aws

Guia do usuário

Agente do Amazon Kinesis do Microsoft Windows

Agente do Amazon Kinesis do Microsoft Windows: Guia do usuário

Table of Contents

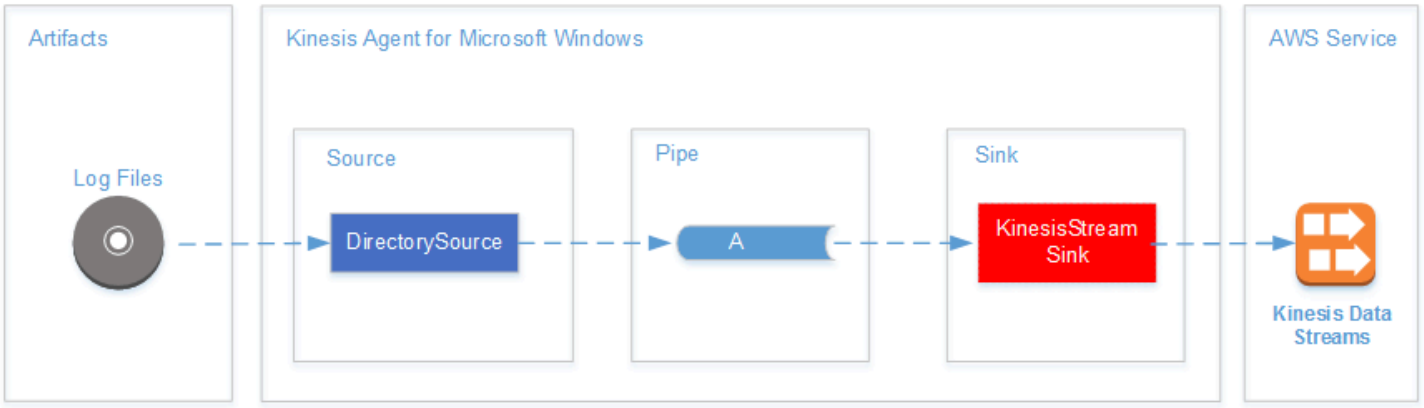
O que é o Amazon Kinesis Agent para o Microsoft Windows?

O Amazon Kinesis Agent for Microsoft Windows (Kinesis Agent for Windows) é um agente configurável e extensível. Ele é executado em computadores desktop e servidores Windows, no local ou na Nuvem AWS. O Kinesis Agent for Windows reúne, analisa, transforma e faz streaming de logs, eventos e métricas de forma eficiente e confiável para vários serviços da AWS, incluindo oKinesis Data Streams,Kinesis Data Firehose,Amazon CloudWatch, eCloudWatch Logs.

Nesses serviços, você pode armazenar, analisar e visualizar os dados usando uma variedade de outros serviços da AWS, incluindo o seguinte:

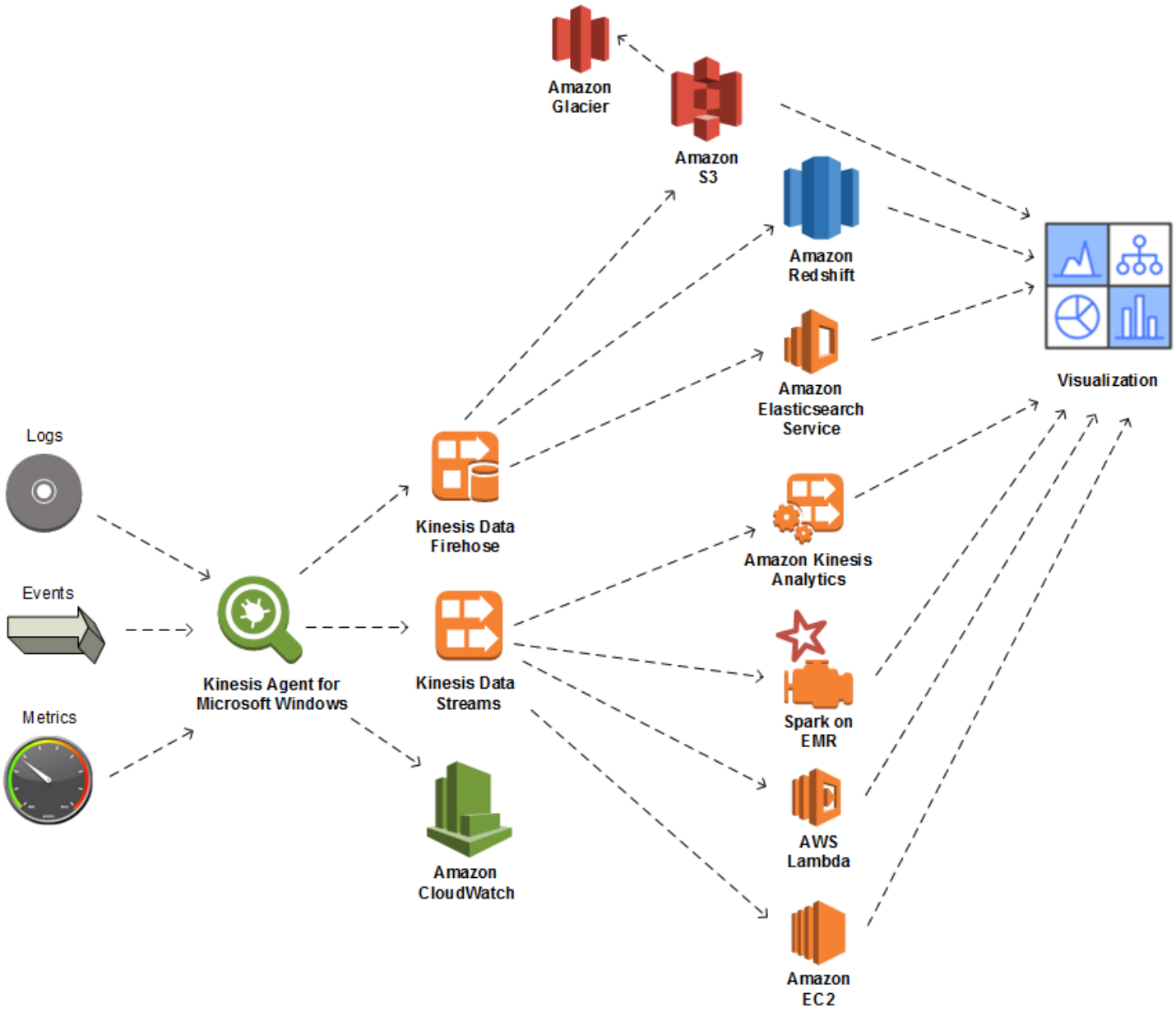
- Amazon Simple Storage Service (Amazon S3)
- Amazon Redshift
- Amazon Elasticsearch Service (Amazon ES)
- Kinesis Data Analytics
- Amazon QuickSight
- Amazon Athena
- Kibana

O diagrama a seguir ilustra uma configuração simples do Kinesis Agent para Windows que faz streaming dos arquivos de log para o Kinesis Data Streams.



Para obter mais informações sobre origens, pipes e coletores, consulte Conceitos do Amazon Kinesis Agent para Microsoft Windows.

O diagrama a seguir ilustra algumas das maneiras de criar pipelines de dados em tempo real personalizados usando estruturas de processamento de streams. Essas estruturas incluem o Kinesis Data Analytics, o Apache Spark no Amazon EMR e o AWS Lambda.



Tópicos

- Sobre a AWS
- O que você pode fazer com o Kinesis Agent for Windows?
- Benefits
- Introdução ao Kinesis Agent para Windows

Sobre a AWS

A Amazon Web Services (AWS) é um conjunto de serviços de infraestrutura digital que você pode utilizar ao desenvolver seus aplicativos. Os serviços incluem computação, armazenamento, banco de dados, análise e sincronização de aplicativos (sistema de mensagens e filas). A AWS usa um modelo de serviço de pagamento por utilização. Você será cobrado apenas pelos serviços que usar — ou por seus aplicativos —. Além disso, para tornar seus serviços mais acessíveis para a criação de protótipos e a experimentação, a AWS oferece um nível de uso gratuito. Neste nível, os serviços são gratuitos abaixo de um determinado nível de uso. Para obter mais informações sobre os custos da AWS e oNível gratuito, consulte oCentro de recursos de conceitos básicos. Para criar uma conta da AWS, abra a página inicial da AWS e cadastre-se.

O que você pode fazer com o Kinesis Agent for Windows?

O Kinesis Agent para Windows oferece os seguintes recursos e funcionalidades:

Coletar logs, eventos e métricas de dados

O Kinesis Agent for Windows coleta, analisa, transforma e faz streaming de logs, eventos e métricas de frotas de servidores e desktops para um ou mais serviços da AWS. A carga recebida pelo serviços pode estar em um formato diferente do original. Por exemplo, um registro pode estar armazenado em um formato de texto específico (como syslog) em um servidor. O Kinesis Agent for Windows pode coletar e analisar esse texto e também transformá-lo no formato JSON, por exemplo, antes do streaming para a AWS. Isso facilita o processamento mais simples feito por alguns serviços da AWS que consomem JSON. Os dados transmitidos por streaming para Kinesis Data Streams podem ser processados de forma contínua pelo Kinesis Data Analytics para gerar métricas adicionais e métricas agregadas que, por sua vez, podem alimentar painéis ao vivo. Você pode armazenar os dados usando uma variedade de serviços da AWS (como o Amazon S3), dependendo de como os dados são usados downstream em um pipeline de dados.

Integrar aos serviços da AWS

Você pode configurar o Kinesis Agent for Windows para enviar arquivos de log, eventos e métricas para vários serviços da AWS:

- Kinesis Data Firehose— Armazene dados transmitidos por streaming no Amazon S3, no Amazon Redshift, no Amazon ES ou noSplunkPara análise mais aprofundada do.
- Kinesis Data Streams— processa dados transmitidos por streaming usando aplicativos personalizados hospedados no Kinesis Data Analytics ou no Apache Spark noAmazon EMR. Ou use código personalizado em execução noAmazon EC2ou funções sem servidor personalizadas em execução noAWS Lambda.
- CloudWatch— Visualize métricas transmitidas por streaming em gráficos, que você pode combinar em painéis. Depois defina alarmes do CloudWatch que são acionados por valores de métrica que violam os limites predefinidos.
- CloudWatch Logs— armazene logs e eventos transmitidos por streaming e visualize e pesquise-os no AWS Management Console ou processe-os downstream em um pipeline de dados.

Instalar e configurar com rapidez

Você pode instalar e configurar o Kinesis Agent para Windows em apenas algumas etapas. Para obter mais informações, consulte Instalando o Kinesis Agent para Windows e Configurando o Amazon Kinesis Agent para Microsoft Windows. Um arquivos de configuração declarativo simples especifica o seguinte:

- As origens e os formatos de logs, eventos e métricas a serem coletados.
- As transformações a serem aplicadas aos dados coletados. É possível incluir outros dados, bem como transformar e filtrar dados existentes.
- Os destinos para os quais os dados finais são enviados por streaming e o armazenamento em buffer, o estilhaçamento e o formato das cargas de streaming.

