros podem ser consultados em uma única consulta do CloudWatch Logs Insights. Não é possível alterar esta cota.

Recurso	Cota padrão
Simultaneidade da consulta	Para grupos de registros da classe Standard, no máximo 30 consultas simultâneas do CloudWatch Logs Insights, incluindo consultas que foram adicionadas aos painéis. Para grupos de registros da classe Acesso Infrequente, no máximo 5 consultas simultâneas do CloudWatch Logs Insights, incluindo consultas que foram adicionadas aos painéis. Essas cotas não podem ser alteradas.
Consultas geradas a partir de linguagem natural	Até cinco solicitações de consulta simultâneas geradas em linguagem natural.
Disponibilidade de consultas	As consultas criadas no console ficam disponíveis por 30 dias, por meio do comando Histórico. Esse período de disponibilidade não pode ser alterado. As definições de consulta criadas usando PutQueryDefinitionnão expiram.
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 ter mais informações, consulte CloudWatch Sintaxe de consulta do Logs Insights.

Recurso	Cota padrão
Expressões regulares	Até 5 padrões de filtros contendo expressões regulares para cada grupo de logs ao criar filtros de métricas ou filtros de assinatura. Não é possível alterar esta cota. Até 2 expressões regulares para cada padrão de filtro ao criar um padrão de filtro delimitado ou JSON para filtros de métricas e filtros de assinatura ou ao filtrar eventos de log.
Políticas de recursos	Até 10 políticas de recursos de CloudWatch registros por região por conta. Não é possível alterar esta cota.
Consultas salvas	Você pode salvar até 1000 consultas do CloudWatch Logs Insights, por região e por conta. Não é possível alterar esta cota.
Filtros de assinatura	Dois por grupo de logs. Não é possível alterar esta cota.

Gerenciando suas cotas do serviço CloudWatch Logs

CloudWatch O Logs foi integrado ao Service Quotas, um AWS serviço que permite visualizar e gerenciar suas cotas a partir de um local central. Para obter mais informações, consulte O que são Service Quotas? no Guia do usuário do Service Quotas.

As cotas de serviço facilitam a pesquisa do valor de suas cotas de serviço do CloudWatch Logs.

AWS Management Console

Para ver CloudWatch as cotas de serviço do Logs usando o console

1. Abra o console Service Quotas em https://console.aws.amazon.com/servicequotas/.
2. No painel de navegação, selecione serviços da AWS .
3. Na lista de AWS serviços, pesquise e selecione Amazon CloudWatch Logs.

   Na lista Service quotas é possível ver o nome da service quota, o valor aplicado (se estiver disponível), a cota padrão da AWS e se o valor da cota é ajustável.

4. Para visualizar informações adicionais sobre uma Service Quota, como a descrição, escolha o nome da cota.
5. (Opcional) Para solicitar um aumento de cota, selecione a cota que deseja aumentar, selecione Requesting a quota increase (Solicitar aumento de cota), insira ou selecione as informações necessárias e selecione Request (Solicitar).

Para trabalhar mais com service quotas usando o console, consulte o Guia do usuário de Service Quotas. Para solicitar o aumento da quota, consulte Solicitar um aumento de quota no Guia do usuário do Service Quotas.