O Kinesis Agent for Windows vem com analisadores integrados para arquivos de log gerados por serviços empresariais comuns da Microsoft, como:

- Microsoft Exchange
- SharePoint
- Controladores de domínio do Active Directory
- Servidores DHCP

Sem administração contínua

O Kinesis Agent for Windows se adapta automaticamente a várias situações sem perda de dados. Isso inclui a rotação de logs, a recuperação após a reinicialização e interrupções temporárias de serviço e da rede. Você pode configurar o Kinesis Agent para Windows para ser atualizado automaticamente para novas versões. Nenhuma intervenção do operador é necessária em nenhuma dessas situações.

Estender usando arquitetura aberta

Se os recursos declarativos e os plug-ins integrados forem insuficientes para o monitoramento dos sistemas de servidores ou desktops, você poderá estender o Kinesis Agent for Windows criando plug-ins. Novos plug-ins ativam novas origens e destinos para logs, eventos e métricas. O código-fonte do Kinesis Agent para Windows está disponível emhttps://github.com/awslabs/kinesis-agent-windows.

Benefits

O Kinesis Agent para Windows executa a coleta de dados inicial, a transformação e o streaming de logs, eventos e métricas para pipelines de dados. A compilação desses pipelines de dados tem vários benefícios:

Análise e visualização

A integração do Kinesis Agent for Windows com o Kinesis Data Firehose e seus recursos de transformação facilitam a integração com vários serviços analíticos e de visualização diferentes:

- Amazon QuickSight— Um serviço de BI baseado na nuvem que pode consumir de diferentes origens. O Kinesis Agent para Windows pode transformar dados e transmiti-los para o Amazon S3 e o Amazon Redshift por meio do Kinesis Data Firehose. Esse processo permite a descoberta de informações detalhadas dos dados usando visualizações do Amazon QuickSight.
- Athena— Um serviço de consulta interativo que permite a consulta de dados baseada em SQL. O Kinesis Agent para Windows pode transformar e transmitir dados para o Amazon S3 por meio do Kinesis Data Firehose. O Athena pode executar, de forma interativa, consultas SQL nesses dados para inspecionar e analisar logs e eventos com rapidez.
- Kibana— Uma ferramenta de visualização de dados de código aberto. O Kinesis Agent para Windows pode transformar e transmitir dados para o Amazon ES por meio do Kinesis Data Firehose. Depois, você pode usar o Kibana para explorar esses dados. Crie e abra visualizações diferentes, incluindo histogramas, gráficos de linha, gráficos de pizza, mapas de calor e gráficos geoespaciais.

Security

Um pipeline de análise de dados de eventos e logs que inclui o Kinesis Agent for Windows pode detectar e alertar sobre violações de segurança em organizações, o que pode ajudá-lo a bloquear ou interromper ataques.

Desempenho do aplicativo

O Kinesis Agent for Windows pode coletar logs, eventos e dados de métricas sobre o desempenho do aplicativo ou do serviço. Um pipeline de dados completo pode, então, analisar esses dados. Essa análise ajuda a melhorar o desempenho e a confiabilidade de seu aplicativo e serviço detectando

e relatando defeitos que, de outra forma, poderiam não estar aparentes. Por exemplo, você pode detectar alterações significativas no tempo de execução de chamadas de API de serviço. Quando correlacionado a uma implantação, esse recurso ajuda você a localizar e resolver novos problemas de desempenho com serviços que você tem.

Operações de serviço

Um pipeline de dados pode analisar os dados coletados para prever possíveis problemas operacionais e fornecer informações sobre como evitar interrupções de serviço. Por exemplo, você pode analisar logs, eventos e métricas para determinar o uso atual e projetado da capacidade para que você possa criar capacidade online adicional antes que os usuários do serviço sejam afetados. Se ocorrer uma interrupção de serviço, você poderá analisar os dados para determinar o impacto nos clientes durante o período de interrupção.

Auditing

Um pipeline de dados pode processar os logs, eventos e métricas que o Kinesis Agent for Windows coleta e transforma. Depois, você pode auditar esses dados processados usando vários serviços da AWS. Por exemplo, o Kinesis Data Firehose pode receber um stream de dados do Kinesis Agent para Windows, que armazena os dados no Amazon S3. Depois, você pode auditar esses dados executando consultas SQL interativas com o Athena.

Archiving

Muitas vezes, os dados operacionais mais importantes são os dados coletados recentemente. No entanto, a análise de dados coletados sobre aplicativos e serviços ao longo de vários anos também

pode ser útil, por exemplo, para planejamento de longo alcance. Manter grandes quantidades de dados pode ser caro. O Kinesis Agent para Windows pode coletar, transformar e armazenar dados no Amazon S3 por meio do Kinesis Data Firehose. Portanto,Amazon S3 GlacierO está disponível para reduzir os custos de arquivamento de dados mais antigos.

Alerting

O Kinesis Agent para Windows transmite métricas para o CloudWatch. Por sua vez, você pode criar alarmes do CloudWatch para enviar uma notificação por meio doAmazon Simple Notification Service (Amazon SNS)quando uma métrica viola consistentemente um limite específico. Isso proporciona aos engenheiros maior consciência dos problemas operacionais com seus aplicativos e serviços.

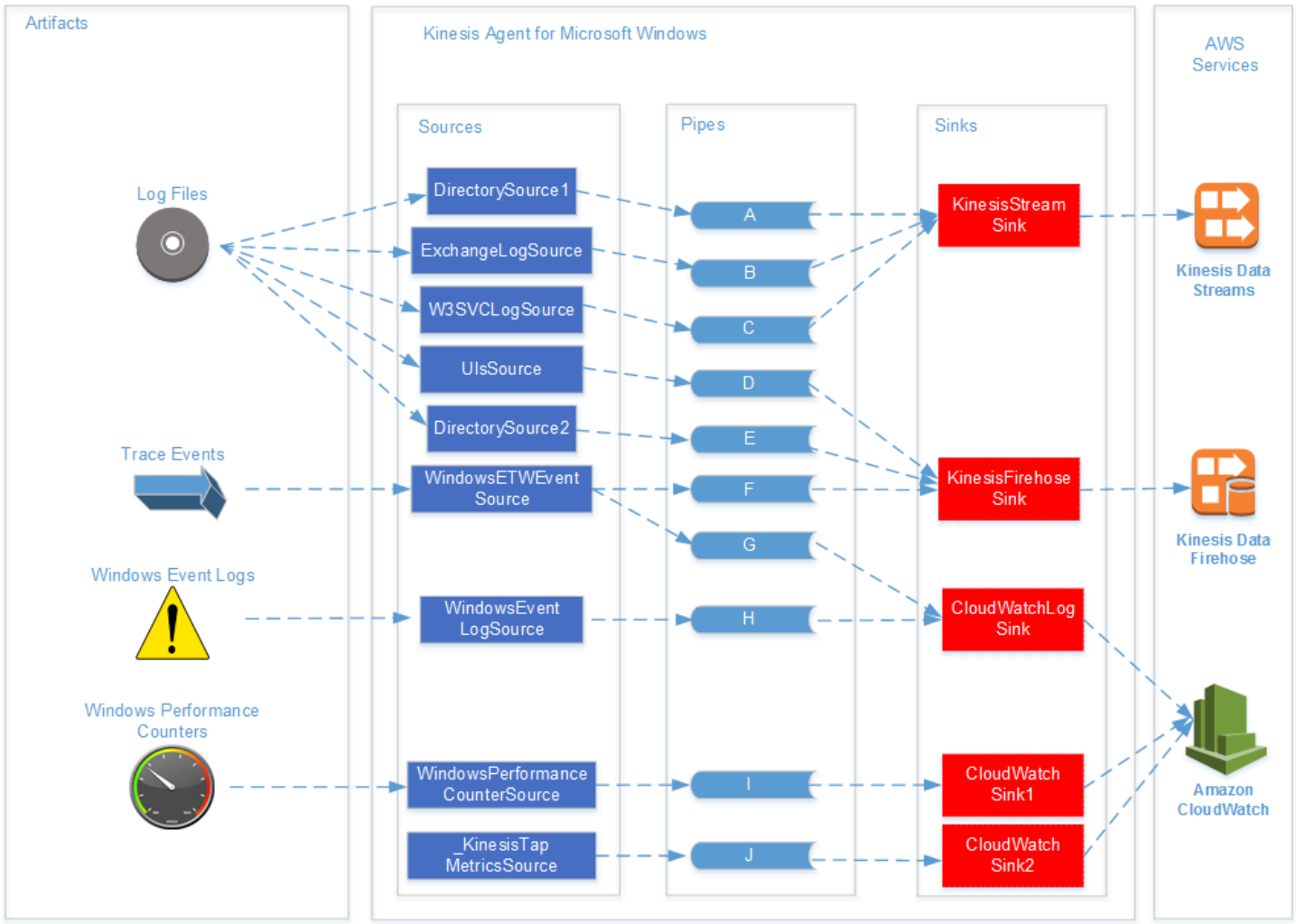
Introdução ao Kinesis Agent para Windows

Para saber mais sobre o Kinesis Agent para Windows, recomendamos começar com as seguintes seções:

- Conceitos do Amazon Kinesis Agent para Microsoft Windows
- Conceitos básicos do Amazon Kinesis Agent para Microsoft Windows

Conceitos do Amazon Kinesis Agent para Microsoft Windows

Compreender os principais conceitos do Amazon Kinesis Agent para Microsoft Windows (Agente do Kinesis para Windows) pode facilitar a coleta e o streaming de dados em frotas de desktops e servidores para o restante do pipeline de dados para processamento.



Este diagrama de um pipeline de dados ilustra os seguintes componentes e processos:

Servidores e áreas de trabalho têm artefatos como arquivos de log, eventos e métricas coletados por um ou mais agentes do Kinesis para Windowssources. Os dados podem ser transformados, por exemplo, de um formato de texto de arquivo simples em um objeto.

Os dados (em formato de objeto ou texto) podem então fluir para um ou mais agentes do Kinesis para WindowsPipes. Um pipe conecta uma origem a um agente Kinesis para Windowssink. O pipe também pode filtrar dados desnecessários.

Um coletor também pode transformar dados analisados em objetos em JSON ou XML. O coletor envia os dados para um serviço específico da AWS, como Kinesis Data Streams, Kinesis Data Firehose ou Amazon CloudWatch.

Usando vários pipes, uma única origem pode enviar os mesmos dados para vários coletores (por exemplo, consulte os pipes F e G no diagrama). Usando vários pipes, origens diferentes podem fazer streaming de dados para um único coletor (por exemplo, consulte os pipes A, B e C no diagrama). Também é possível usar vários pipes para fazer streaming de dados de vários coletores para várias origens. Origens, coletores e pipes têm tipos e pode haver mais de uma origem, coletor ou pipe do mesmo tipo.

Para obter exemplos de arquivos de configuração que declaram origens, coletores e pipes, consulte Exemplos de configuração do Kinesis Agent para Windows.

Tópicos

Pipelines de dados

AData Pipelineé usado para reunir, processar, visualizar e possivelmente gerar alarmes para aplicativos e serviços. O Kinesis Agent para Windows se encaixa em pipelines de dados no início, onde logs, eventos e métricas são coletados de frotas de computadores desktop ou servidores. O Kinesis Agent para Windows transmite os dados coletados para os vários serviços da AWS que formam o restante do pipeline de dados. Um pipeline de dados tem uma finalidade, como visualizar a integridade de um serviço em tempo real para ajudar os engenheiros a operar esse serviço com mais eficiência. Um pipeline de dados de integridade do serviço pode realizar qualquer uma das seguintes ações:

- Alertar os engenheiros para problemas antes que eles afetem a experiência dos clientes dos serviços.

- Ajudar os engenheiros a gerenciar com eficiência o custo do serviço mostrando as tendências de uso de recursos. Essas tendências permitem que eles ajustem os níveis de recursos adequadamente ou até mesmo implementem cenários de escalabilidade automática.
- Fornecer informações sobre a causa raiz de problemas que são relatados por clientes do serviço. Isso acelera a resolução desses problemas e reduz os custos de suporte.

Para obter um exemplo passo a passo de como criar um pipeline de dados usando o Kinesis Agent para Windows, consulteTutorial: Transmitir arquivos de log JSON para o Amazon S3 usando o Kinesis Agent para Windows.

Sources

Um agente Kinesis para Windowssourcereúne logs, eventos ou métricas. Uma origem reúne um tipo de dados de um produtor desses dados com base no tipo de origem. Por exemplo, o tipo `DirectorySource` reúne os arquivos de log de diretórios específicos no sistema de arquivos. Se os dados ainda não estiverem estruturados (como com alguns tipos de arquivos de log), uma origem poderá ser útil ao analisar a representação textual em um formato estruturado. Cada origem corresponde a umDeclaração de origemno Kinesis Agent para Windows`appsettings.json`Arquivo de configuração. A declaração de origem fornece detalhes essenciais para configurar a origem para ajustá-la com base nos requisitos de coleta de dados específicos. Os tipos de detalhes que podem ser configurados variam de acordo com o tipo de origem. Por exemplo, o tipo de origem `DirectorySource` requer a especificação do diretório onde os arquivos de log estão localizados.

Para obter mais detalhes sobre tipos e declarações de origem, consulte Declarações de origem.

Sinks

Um agente Kinesis para Windowssinkleva os dados coletados por uma fonte do Kinesis Agent para Windows e transmite esses dados para um dos vários serviços da AWS possíveis que formam o restante do pipeline de dados. Cada coletor corresponde a umDeclaração de coletorno Kinesis Agent para Windows`appsettings.json`Arquivo de configuração. A declaração de coletor fornece detalhes essenciais para configurar o coletor para ajustá-lo com base nos requisitos de streaming de dados específicos. Os tipos de detalhes que podem ser configurados variam de acordo com o tipo de coletor. Por exemplo, alguns tipos de coletores permitem que uma declaração de coletor especifique um `Format` de serialização para os dados fornecidos a eles. Quando essa opção é especificada na declaração de coletor, a serialização dos dados coletados ocorre antes do streaming dos dados para o serviço da AWS que está associado ao coletor.

Para obter mais informações sobre tipos e declarações de coletor, consulte Declarações de coletor.

Pipes

Um agente Kinesis para WindowsPipesconecta a saída de uma origem do Kinesis Agent para Windows à entrada de um coletor Kinesis Agent para Windows. Ele também pode transformar os dados à medida que eles percorrem o pipe. Cada pipe corresponde a uma declaração de pipe no Kinesis Agent para Windows`appsettings.json`Arquivo de configuração. A declaração de pipe fornece detalhes essenciais para configurar o coletor, como a origem e o coletor para o pipe.

Para obter mais informações sobre tipos e declarações de pipe, consulte Declarações de pipe.

Conceitos básicos do Amazon Kinesis Agent para Microsoft Windows

Você pode usar o Amazon Kinesis Agent para Microsoft Windows (Kinesis Agent para Windows) para coletar, analisar, transformar e fazer streaming de logs, eventos e métricas da frota do Windows para vários serviços da AWS. As informações a seguir contêm pré-requisitos e instruções passo a passo para a instalação e a configuração do Kinesis Agent para Windows.

Tópicos

Prerequisites

Antes de instalar o Kinesis Agent para Windows, verifique se você tem os seguintes pré-requisitos:

- Familiaridade com os conceitos do Kinesis Agent para Windows. Para obter mais informações, consulte Conceitos do Amazon Kinesis Agent para Microsoft Windows.
- Uma conta da AWS para usar os vários serviços da AWS relacionados ao seu pipeline de dados. Para obter informações sobre a criação e a configuração de uma conta da AWS, consulte Configuração de uma conta da AWS.
- Microsoft .NET Framework 4.6 ou posterior em cada desktop ou servidor que executará o Kinesis Agent para Windows. Para obter mais informações, consulte Instalar o .NET Framework para desenvolvedores na documentação do Microsoft .NET.

  Para determinar a versão mais recente do .NET Framework que está instalada em um desktop ou servidor, use o seguinte script do PowerShell:

```
[System.Version](
(Get-ChildItem 'HKLM:\SOFTWARE\Microsoft\NET Framework Setup\NDP' -recurse `
| Get-ItemProperty -Name Version -ErrorAction SilentlyContinue `
| Where-Object { ($_.PSChildName -match 'Full') } `
| Select-Object Version | Sort-Object -Property Version -Descending)[0]).Version
```

- Os streams para onde você deseja enviar dados do Kinesis Agent para Windows (se você estiver usando o Amazon Kinesis Data Streams). Crie os fluxos usando oConsole do Kinesis Data Streams, oCLI DA AWS, ouAWS Tools para Windows PowerShell. Para obter mais informações, consulteCriar e atualizar streamings de dadosnoGuia do desenvolvedor do Amazon Kinesis Data Streams.
- Os fluxos de entrega do Firehose para onde você deseja enviar dados do Kinesis Agent para Windows (se você estiver usando o Amazon Kinesis Data Firehose). Crie fluxos de entrega usando oConsole do Kinesis Data Firehose, oCLI DA AWS, ouAWS Tools para Windows PowerShell. Para obter mais informações, consulte Como criar um fluxo de entrega do Amazon Kinesis Firehose Data no Guia do desenvolvedor do Amazon Kinesis Data Firehose.

Configuração de uma conta da AWS

Se você ainda não tiver uma conta da AWS, siga estas etapas para criar uma.

Para se cadastrar em uma conta da AWS

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

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

Para criar um usuário administrador para você mesmo e adicionar o usuário a um grupo de administradores (console)

1. Faça login no console do IAM como o proprietário da conta escolhendo Root user (Usuário raiz) e inserindo seu endereço de e-mail da conta da AWS. Na próxima página, insira sua senha.

   > **Note**
   >
   > Recomendamos que você siga as melhores práticas para utilizar o**`Administrator`**Usuário do IAM que segue e armazene as credenciais do usuário raiz com segurança. Cadastre-se como o usuário raiz apenas para executar algumas tarefas de gerenciamento de serviços e contas.

2. No painel de navegação, escolha Usuários e depois Adicionar usuário.
3. Em User name (Nome do usuário), digite **`Administrator`**.
4. Marque a caixa de seleção ao lado de AWS Management Console access (Acesso ao Console de Gerenciamento da AWS). Então, selecione Custom password (Senha personalizada), e insira sua nova senha na caixa de texto.
5. (Opcional) Por padrão, a AWS exige que o novo usuário crie uma senha ao fazer login pela primeira vez. Você pode desmarcar a caixa de seleção próxima de User must create a new password at next sign-in (O usuário deve criar uma senha no próximo login) para permitir que o novo usuário redefina a senha depois de fazer login.
6. SelecionePróximo: Permissões
7. Em Set permissions (Conceder permissões), escolha Add user to group (Adicionar usuário ao grupo).
8. Escolha Create group (Criar grupo).
9. Na caixa de diálogo Create group (Criar grupo), em Group name (Nome do grupo), digite **`Administrators`**.
10. SelecionePolíticas de filtroe, depois, selecioneAWS gerenciado — função de trabalhopara filtrar o conteúdo da tabela.
11. Na lista de políticas, marque a caixa de seleção AdministratorAccess. A seguir escolha Criar grupo.

> **Note**
>
> Ative acesso do usuário e da função do IAM ao Faturamento para poder usar as permissões de `AdministratorAccess` para acessar o console de Gerenciamento de custos e faturamento da AWS. Para fazer isso, siga as instruções na etapa 1 do tutorial sobre como delegar acesso ao console de faturamento.

12. Suporte a lista de grupos, selecione a caixa de seleção para seu novo grupo. Escolha Refresh (Atualizar) caso necessário, para ver o grupo na lista.
13. SelecionePróximo: Tags.
14. (Opcional) Adicione metadados ao usuário anexando tags como pares de chave-valor. Para obter mais informações sobre como usar tags no IAM, consulte Marcar entidades do IAM no Guia do usuário do IAM.

15. SelecionePróximo: Review (Revisar)Para ver uma lista de associações a grupos a serem adicionadas ao novo usuário. Quando você estiver pronto para continuar, selecione Criar usuário.

É possível usar esse mesmo processo para criar mais grupos e usuários e conceder aos usuários acesso aos recursos da conta da AWS. Para saber como usar políticas para restringir as permissões de usuário a recursos específicos da AWS, consulte Gerenciamento de acesso e Políticas de exemplo.

Para se cadastrar na AWS e criar uma conta de administrador

1. Se você não tiver uma conta da AWS, abrahttps://aws.amazon.com/. Escolha Create an AWS Account (Criar uma conta da AWS) e siga as instruções online.

   Parte do procedimento de cadastro envolve uma chamada telefônica e a digitação de um PIN usando o teclado do telefone.
2. Faça login no Console de Gerenciamento da AWS e abra o console do IAM em https://console.aws.amazon.com/iam/.
3. No painel de navegação, escolha Groups (Grupos) e Create New Group (Criar novo grupo).
4. Em Group Name (Nome do grupo), digite um nome para o grupo, como **`Administrators`** e escolha Next Step (Próxima etapa).
5. Na lista de políticas, marque a caixa de seleção ao lado da política AdministratorAccess. Você pode usar o menu Filtro e a caixa Pesquisar para filtrar a lista de políticas.
6. Escolha Next Step. Escolha Create Group (Criar grupo), e seu novo grupo será exibido em Group Name (Nome do grupo).
7. No painel de navegação, escolha Users e depois Create New Users.
8. Na caixa 1, digite um nome de usuário, desmarque a caixa de seleção ao lado de Generate an access key for each user (Gerar uma chave de acesso para cada usuário) e escolha Create (Criar).
9. Na lista de usuários, escolha o nome (não a caixa de seleção) do usuário que você acabou de criar. Você pode usar a caixa Search (Pesquisar) para pesquisar o nome do usuário.
10. Selecione a guia Groups (Grupos) e Add User to Groups (Adicionar usuários a grupos).
11. Marque a caixa de seleção ao lado do grupo de administradores e escolha Add to Groups (Adicionar a grupos).

12. Selecione a guia Security Credentials (Credenciais de segurança). Em Credenciais de login, escolha Gerenciar senha.
13. Selecione Assign a custom password (Atribuir uma senha personalizada), insira uma senha nas caixas Password (Senha) e Confirm Password (Confirmar senha) e Apply (Aplicar).

Instalando o Kinesis Agent para Windows

Há três maneiras de instalar o Kinesis Agent para Windows no Windows:

- Instale usando MSI (um pacote de instalação do Windows).
- Instalar emAWS Systems Manager, um conjunto de serviços para administrar servidores e desktops.
- Execute um script do PowerShell.