AWS CLI

Para visualizar CloudWatch as cotas do serviço Logs usando o AWS CLI

Execute o comando a seguir para ver as cotas de CloudWatch registros padrão.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code logs \
    --output table
```

Para trabalhar mais com cotas de serviço usando o AWS CLI, consulte a Referência de Comandos de AWS CLI Cotas de Serviço. Para solicitar um aumento de quotas, consulte o comando `request-service-quota-increase` na Referência de comandos do AWS CLI.

Histórico do documento

A tabela a seguir descreve mudanças importantes em cada versão do Guia do usuário de CloudWatch registros, a partir de junho de 2018. Para receber notificações sobre atualizações dessa documentação, você poderá se inscrever em um feed RSS.

Alteração	Descrição	Data
CloudWatch O suporte do Logs Insights para geração de consultas em linguagem natural está disponível ao público em geral	CloudWatch O Logs Insights oferece suporte à linguagem natural para gerar e atualizar consultas. Para obter mais informações, consulte Usar linguagem natural para gerar e atualizar consultas do CloudWatch Logs Insights.	20 de junho de 2024
CloudWatchLogsRead OnlyAccesspolítica atualizada	CloudWatch Os registros adicionaram a `cloudwatch:GenerateQuery` permissão para CloudWatchLogsReadOnlyAccessque os usuários com essa política possam gerar uma sequência de caracteres de consulta do CloudWatch Logs Insights a partir de um prompt em linguagem natural.	26 de novembro de 2023
CloudWatchLogsFull Accesspolítica atualizada	CloudWatch Os registros adicionaram a `cloudwatch:GenerateQuery` permissão para CloudWatchLogsFullAccessque os usuários com essa política possam gerar uma sequência de caracteres de consulta do	26 de novembro de 2023

	CloudWatch Logs Insights a partir de um prompt em linguagem natural.	
CloudWatch Logs adiciona análise de padrões de log	CloudWatch Agora, o Logs verifica padrões em eventos de registro toda vez que você realiza uma consulta do CloudWatch Logs Insights. Para obter mais informações, consulte Análise de padrões.	26 de novembro de 2023
CloudWatch Logs adiciona detecção de anomalias de log	Você pode criar um detector de anomalias de log para um grupo de logs. O detector de anomalias examina os eventos de registro ingeridos no grupo de registros e encontra anomalias nos dados de registro. Para obter mais informações, consulte Log anomaly detection.	26 de novembro de 2023
CloudWatch Logs adiciona recurso de comparação	Agora você pode usar o CloudWatch Logs Insights para comparar as alterações em seus eventos de registro ao longo do tempo. Para obter mais informações, consulte Compare (diff) com intervalos de tempo anteriores.	26 de novembro de 2023

CloudWatch Logs adiciona uma nova classe de log	CloudWatch O Logs oferece suporte a duas classes de grupos de registros para que você possa ter uma opção econômica para registros que você acessa com pouca frequência, além de uma opção completa para registros que exigem monitoramento em tempo real ou outros recursos. Para obter mais informações, consulte Log classes.	26 de novembro de 2023
CloudWatch O Logs Insights oferece suporte à geração de consultas em linguagem natural	CloudWatch O Logs Insights oferece suporte à linguagem natural para gerar e atualizar consultas. Para obter mais informações, consulte Usar linguagem natural para gerar e atualizar consultas do CloudWatch Logs Insights.	26 de novembro de 2023
CloudWatch Logs adiciona suporte à sintaxe de padrão de filtro de expressão regular para Live Tail	Agora é possível personalizar ainda mais suas operações de pesquisa e correspon dência para atender às suas necessidades com expressõe s regulares flexíveis com padrões de filtros do Live Tail. Para obter mais informações, consulte Sintaxe do padrão de filtro no Guia do usuário do Amazon CloudWatch Logs.	13 de novembro de 2023

CloudWatch O Logs adiciona suporte à sintaxe de padrão de filtro de expressão regular para filtros métricos, filtros de assinatura e eventos de registro de filtro	Agora é possível personalizar ainda mais suas operações de pesquisa e correspon dência para atender às suas necessidades com expressõe s regulares flexíveis em padrões de filtros. Para obter mais informações, consulte Sintaxe do padrão de filtro no Guia do usuário do Amazon CloudWatch Logs.	5 de setembro de 2023
CloudWatch O Logs Insights adiciona um comando padrão	Agora você pode usar o padrão em suas consultas do CloudWatch Logs Insights para agrupar automaticamente seus dados de registro em padrões. Um padrão é uma estrutura de texto compartil hada que se repete entre seus campos de log. Para obter mais informações, consulte o padrão no Guia do usuário do Amazon CloudWatch Logs.	17 de julho de 2023
CloudWatch O Logs Insights adiciona um comando de desduplicação	Agora você pode usar a desduplicação em suas consultas do CloudWatch Logs Insights para remover resultados duplicados com base em valores específic os nos campos que você especifica. Para obter mais informações, consulte dedup no Guia do usuário do Amazon CloudWatch Logs.	20 de junho de 2023