> **Note**
>
> As instruções a seguir ocasionalmente usam os termos KinesisTap e AWSKinesisTap. Essas palavras são o mesmo que o Kinesis Agent para Windows, mas você deve especificá-las no estado em que se encontram ao seguir estas instruções.

Instalar o Kinesis Agent para Windows usando MSI

É possível fazer download do pacote MSI do Kinesis Agent para Windows norepositório kinesis-agent-windows no GitHub. Depois de baixar o MSI, use o Windows para iniciá-lo e siga as instruções do instalador. Após a instalação, você pode desinstalar como faria com qualquer aplicativo do Windows.

Se preferir, você também pode usar amsiexecNo prompt de comando do Windows para instalar silenciosamente, ativar o registro em log e desinstalar conforme mostrado nos exemplos a seguir. Substituir*AWSKinesisTap.1.1.216.4.msi* `with the appropriate version of Kinesis Agent for Windows for your application.`

Para instalar silenciosamente o Kinesis Agent para Windows:

```
msiexec /i AWSKinesisTap.1.1.216.4.msi /q
```

Para registrar mensagens de instalação para solução de problemas em um arquivo chamado*logfile.log*:

```
msiexec /i AWSKinesisTap.1.1.216.4.msi /q /L*V logfile.log
```

Para desinstalar o Kinesis Agent para Windows usando o prompt de comando:

```
msiexec.exe /x {ADAB3982-68AA-4B45-AE09-7B9C03F3EBD3} /q
```

Instalar o Kinesis Agent para Windows usando o AWS Systems Manager

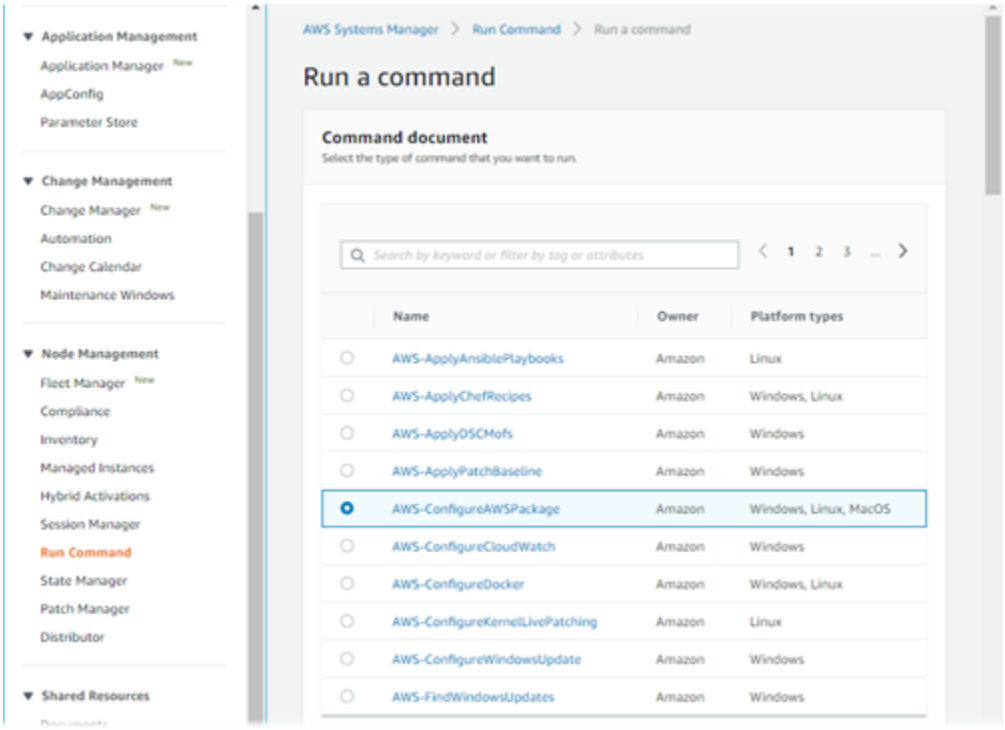
Siga estas etapas para instalar o Kinesis Agent para Windows usando o Systems Manager Run Command. Para obter mais informações sobre o Run Command, consulteAWS Systems ManagernoGuia do usuário do AWS Systems Manager. Além de usar o Systems Manager Run Command, você também pode usar o Systems ManagerJanelas de manutençãoeState Managerpara automatizar a implantação do Kinesis Agent para Windows ao longo do tempo.

> **Note**
>
> A instalação do Systems Manager para o Kinesis Agent para Windows está disponível nas regiões da AWS listadas emAWS Systems ManagerExceto o seguinte:
>
> - cn-north-1
> - cn-northwest-1
> - Todas as regiões do AWS GovCloud.

Para instalar o Kinesis Agent para Windows usando o Systems Manager

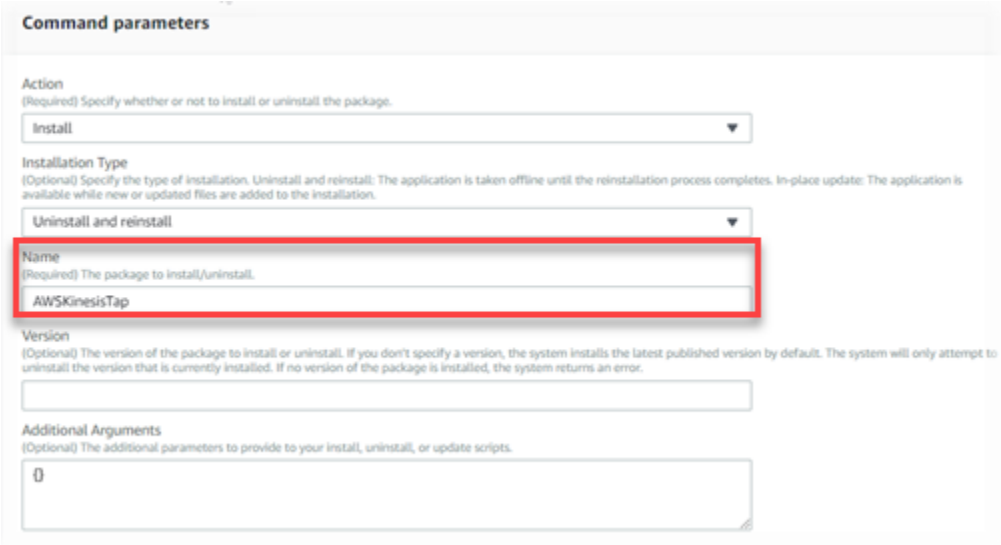
1. Verifique se a versão 2.2.58.0 ou posterior do SSM Agent está instalada em instâncias nas quais você deseja instalar o Kinesis Agent para Windows. Para obter mais informações, consulteInstalar e configurar o SSM Agent em instâncias do WindowsnoGuia do usuário do AWS Systems Manager.
2. Abra o console do AWS Systems Manager emhttps://console.aws.amazon.com/systems-manager/.
3. No painel de navegação, emGerenciamento de nós, escolhaComando de execução doe, depois, escolhaComando de execução do.

4. DoDocumento de comando, selecione a caixa de seleçãoConfigureAWSPackagedocument.



5. UnderParâmetros de comando, paraName (Nome), insiraAWSkineSistap. Deixe outras configurações com seus valores padrão.

> **Note**
>
> SAIRVersãoEm branco para especificar a versão mais recente do pacote AWSkineSystap. Se preferir, você poderá inserir uma versão específica para instalar.



6. UnderDestinos, especifique as instâncias nas quais executar o comando. Você pode optar por especificar instâncias com base em tags associadas a instâncias, escolher instâncias manualmente ou especificar um grupo de recursos que inclua instâncias.
7. Deixe todas as outras configurações com seus valores padrão e escolhaExecução do.

Instalar o Kinesis Agent para Windows usando o PowerShell

Use um editor de texto para copiar os seguintes comandos em um arquivo e salve-o como um script do PowerShell. Usamos`InstallKinesisAgent.ps1`No exemplo a seguir.

```
Param(
    [ValidateSet("prod", "beta", "test")]
    [string] $environment = 'prod',
    [string] $version,
    [string] $baseurl
)

# Self-elevate the script if required.
if (-Not ([Security.Principal.WindowsPrincipal]
 [Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole([Security.Principal.WindowsBuiltI
 'Administrator')) {
    if ([int](Get-CimInstance -Class Win32_OperatingSystem | Select-Object -
ExpandProperty BuildNumber) -ge 6000) {
        $CommandLine = '-File "' + $MyInvocation.MyCommand.Path + '" ' +
 $MyInvocation.UnboundArguments
```

```
        Start-Process -FilePath PowerShell.exe -Verb Runas -ArgumentList $CommandLine
        Exit
    }
}

# Allows input to change base url. Useful for testing.
if ($baseurl) {
    if (!$baseUrl.EndsWith("/")) {
        throw "Invalid baseurl param value. Must end with a trailing forward slash
 ('/')"
    }

    $kinesistapBaseUrl = $baseurl
} else {
    $kinesistapBaseUrl = "https://s3-us-west-2.amazonaws.com/kinesis-agent-windows/
downloads/"
}

Write-Host "Using $kinesistapBaseUrl as base url"

$webClient = New-Object System.Net.WebClient

try {
    $packageJson = $webClient.DownloadString($kinesistapBaseUrl + 'packages.json' + '?
_t=' + [System.DateTime]::Now.Ticks) | ConvertFrom-Json
} catch {
    throw "Downloading package list failed."
}



if ($version) {
    $kinesistapPackage = $packageJson.packages | Where-Object { $_.packageName -eq
 "AWSKinesisTap.$version.nupkg" }

    if ($null -eq $kinesistapPackage) {
        throw "No package found matching input version $version"
    }
} else {
    $packageJson = $packageJson.packages | Where-Object { $_.packageName -match
 ".nupkg" }
    $kinesistapPackage = $packageJson[0]
}

$packageName = $kinesistapPackage.packageName
```

```
$checksum = $kinesistapPackage.checksum

#Create %TEMP%/kinesistap if not exists
$kinesistapTempDir = Join-Path $env:TEMP 'kinesistap'
if (![System.IO.Directory]::Exists($kinesistapTempDir)) {[void]
[System.IO.Directory]::CreateDirectory($kinesistapTempDir)}

#Download KinesisTap.x.x.x.x.nupkg package
$kinesistapNupkgPath = Join-Path $kinesistapTempDir $packageName
$webClient.DownloadFile($kinesistapBaseUrl + $packageName, $kinesistapNupkgPath)
$kinesistapUnzipPath = $kinesistapNupkgPath.Replace('.nupkg', '')

# Calculates hash of downloaded file. Downlevel compatible using .Net hashing on PS < 4
if ($PSVersionTable.PSVersion.Major -ge 4) {
    $calculatedHash = Get-FileHash $kinesistapNupkgPath -Algorithm SHA256
    $hashAsString = $calculatedHash.Hash.ToLower()
} else {
    $sha256 = New-Object System.Security.Cryptography.SHA256CryptoServiceProvider
    $calculatedHash =
 [System.BitConverter]::ToString($sha256.ComputeHash([System.IO.File]::ReadAllBytes($kinesistap
    $hashAsString = $calculatedHash.Replace("-", "").ToLower()
}

if ($checksum -eq $hashAsString) {
 Write-Host 'Local file hash matches checksum.' -ForegroundColor Green
} else {
 throw ("Get-FileHash does not match! Package may be corrupted.")
}

#Delete Unzip path if not empty
if ([System.IO.Directory]::Exists($kinesistapUnzipPath)) {Remove-Item –Path
 $kinesistapUnzipPath -Recurse -Force}

#Unzip KinesisTap.x.x.x.x.nupkg package
$null =
 [System.Reflection.Assembly]::LoadWithPartialName('System.IO.Compression.FileSystem')
[System.IO.Compression.ZipFile]::ExtractToDirectory($kinesistapNupkgPath,
 $kinesistapUnzipPath)

#Execute chocolaeyInstall.ps1 in the package and wait for completion.
$installScript = Join-Path $kinesistapUnzipPath '\tools\chocolateyInstall.ps1'
& $installScript

# Verify service installed.
```

```
$serviceName = 'AWSKinesisTap'
$service = Get-Service -Name $serviceName -ErrorAction Ignore
if ($null -eq $service) {
    throw ("Service not installed correctly.")
} else {
    Write-Host "Kinesis Tap Installed." -ForegroundColor Green
    Write-Host "After configuring run the following to start the service: Start-Service
 -Name $serviceName." -ForegroundColor Green
}
```

Abra uma janela de prompt de comando elevado (com privilégios administrativos). No diretório em que você fez download do arquivo, use o seguinte comando para executar o script:

```
PowerShell.exe -File ".\InstallKinesisAgent.ps1"
```

Para instalar uma versão específica do Kinesis Agent para Windows, adicione a `-version` Opção:

```
PowerShell.exe -File ".\InstallKinesisAgent.ps1" -version "version"
```

Substituir *`version`* com um número de versão válido do Kinesis Agent para Windows. Para obter informações sobre versões, consulte a repositório kinesis-agent-windows no GitHub.

Há muitas ferramentas de implantação que podem executar remotamente scripts do PowerShell. Eles podem ser usados para automatizar a instalação do Kinesis Agent para Windows em frotas de servidores ou desktops.

Configurando e iniciando o Kinesis Agent para Windows

Depois de instalar o Kinesis Agent para Windows, você deve configurar e iniciar o agente. Depois disso, nenhuma outra intervenção de operação deve ser necessária.

Para configurar e iniciar o Kinesis Agent para Windows

1. Crie e implante um arquivo de configuração do Kinesis Agent para Windows. Este arquivo configura origens, coletores e pipes junto com outros itens de configuração global.

   Para obter mais informações sobre a configuração do Kinesis Agent para Windows, consulte Configurando o Amazon Kinesis Agent para Microsoft Windows.

Para obter exemplos de arquivos de configuração que você pode personalizar e instalar, consulte Exemplos de configuração do Kinesis Agent para Windows.

2. Abra uma janela de prompt de comando elevado do PowerShell e inicie o Kinesis Agent para Windows usando o seguinte comando do PowerShell:

```
Start-Service -Name AWSKinesisTap
```

Configurando o Amazon Kinesis Agent para Microsoft Windows

Antes de iniciar o Amazon Kinesis Agent para Microsoft Windows, você deve criar um arquivo de configuração e implantá-lo. O arquivo de configuração fornece as informações necessárias para coletar, transformar e fazer streaming de dados em servidores e desktops Windows para vários serviços da AWS. Os arquivos de configuração definem conjuntos de origens, coletores e pipes que conectam origens a coletores, juntamente com transformações opcionais.

O arquivo de configuração do Kinesis Agent para Windows é chamado`appsettings.json`. Implante esse arquivo em `%PROGRAMFILES%\Amazon\AWSKinesisTap`.

Tópicos

Estrutura de configuração básica

A estrutura básica do arquivo de configuração do Amazon Kinesis Agent para Microsoft Windows é um documento JSON com o seguinte modelo:

```
{
     "Sources": [ ],
     "Sinks": [ ],
     "Pipes": [ ]
}
```

- O valor de `Sources` é um ou mais Declarações de origem.
- O valor de `Sinks` é um ou mais Declarações de coletor.
- O valor de `Pipes` é um ou mais Declarações de pipe.

Para obter mais informações sobre os conceitos de origem, pipe e coletor do Kinesis Agent para Windows, consulteConceitos do Amazon Kinesis Agent para Microsoft Windows.

O exemplo a seguir é um completo`appsettings.json`Configure o Kinesis Agent para Windows para fazer streaming de eventos de log de aplicativos do Windows para o Kinesis Data Firehose.

```
{
  "Sources": [
    {
      "LogName": "Application",
      "Id": "ApplicationLog",
      "SourceType": "WindowsEventLogSource"
    }
  ],
  "Sinks": [
    {
      "StreamName": "ApplicationLogFirehoseStream",
      "Region": "us-west-2",
      "Id": "MyKinesisFirehoseSink",
      "SinkType": "KinesisFirehose"
    }
  ],
  "Pipes": [
    {
      "Id": "ApplicationLogTotestKinesisFirehoseSink",
      "SourceRef": "ApplicationLog",
      "SinkRef": "MyKinesisFirehoseSink"
    }
  ]
}
```

Para obter informações sobre tipo de declaração, consulte as seguintes seções:

- Declarações de origem
- Declarações de coletor
- Declarações de pipe

Distinção de maiúsculas e minúsculas da configuração

Arquivos formatados JSON normalmente fazem distinção de maiúsculas de minúsculas, e você deve presumir que todas as chaves e valores em arquivos de configuração do Kinesis Agent para

Windows também façam essa distinção. Algumas chaves e valores no arquivo de configuração `appsettings.json` não fazem distinção de maiúsculas e minúsculas; por exemplo:

- O valor do par de chave-valor `Format` para coletores. Para obter mais informações, consulte Declarações de coletor.
- O valor do par de chave-valor `SourceType` de origens, o par de chave-valor `SinkType` para coletores e o par de chave-valor `Type` para pipes e plug-ins.
- O valor do par de chave-valor `RecordParser` para a origem `DirectorySource`. Para obter mais informações, consulte Configuração de DirectorySource.
- O valor do par de chave-valor `InitialPosition` para origens. Para obter mais informações, consulte Configuração de marcador.
- Prefixos para substituições de variáveis. Para obter mais informações, consulte Configuração de substituições de variáveis de coletor.

Declarações de origem

No Amazon Kinesis Agent para Microsoft Windows,Declarações de origemDescrever onde e o que os dados de log, eventos e métricas devem ser coletados. Elas também podem especificar informações para analisar esses dados para que possas ser transformados. As seções a seguir descrevem as configurações para os tipos de origem integrados que estão disponíveis no Kinesis Agent para Windows. Como o Kinesis Agent para Windows é extensível, você pode adicionar tipos de origem personalizados. Cada tipo de origem geralmente requer pares de chave-valor específicos nos objetos de configuração que são relevantes para esse tipo de origem.

Todas as declarações de origem devem conter pelo menos os seguintes pares de chave-valor:

`Id`

Uma string exclusiva que identifica um objeto de origem específico no arquivo de configuração.

`SourceType`

O nome do tipo de origem para esse objeto de origem. O tipo de origem especifica a origem dos dados de log, eventos ou métricas que estão sendo coletados por esse objeto de origem. Ele também controla quais outros aspectos da origem podem ser declarados.

Para obter exemplos de arquivos de configuração completos que usam tipos diferentes de declarações de origem, consulte Streaming de várias origens para o Kinesis Data Streams.

Tópicos

Configuração de DirectorySource

Overview

O tipo de origem `DirectorySource` reúne logs de arquivos armazenados no diretório especificado. Como os arquivos de log são fornecidos em vários formatos diferentes, a declaração `DirectorySource` permite que você especifique o formato dos dados no arquivo de log. Depois você pode transformar o conteúdo do log em um formato padrão, como JSON ou XML, antes do streaming para vários serviços da AWS.

Veja a seguir um exemplo de declaração `DirectorySource`:

```
{
    "Id": "myLog",
    "SourceType": "DirectorySource",
    "Directory": "C:\\Program Data\\MyCompany\\MyService\\logs",
    "FileNameFilter": "*.log",
    "IncludeSubdirectories": true,
    "IncludeDirectoryFilter": "cpu\\cpu-1;cpu\\cpu-2;load;memory",
    "RecordParser": "Timestamp",
    "TimestampFormat": "yyyy-MM-dd HH:mm:ss.ffff",
    "Pattern": "\\d{4}-\\d{2}-\\d(2}",
    "ExtractionPattern": "",
```

```
    "TimeZoneKind": "UTC",
    "SkipLines": 0,
    "Encoding": "utf-16",
    "ExtractionRegexOptions": "Multiline"
}
```

Todas as declarações `DirectorySource` podem fornecer os seguintes pares de chave/valor:

`SourceType`

Deve ser a string literal `"DirectorySource"` (obrigatória).

`Directory`

O caminho para o diretório que contém os arquivos de log (obrigatório).

`FileNameFilter`

Opcionalmente, limita o conjunto de arquivos no diretório em que os dados de log são coletados com base em um padrão de nomenclatura de arquivos curinga. Se você tiver vários padrões de nome de arquivo de log, esse recurso permite que você use um único`DirectorySource`, conforme mostrado no exemplo a seguir.

```
FileNameFilter: "*.log|*.txt"
```

Às vezes, os administradores de sistema compactam arquivos de log antes de arquivá-los. Se você especificar`"*.*"`em`FileNameFilter`, os arquivos compactados conhecidos agora são excluídos. Este recurso impede`.zip`, `.gz`, e `.bz2`sejam transmitidos acidentalmente. Se esse par de chave/valor não for especificado, os dados de todos os arquivos no diretório serão coletados por padrão.

`IncludeSubdirectories`

Especifica monitorar subdiretórios para profundidade arbitrária limitada pelo sistema operacional. Esse recurso é útil para monitorar servidores da Web com vários sites. Você também pode usar o`IncludeDirectoryFilter`para monitorar apenas determinados subdiretórios especificados no filtro.

`RecordParser`

Especifica como o tipo de origem `DirectorySource` deve analisar os arquivos de log que são encontrados no diretório especificado. Esse par de chave-valor é obrigatório, e os valores válidos são os seguintes:

- `SingleLine`— cada linha do arquivo de log é um registro de log.
- `SingleLineJson`— cada linha do arquivo de log é um registro de log formatado JSON. Esse analisador é útil quando você deseja adicionar outros pares de chave-valor ao objeto JSON usando decoração de objeto. Para obter mais informações, consulte Configuração de decorações de coletor. Para obter um exemplo que usa o analisador de registros `SingleLineJson`, consulte Tutorial: Transmitir arquivos de log JSON para o Amazon S3 usando o Kinesis Agent para Windows.
- `Timestamp`— uma ou mais linhas podem incluir um registro de log. O registro de log começa com um timestamp. Essa opção exige a especificação do par de chave-valor `TimestampFormat`.
- `Regex`— cada registro começa com o texto que corresponde a uma expressão regular específica. Essa opção exige a especificação do par de chave-valor `Pattern`.
- `SysLog`— Indica que o arquivo de log foi gravado nosyslogformato padrão. O arquivo de log é analisado em registros de acordo com essa especificação.
- `Delimited`— uma versão mais simples do analisador de registros Regex em que os itens de dados nos registros do log são separados por um delimitador consistente. Essa opção é mais fácil de usar e é executada com mais rapidez do que o analisador Regex, e é a opção preferencial quando está disponível. Ao usar essa opção, você deve especificar o par de chave-valor `Delimiter`.