Políticas de proteção de dados no nível da conta	Agora, é possível definir políticas de proteção de dados no nível da conta. Essas políticas podem auditar e mascarar informações confidenciais em eventos de log em todos os grupos de logs da conta. Para obter mais informações, consulte Ajude a proteger dados de log confidenciais com mascaramento no Guia do usuário do Amazon CloudWatch Logs.	8 de junho de 2023
Adição do recurso Live Tail	CloudWatch Os registros adicionaram o recurso Live Tail, para que você possa escanear os registros à medida que são ingeridos para ajudar na solução de problemas. Há a opção de filtrar o fluxo exibido de eventos de log com base em termos especificados e também de realçar eventos de log que tenham termos especificados. Para obter mais informações, consulte Usar o Live Tail para visualizar registros quase em tempo real.	6 de junho de 2023

CloudWatchLogsRead OnlyAccesspolítica atualizada	CloudWatch Os registros adicionaram permissões ao CloudWatchLogsRead OnlyAccess. As `logs:StopLiveTail` permissões `logs:StartLiveTail` e foram adicionadas para que os usuários com essa política possam usar o console para iniciar e interromper CloudWatch as sessões de live tail do Logs. Para obter mais informações, consulte Usar o Live Tail para visualizar registros quase em tempo real.	6 de junho de 2023
CloudWatch Lançado o Logs Insights	Você pode usar o CloudWatch Logs Insights para pesquisar e analisar interativamente seus dados de registro. Para obter mais informações, consulte Analisar dados de log com o CloudWatch Logs Insights no Guia do usuário do Amazon CloudWatch Logs	27 de novembro de 2018
Suporte para endpoints da Amazon VPC	Agora você pode estabelecer uma conexão privada entre sua VPC e CloudWatch o Logs. Para obter mais informações, consulte Como usar CloudWatch registros com endpoints VPC de interface no Guia do usuário do Amazon CloudWatch Logs.	28 de junho de 2018

A tabela a seguir descreve as mudanças importantes no Guia do usuário do Amazon CloudWatch Logs.

Alteração	Descrição	Data de lançamento
Endpoints da VPC de interface	Em algumas regiões, você pode usar uma interface VPC endpoint para impedir que o tráfego entre sua Amazon VPC e Logs saia CloudWatch da rede Amazon. Para obter mais informações, consulte Usando CloudWatch registros com endpoints VPC de interface.	7 de março de 2018
Logs de consulta de DNS do Route 53	Você pode usar CloudWatch Logs para armazenar registros sobre as consultas de DNS recebidas pelo Route 53. Para mais informações, consulte O que é o Amazon CloudWatch Logs? ou Registrar consultas de DNS no Guia do desenvolvedor do Amazon Route 53.	7 de setembro de 2017
Etiquetar grupos de logs	Você pode usar marcas para categorizar seus grupos de logs. Para ter mais informações, consulte Marque grupos de registros no Amazon CloudWatch Logs.	13 de dezembro de 2016
Melhorias no console	Você pode navegar de gráficos de métricas para os grupos de logs associados. Para ter mais informações, consulte Passar de métricas para logs.	7 de novembro de 2016
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 ter mais informações, consulte Exibir dados de registro enviados para o CloudWatch Logs.	29 de agosto de 2016
AWS CloudTrail Suporte	Foi adicionado AWS CloudTrail suporte para CloudWatch Logs. Para ter mais informações,	10 de março de 2016

Alteração	Descrição	Data de lançamento
adicionado para Amazon CloudWatch Logs e novas métricas de CloudWatch Logs	consulte Operações da API CloudWatch Logging Logs e do console em AWS CloudTrail.	
Suporte adicionado para exportação de CloudWatch registros para o Amazon S3	Foi adicionado suporte para exportação de dados de CloudWatch registros para o Amazon S3. Para ter mais informações, consulte Exportar dados de log para o Amazon S3.	7 de dezembro de 2015
Foi adicionad o suporte para eventos AWS CloudTrail registrados no Amazon CloudWatch Logs	Você pode criar alarmes CloudWatch e receber notificações de atividades específicas da API, conforme capturadas por, CloudTrail e usar a notificação para solucionar problemas.	10 de novembro de 2014
Suporte adicionado para Amazon CloudWatch Logs	Você pode usar o Amazon CloudWatch Logs para monitorar, armazenar e acessar seu sistema, aplicativo e arquivos de log personalizados a partir de instâncias do Amazon Elastic Compute Cloud (Amazon EC2) ou de outras fontes. Em seguida, você pode recuperar os dados de log associado s do CloudWatch Logs usando o CloudWatch console da Amazon, os comandos CloudWatch Logs no ou o AWS CLI SDK do CloudWatch Logs. Para ter mais informações, consulte O que é o Amazon CloudWatch Logs?.	10 de julho de 2014

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o AWS glossário naGlossário da AWS Referência.

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.