`TimestampField`

Especifica qual campo JSON contém o timestamp para o registro. É usado somente com o `SingleLineJson` `RecordParser`. Esse par de chave/valor é opcional. Se não for especificado, o Kinesis Agent para Windows usará a hora em que o registro foi lido para o timestamp. Uma vantagem de especificar o par de chave-valor é que as estatísticas de latência geradas pelo Kinesis Agent para Windows são mais precisas.

`TimestampFormat`

Especifica como analisar a data e a hora associadas ao registro. O valor é a string `epoch` ou uma string de formato de data/hora .NET. Se o valor for `epoch`, o tempo será analisado com base no horário UNIX Epoch. Para obter mais informações sobre o horário UNIX Epoch, consulte Horário Unix. Para obter mais informações sobre strings de formato de data/hora do .NET, consulte Strings de formato de data e hora personalizadas na documentação do Microsoft .NET. Esse par de chave-valor será obrigatório somente se o analisador de registros `Timestamp` for especificado, ou o analisador de registros `SingleLineJson` for especificado junto com o par de chave-valor `TimestampField`.

`Pattern`

Especifica uma expressão regular que deve corresponder à primeira linha de um registro possivelmente de várias linhas. Esse par de chave-valor só é obrigatório para o analisador de registros `Regex`.

`ExtractionPattern`

Especifica uma expressão regular que deve usar grupos nomeados. O registro é analisado usando essa expressão regular e os grupos nomeados formam os campos do registro analisado. Esses campos são usados como base para criar objetos JSON ou XML ou documentos dos quais é feito streaming pelos coletores para vários serviços da AWS. Esse par de chave/valor é opcional e está disponível com o`Regex`e o analisador de carimbo de data/hora.

O nome do grupo `Timestamp` é especialmente processado, uma vez que ele indica para o analisador `Regex` qual campo contém a data e a hora para cada registro em cada arquivo de log.

`Delimiter`

Especifica o caractere ou a string que separa cada item em cada registro de log. Esse par de chave-valor deve e só pode ser usado com o analisador de registros `Delimited`. Use a sequência de dois caracteres `\t` para representar o caractere de tabulação.

`HeaderPattern`

Especifica uma expressão regular para corresponder à linha do arquivo de log que contém o conjunto de cabeçalhos para o registro. Se o arquivo de log não contiver informações de cabeçalho, use o par de chave-valor `Headers`para especificar os cabeçalhos implícitos. O par de chave-valor `HeaderPattern` é opcional e só é válido para o analisador de registros `Delimited`.

> **Note**
>
> Uma entrada de cabeçalho vazia (de comprimento 0) para uma coluna faz com que os dados dessa coluna sejam filtrados da saída final da saída analisada `DirectorySource`.

`Headers`

Especifica os nomes das colunas de dados analisados usando o delimitador especificado. Esse par de chave/valor é opcional e só é válido para o analisador de registros `Delimited`.

> **Note**
>
> Uma entrada de cabeçalho vazia (de comprimento 0) para uma coluna faz com que os dados dessa coluna sejam filtrados da saída final da saída analisada `DirectorySource`.

`RecordPattern`

Especifica uma expressão regular que identifica as linhas do arquivo de log que contêm dados de registro. Diferente da linha de cabeçalho opcional identificada por `HeaderPattern`, as linhas que não correspondem ao `RecordPattern` especificado são ignoradas durante o processamento de registros. Esse par de chave/valor é opcional e só é válido para o analisador de registros `Delimited`. Se não for fornecido, o padrão é considerar qualquer linha que não corresponda ao `HeaderPattern` opcional ou ao `CommentPattern` opcional uma linha que contém dados de registro que podem ser analisados.

`CommentPattern`

Especifica uma expressão regular que identifica linhas no arquivo de log que devem ser excluídas antes de analisar os dados no arquivo de log. Esse par de chave/valor é opcional e só é válido para o analisador de registros `Delimited`. Se não for fornecido, o padrão é considerar qualquer linha que não corresponda ao `HeaderPattern` uma linha que contém dados de registro que podem ser analisados, a menos que `RecordPattern` seja especificado.

`TimeZoneKind`

Especifica se o timestamp no arquivo de log deve ser considerado no fuso horário local ou UTC. Isso é opcional e usa como padrão UTC. Os únicos valores válidos para esse par de chave-valor são `Local` ou `UTC`. O timestamp nunca é alterado se `TimeZoneKind` não for especificado ou se o valor for UTC. O carimbo de data/hora é convertido para UTC quando a propriedade`TimeZoneKind`O valor é`Local`E o coletor que recebe o timestamp é o CloudWatch Logs ou o registro analisado é enviado a outros coletores. As datas e horas que são incorporadas em mensagens não são convertidas.

`SkipLines`

Quando especificado, controla o número de linhas ignoradas no início de cada arquivo de log antes da análise do registro. Isso é opcional e o valor padrão é 0.

Codificação

Por padrão, o Kinesis Agent para Windows pode detectar automaticamente a codificação do bytemark. No entanto, a codificação automática pode não funcionar corretamente em alguns formatos unicode mais antigos. O exemplo a seguir especifica a codificação necessária para transmitir um log do Microsoft SQL Server.

```
"Encoding": "utf-16"
```

Para obter uma lista de nomes de codificação, consulteLista de codificaçõesna documentação do Microsoft .NET.

ExtractionRegexOptions

Você pode usar`ExtractionRegexOptions`Para simplificar expressões regulares. Esse par de chave/valor é opcional. O padrão é `"None"`.

O exemplo a seguir especifica que o`"."`corresponde a qualquer caractere, incluindo`\r\n`.

```
"ExtractionRegexOptions" = "Multiline"
```

Para obter uma lista dos possíveis campos para ExtractionRegexOptions, consulte a propriedadeRegExOptions Enumna documentação do Microsoft .NET.

Analisador de registros **`Regex`**

Você pode analisar logs de texto não estruturado usando o analisador de registros `Regex` junto com os pares de chave-valor `TimestampFormat`, `Pattern` e `ExtractionPattern`. Por exemplo, digamos que seu arquivo de log tenha a seguinte aparência:

```
[FATAL][2017/05/03 21:31:00.534][0x00003ca8][0000059c][][ActivationSubSystem]
[GetActivationForSystemID][0] 'ActivationException.File: EQCASLicensingSubSystem.cpp'
[FATAL][2017/05/03 21:31:00.535][0x00003ca8][0000059c][][ActivationSubSystem]
[GetActivationForSystemID][0] 'ActivationException.Line: 3999'
```

Você pode especificar a expressão regular a seguir para o par de chave-valor `Pattern` para ajudar a dividir o arquivo de log em registros de log individuais:

```
^\[\w+\]\[(?<TimeStamp>\d{4}/\d{2}/\d{2} \d{2}:\d{2}:\d{2}\.\d{3})\]
```

Essa expressão regular corresponde à seguinte sequência:

1. O início da string que está sendo avaliada.
2. Um ou mais caracteres de palavras entre colchetes.
3. Um timestamp entre colchetes. O timestamp corresponde à seguinte sequência:
   a. Um ano com quatro dígitos
   b. Uma barra
   c. Um mês com dois dígitos
   d. Uma barra
   e. Um dia com dois dígitos
   f. Um caractere de espaço
   g. Uma hora com dois dígitos
   h. Dois pontos
   i. Minuto com dois dígitos
   j. Dois pontos
   k. Segundo com dois dígitos
   l. Um ponto
   m. Milissegundo com três dígitos

Você pode especificar o seguinte formato para o par de chave-valor `TimestampFormat` para converter o timestamp textual em uma data e hora:

```
yyyy/MM/dd HH:mm:ss.fff
```

Você pode usar a seguinte expressão regular para extrair os campos do registro de log por meio do par de chave-valor `ExtractionPattern`.

```
^\[(?<Severity>\w+)\]\[(?<TimeStamp>\d{4}/\d{2}/\d{2} \d{2}:\d{2}:\d{2}\.
\d{3})\]\[[^]]*\]\[[^]]*\]\[[^]]*\]\[(?<SubSystem>\w+)\]\[(?<Module>\w+)\]\[[^]]*\] '(?
<Message>.*)'$
```

Essa expressão regular corresponde aos seguintes grupos em sequência:

1. `Severity`— um ou mais caracteres de palavras entre colchetes.
2. `TimeStamp`— consulte a descrição anterior para o timestamp.
3. Três sequências entre colchetes de zeros ou mais caracteres são ignoradas.
4. `SubSystem`— um ou mais caracteres de palavras entre colchetes.
5. `Module`— um ou mais caracteres de palavras entre colchetes.
6. Uma sequência entre colchetes de zeros ou mais caracteres é ignorada.
7. Um espaço sem nome é ignorado.
8. `Message`— Zero ou mais caracteres entre aspas simples.

A seguinte declaração de origem combina essas expressões regulares e o formato de data e hora para fornecer as instruções completas para o Kinesis Agent para Windows para analisar esse tipo de arquivo de log.

```
{
    "Id": "PrintLog",
    "SourceType": "DirectorySource",
    "Directory": "C:\\temp\\PrintLogTest",
    "FileNameFilter": "*.log",
    "RecordParser": "Regex",
    "TimestampFormat": "yyyy/MM/dd HH:mm:ss.fff",
    "Pattern": "^\\[\\w+\\]\\[(?<TimeStamp>\\d{4}/\\d{2}/\\d{2} \\d{2}:\\d{2}:\\d{2}\\.
\\d{3})\\]",
    "ExtractionPattern": "^\\[(?<Severity>\\w+)\\]\\[(?<TimeStamp>\\d{4}/\\d{2}/\\d{2}
 \\d{2}:\\d{2}:\\d{2}\\.\\d{3})\\]\\[[^]]*\\]\\[[^]]*\\]\\[[^]]*\\]\\[(?<SubSystem>\\w
+)\\]\\[(?<Module>\\w+)\\]\\[[^]]*\\] '(?<Message>.*)'$",
    "TimeZoneKind": "UTC"
}
```

> **Note**
>
> As barras invertidas nos arquivos em formato JSON devem ser precedidas por uma barra invertida adicional como caractere de escape.

Para obter mais informações sobre expressões regulares, consulte Linguagem de expressões regulares – referência rápida na documentação do Microsoft .NET.

Analisador de registros **`Delimited`**

Você pode usar o analisador de registros `Delimited` para analisar arquivos de log e dados semiestruturados nos quais há uma sequência de caracteres consistente separando cada coluna de dados em cada linha. Por exemplo, arquivos CSV usam uma vírgula para separar cada coluna de dados, e arquivos TSV usam uma guia.

Suponha que você queira analisar um arquivo de log no formato de banco de dados NPS da Microsoft produzido por um servidor de política de rede. Esse arquivo pode ser o seguinte:

```
"NPS-
MASTER","IAS",03/22/2018,23:07:55,1,"user1","Domain1\user1",,,,,,,,0,"192.168.86.137","Nate
 - Test 1",,,,,,,1,,0,"311 1 192.168.0.213 03/15/2018 08:14:29
 1",,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,"Use Windows authentication for all users",1,,,,
"NPS-
MASTER","IAS",03/22/2018,23:07:55,3,,"Domain1\user1",,,,,,,,0,"192.168.86.137","Nate
 - Test 1",,,,,,,1,,16,"311 1 192.168.0.213 03/15/2018 08:14:29
 1",,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,"Use Windows authentication for all users",1,,,,
```

O exemplo de arquivo de configuração `appsettings.json` a seguir inclui uma declaração `DirectorySource` que usa o analisador de registros `Delimited` para analisar esse texto em uma representação de objeto. Depois ele faz streaming de dados com formato JSON para o Kinesis Data Firehose:

```
{
    "Sources": [
        {
            "Id": "NPS",
            "SourceType": "DirectorySource",
            "Directory": "C:\\temp\\NPS",
            "FileNameFilter": "*.log",
            "RecordParser": "Delimited",
            "Delimiter": ",",
            "Headers": "ComputerName,ServiceName,Record-Date,Record-Time,Packet-
Type,User-Name,Fully-Qualified-Distinguished-Name,Called-Station-ID,Calling-Station-
ID,Callback-Number,Framed-IP-Address,NAS-Identifier,NAS-IP-Address,NAS-Port,Client-
Vendor,Client-IP-Address,Client-Friendly-Name,Event-Timestamp,Port-Limit,NAS-Port-
Type,Connect-Info,Framed-Protocol,Service-Type,Authentication-Type,Policy-Name,Reason-
Code,Class,Session-Timeout,Idle-Timeout,Termination-Action,EAP-Friendly-Name,Acct-
Status-Type,Acct-Delay-Time,Acct-Input-Octets,Acct-Output-Octets,Acct-Session-Id,Acct-
Authentic,Acct-Session-Time,Acct-Input-Packets,Acct-Output-Packets,Acct-Terminate-
Cause,Acct-Multi-Ssn-ID,Acct-Link-Count,Acct-Interim-Interval,Tunnel-Type,Tunnel-
```

```
Medium-Type,Tunnel-Client-Endpt,Tunnel-Server-Endpt,Acct-Tunnel-Conn,Tunnel-Pvt-
Group-ID,Tunnel-Assignment-ID,Tunnel-Preference,MS-Acct-Auth-Type,MS-Acct-EAP-Type,MS-
RAS-Version,MS-RAS-Vendor,MS-CHAP-Error,MS-CHAP-Domain,MS-MPPE-Encryption-Types,MS-
MPPE-Encryption-Policy,Proxy-Policy-Name,Provider-Type,Provider-Name,Remote-Server-
Address,MS-RAS-Client-Name,MS-RAS-Client-Version",
            "TimestampField": "{Record-Date} {Record-Time}",
            "TimestampFormat": "MM/dd/yyyy HH:mm:ss"
        }
    ],
    "Sinks": [
        {
            "Id": "npslogtest",
            "SinkType": "KinesisFirehose",
            "Region": "us-west-2",
            "StreamName": "npslogtest",
            "Format": "json"
        }
    ],
    "Pipes": [
        {
            "Id": "W3SVCLog1ToKinesisStream",
            "SourceRef": "NPS",
            "SinkRef": "npslogtest"
        }
    ]
}
```

Dados com formato JSON dos quais foi feito streaming para o Kinesis Data Firehose se parecem com o seguinte:

```
{
    "ComputerName": "NPS-MASTER",
    "ServiceName": "IAS",
    "Record-Date": "03/22/2018",
    "Record-Time": "23:07:55",
    "Packet-Type": "1",
    "User-Name": "user1",
    "Fully-Qualified-Distinguished-Name": "Domain1\\user1",
    "Called-Station-ID": "",
    "Calling-Station-ID": "",
    "Callback-Number": "",
    "Framed-IP-Address": "",
    "NAS-Identifier": "",
```

```
    "NAS-IP-Address": "",
    "NAS-Port": "",
    "Client-Vendor": "0",
    "Client-IP-Address": "192.168.86.137",
    "Client-Friendly-Name": "Nate - Test 1",
    "Event-Timestamp": "",
    "Port-Limit": "",
    "NAS-Port-Type": "",
    "Connect-Info": "",
    "Framed-Protocol": "",
    "Service-Type": "",
    "Authentication-Type": "1",
    "Policy-Name": "",
    "Reason-Code": "0",
    "Class": "311 1 192.168.0.213 03/15/2018 08:14:29 1",
    "Session-Timeout": "",
    "Idle-Timeout": "",
    "Termination-Action": "",
    "EAP-Friendly-Name": "",
    "Acct-Status-Type": "",
    "Acct-Delay-Time": "",
    "Acct-Input-Octets": "",
    "Acct-Output-Octets": "",
    "Acct-Session-Id": "",
    "Acct-Authentic": "",
    "Acct-Session-Time": "",
    "Acct-Input-Packets": "",
    "Acct-Output-Packets": "",
    "Acct-Terminate-Cause": "",
    "Acct-Multi-Ssn-ID": "",
    "Acct-Link-Count": "",
    "Acct-Interim-Interval": "",
    "Tunnel-Type": "",
    "Tunnel-Medium-Type": "",
    "Tunnel-Client-Endpt": "",
    "Tunnel-Server-Endpt": "",
    "Acct-Tunnel-Conn": "",
    "Tunnel-Pvt-Group-ID": "",
    "Tunnel-Assignment-ID": "",
    "Tunnel-Preference": "",
    "MS-Acct-Auth-Type": "",
    "MS-Acct-EAP-Type": "",
    "MS-RAS-Version": "",
    "MS-RAS-Vendor": "",
```

```
    "MS-CHAP-Error": "",
    "MS-CHAP-Domain": "",
    "MS-MPPE-Encryption-Types": "",
    "MS-MPPE-Encryption-Policy": "",
    "Proxy-Policy-Name": "Use Windows authentication for all users",
    "Provider-Type": "1",
    "Provider-Name": "",
    "Remote-Server-Address": "",
    "MS-RAS-Client-Name": "",
    "MS-RAS-Client-Version": ""
}
```

Analisador de registros **SysLog**

Para o analisador de registros `SysLog`, a saída analisada da origem inclui as seguintes informações:

Atributo	Tipo	Descrição
`SysLogTimeStamp`	String	A data e a hora originais do arquivo de log com formato syslog.
`Hostname`	String	O nome do computador no qual reside o arquivo de log em formato syslog.
`Program`	String	O nome do aplicativo ou do serviço que gerou o arquivo de log.
`Message`	String	A mensagem de log gerada pelo aplicativo ou serviço.
`TimeStamp`	String	A data e hora analisadas no formato ISO 8601.

Veja a seguir um exemplo de dados SysLog transformados em JSON:

```
{
    "SysLogTimeStamp": "Jun 18 01:34:56",
    "Hostname": "myhost1.example.mydomain.com",
    "Program": "mymailservice:",
```

```
    "Message": "Info: ICID 123456789 close",
    "TimeStamp": "2017-06-18T01:34.56.000"
}
```

Summary

Veja a seguir um resumo dos pares de chave/valor disponíveis para a origem `DirectorySource` e dos `RecordParsers` relacionados com esses pares de chave/valor.

Nome da chave	RecordParser	Observações
`SourceType`	Obrigatório para todos	Deve ter o valor `DirectorySource`
`Directory`	Obrigatório para todos	
`FileNameFilter`	Opcional para todos	
`RecordParser`	Obrigatório para todos	
`TimestampField`	Opcional para `SingleLineJson`	
`TimestampFormat`	Obrigatório para `Timestamp` e para `SingleLineJson` se `TimestampField` for especificado	
`Pattern`	Obrigatório para `Regex`	
`ExtractionPattern`	Opcional para `Regex`	Obrigatório para `Regex` se o coletor especificar o formato `json` ou `xml`

Nome da chave	RecordParser	Observações
`Delimiter`	Obrigatório para `Delimited`	
`HeaderPattern`	Opcional para `Delimited`	
`Headers`	Opcional para `Delimited`	
`RecordPattern`	Opcional para `Delimited`	
`CommentPattern`	Opcional para `Delimited`	
`TimeZoneKind`	Opcional para `Regex`, `Timestamp` , `SysLog` e `SingleLineJson` quando um campo de timestamp é identificado	
`SkipLines`	Opcional para todos	

Configuração de ExchangeLogSource

O tipo `ExchangeLogSource` é usado para coletar logs do Microsoft Exchange. O Exchange produz logs em vários tipos diferentes de formatos de log. Esse tipo de origem analisa todos eles. Embora seja possível analisá-los usando o tipo `DirectorySource` com o analisador de registros `Regex`, é muito mais simples de usar o `ExchangeLogSource`. Isso ocorre porque você não precisa projetar e fornecer expressões regulares para os formatos de arquivo de log. Veja a seguir um exemplo de declaração `ExchangeLogSource`:

```
{
   "Id": "MyExchangeLog",
   "SourceType": "ExchangeLogSource",
   "Directory": "C:\\temp\\ExchangeLogTest",
   "FileNameFilter": "*.log"
```

```
}
```

Todas as declarações do Exchange podem fornecer os seguintes pares de chave/valor:

`SourceType`

Deve ser a string literal `"ExchangeLogSource"` (obrigatória).

`Directory`

O caminho para o diretório que contém os arquivos de log (obrigatório).

`FileNameFilter`

Opcionalmente, limita o conjunto de arquivos no diretório em que os dados de log são coletados com base em um padrão de nomenclatura de arquivos curinga. Se esse par de chave/valor não for especificado, por padrão, os dados de log de todos os arquivos no diretório serão coletados.

`TimestampField`

O nome da coluna que contém a data e a hora para o registro. Esse par de chave-valor é opcional e não precisará ser especificado se o nome do campo for `date-time` ou `DateTime`. Caso contrário, ele será obrigatório.

Configuração de W3SVCLogSource

O tipo `W3SVCLogSource` é usado para coletar logs do Internet Information Services (IIS) para Windows.

Veja a seguir um exemplo de declaração `W3SVCLogSource`:

```
{
   "Id": "MyW3SVCLog",
   "SourceType": "W3SVCLogSource",
   "Directory": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
   "FileNameFilter": "*.log"
}
```

Todas as declarações `W3SVCLogSource` podem fornecer os seguintes pares de chave/valor:

`SourceType`

Deve ser a string literal `"W3SVCLogSource"` (obrigatória).

`Directory`

O caminho para o diretório que contém os arquivos de log (obrigatório).

`FileNameFilter`

Opcionalmente, limita o conjunto de arquivos no diretório em que os dados de log são coletados com base em um padrão de nomenclatura de arquivos curinga. Se esse par de chave/valor não for especificado, por padrão, os dados de log de todos os arquivos no diretório serão coletados.

Configuração de UlsSource

O tipo `UlsSource` é usado para coletar logs do Microsoft SharePoint. Veja a seguir um exemplo de declaração `UlsSource`:

```
{
    "Id": "UlsSource",
    "SourceType": "UlsSource",
    "Directory": "C:\\temp\\uls",
    "FileNameFilter": "*.log"
}
```

Todas as declarações `UlsSource` podem fornecer os seguintes pares de chave/valor:

`SourceType`

Deve ser a string literal `"UlsSource"` (obrigatória).

`Directory`

O caminho para o diretório que contém os arquivos de log (obrigatório).

`FileNameFilter`

Opcionalmente, limita o conjunto de arquivos no diretório em que os dados de log são coletados com base em um padrão de nomenclatura de arquivos curinga. Se esse par de chave/valor não for especificado, por padrão, os dados de log de todos os arquivos no diretório serão coletados.

Configuração de WindowsEventLogSource

O tipo `WindowsEventLogSource` é usado para coletar eventos do serviço de log de eventos do Windows. Veja a seguir um exemplo de declaração `WindowsEventLogSource`:

```
{
    "Id": "mySecurityLog",
    "SourceType": "WindowsEventLogSource",
    "LogName": "Security"
}
```

Todas as declarações `WindowsEventLogSource` podem fornecer os seguintes pares de chave/valor:

`SourceType`

Deve ser a string literal `"WindowsEventLogSource"` (obrigatória).

`LogName`

Os eventos são coletados do log especificado. Os valores comuns incluem `Application`, `Security` e `System`, mas você pode especificar qualquer nome de log de eventos válido do Windows. Esse par de chave/valor é obrigatório.

`Query`

Opcionalmente limita os eventos que são gerados do `WindowsEventLogSource`. Se esse par de chave/valor não for especificado, por padrão, todos os eventos serão gerados. Para obter informações sobre a sintaxe desse valor, verifique Consultas de eventos e Event XML na documentação do Windows. Para obter informações sobre definições no nível de log, consulte Tipos de eventos na documentação do Windows.

`IncludeEventData`

Opcionalmente, permite a coleta e o streaming de dados de eventos específicos do provedor associados a eventos do log de eventos especificado do Windows quando o valor desse par de chave/valor é `"true"`. Somente dados de eventos que podem ser serializados com êxito são incluídos. Esse par de chave-valor é opcional e, se não for especificado, os dados do evento específico do provedor não serão coletados.

> **Note**
>
> Incluir dados de eventos pode aumentar significativamente a quantidade de dados dos quais é feito streaming da origem. O tamanho máximo de um evento pode ser 262.143 bytes com dados de eventos incluídos.

A saída analisada do `WindowsEventLogSource` contém as seguintes informações:

Atributo	Tipo	Descrição
`EventId`	Int	O identificador do tipo de evento.
`Description`	String	Texto que descreve os detalhes do evento.
`LevelDisplayName`	String	A categoria do evento (um de Erro, Aviso, Informação, Auditoria de êxito, Auditoria de falha).
`LogName`	String	Se o evento foi registrado (os valores típicos são `Application` , `Security` e `System`, mas há muitas possibilidades).
`MachineName`	String	Qual computador registrou o evento.
`ProviderName`	String	Qual aplicativo ou serviço registrou o evento.
`TimeCreated`	String	Quando o evento ocorreu no formato ISO 8601.
`Index`	Int	Onde a entrada está localizada no log.
`UserName`	String	Quem fez a entrada, se for conhecido.
`Keywords`	String	O tipo de evento. Os valores padrão incluem `AuditFailure` (eventos de auditoria de segurança com falha), `AuditSuccess` (eventos de auditoria de segurança bem-suced ida), `Classic` (eventos gerados com a função `RaiseEvent`), `Correlati on Hint` (eventos de transferê ncia), `SQM` (eventos do Service Quality

Atributo	Tipo	Descrição
		Mechanism), `WDI Context` (eventos do contexto da Windows Diagnostic Infrastructure) e `WDI Diag` (eventos de diagnóstico da Windows Diagnostic Infrastructure).
`EventData`	Lista de objetos	Dados extras opcionais específicos do provedor sobre o evento de log. Só serão incluídos se o valor para o par de chave/valor `IncludeEventData` for `"true"`.

Veja a seguir um exemplo de evento transformado em JSON:

```
{[
    "EventId": 7036,
    "Description": "The Amazon SSM Agent service entered the stopped state.",
    "LevelDisplayName": "Informational",
    "LogName": "System",
    "MachineName": "mymachine.mycompany.com",
    "ProviderName": "Service Control Manager",
    "TimeCreated": "2017-10-04T16:42:53.8921205Z",
    "Index": 462335,
    "UserName": null,
    "Keywords": "Classic",
    "EventData": [
    "Amazon SSM Agent",
    "stopped",
    "rPctBAMZFhYubF8zVLcrBd3bTTcNzHvY5Jc2Br0aMrxxx=="
]}
```

Configuração WindowsEventLogPollingSource

`WindowsEventLogPollingSource`usa um mecanismo baseado em sondagem para reunir todos os novos eventos do log de eventos que correspondem aos parâmetros configurados. O intervalo de sondagem é atualizado dinamicamente entre 100 ms e 5000 ms dependendo de quantos

eventos foram coletados durante a última pesquisa. Veja a seguir um exemplo de declaração `WindowsEventLogPollingSource`:

```
{
    "Id": "MySecurityLog",
    "SourceType": "WindowsEventLogPollingSource",
    "LogName": "Security",
    "IncludeEventData": "true",
    "Query": "",
    "CustomFilters": "ExcludeOwnSecurityEvents"
}
```

Todas as declarações `WindowsEventLogPollingSource` podem fornecer os seguintes pares de chave/valor:

`SourceType`

Deve ser a string literal "`WindowsEventLogPollingSource`" (obrigatória).

`LogName`

Especifica o log. As opções válidas são`Application`,`Security`,`System`ou outros logs válidos.

`IncludeEventData`

Optional. Quando`true`, especifica que EventData extra quando transmitido como JSON e XML é incluído. O padrão é `false`.

`Query`

Optional. Os logs de eventos do Windows suportam consultas de eventos usando expressões XPath, que você pode especificar usando`Query`. Para obter mais informações, consulte[Consultas de eventos e XML de eventos]na documentação da Microsoft.

`CustomFilters`

Optional. Lista de filtros separados por ponto e vírgula (`;`). Os seguintes filtros podem ser especificados.

`ExcludeOwnSecurityEvents`

Exclui eventos de segurança gerados pelo Kinesis Agent para Windows propriamente dito.

Configuração de WindowsETWEventSource

O tipo `WindowsETWEventSource` é usado para coletar rastreamentos de eventos de serviço e de aplicativos usando um recurso chamado Rastreamento de eventos para Windows (ETW). Para obter mais informações, consulte Rastreamento de eventos na documentação do Windows.

Veja a seguir um exemplo de declaração `WindowsETWEventSource`:

```
{
    "Id": "ClrETWEventSource",
    "SourceType": "WindowsETWEventSource",
    "ProviderName": "Microsoft-Windows-DotNETRuntime",
    "TraceLevel": "Verbose",
    "MatchAnyKeyword": 32768
}
```

Todas as declarações `WindowsETWEventSource` podem fornecer os seguintes pares de chave/valor:

`SourceType`

Deve ser a string literal `"WindowsETWEventSource"` (obrigatória).

`ProviderName`

Especifica qual provedor de evento a ser usado para coletar eventos de rastreamento. Deve ser um nome de provedor ETW válido para um provedor instalado. Para determinar quais provedores estão instalados, execute o seguinte em uma janela de prompt de comando do Windows:

```
logman query providers
```

`TraceLevel`

Especifica quais categorias de eventos de rastreamento devem ser coletadas. Os valores permitidos são `Critical`, `Error`, `Warning`, `Informational` e `Verbose`. O significado exato depende do provedor ETW que está selecionado.

`MatchAnyKeyword`

Esse valor é um número de 64 bits, em que cada bit representa uma palavra-chave individual. Cada palavra-chave descreve uma categoria de eventos a serem coletados. Para as palavras-chave com suporte e seus valores e como elas se relacionam com `TraceLevel`, consulte a

documentação do provedor. Por exemplo, para obter informações sobre o provedor ETW CLR, consulte Palavras-chave de CLR ETW e níveis na documentação do Microsoft .NET Framework.

No exemplo anterior, 32768 (0x00008000) representa a `ExceptionKeyword` para o provedor ETW CLR que instrui o provedor para coletar informações sobre as exceções lançadas. Embora o JSON não ofereça suporte nativamente a constantes hexadecimais, você pode especificá-las para `MatchAnyKeyword` colocando-as em uma string. Você também pode especificar várias constantes separadas por vírgulas. Por exemplo, use o seguinte para especificar a `ExceptionKeyword` e a `SecurityKeyword` (0x00000400):

```
{
   "Id": "MyClrETWEventSource",
   "SourceType": "WindowsETWEventSource",
   "ProviderName": "Microsoft-Windows-DotNETRuntime",
   "TraceLevel": "Verbose",
   "MatchAnyKeyword": "0x00008000, 0x00000400"
}
```

Para garantir que todas as palavras-chave especificadas sejam ativadas para um provedor, vários valores de palavra-chave são combinados usando OR e são transmitidos para esse provedor.

A saída de `WindowsETWEventSource` contém as seguintes informações de cada evento:

Atributo	Tipo	Descrição
`EventName`	String	Que tipo de evento ocorrido.
`ProviderName`	String	Qual provedor detectou o evento.
`FormattedMessage`	String	Um texto de resumo do evento.
`ProcessID`	Int	Qual processo relatou o evento.
`ExecutingThreadID`	Int	Qual thread dentro do processo relatou o evento.
`MachineName`	String	O nome do desktop ou do servidor que está relatando o evento.

Atributo	Tipo	Descrição
`Payload`	Tabela de hash	Uma tabela com uma chave de string e qualquer tipo de objeto como um valor. A chave é o nome do item de carga, e o valor é o valor do item de carga. A carga depende do provedor.

Veja a seguir um exemplo de evento transformado em JSON:

```
{
     "EventName": "Exception/Start",
     "ProviderName": "Microsoft-Windows-DotNETRuntime",
     "FormattedMessage": "ExceptionType=System.Exception;\r
\nExceptionMessage=Intentionally unhandled exception.;\r\nExceptionEIP=0x2ab0499;\r
\nExceptionHRESULT=-2,146,233,088;\r\nExceptionFlags=CLSCompliant;\r\nClrInstanceID=9
 ",
     "ProcessID": 3328,
     "ExecutingThreadID": 6172,
     "MachineName": "MyHost.MyCompany.com",
     "Payload":
      {
        "ExceptionType": "System.Exception",
        "ExceptionMessage": "Intentionally unhandled exception.",
        "ExceptionEIP": 44762265,
        "ExceptionHRESULT": -2146233088,
        "ExceptionFlags": 16,
        "ClrInstanceID": 9
      }
}
```

Configuração de WindowsPerformanceCounterSource

O tipo `WindowsPerformanceCounterSource` coleta métricas do contador de desempenho do Windows. Veja a seguir um exemplo de declaração `WindowsPerformanceCounterSource`:

```
{
 "Id": "MyPerformanceCounter",
 "SourceType": "WindowsPerformanceCounterSource",
```

```
 "Categories": [{
   "Category": "Server",
   "Counters": ["Files Open", "Logon Total", "Logon/sec", "Pool Nonpaged Bytes"]
  },
  {
   "Category": "System",
   "Counters": ["Processes", "Processor Queue Length", "System Up Time"]
  },
  {
   "Category": "LogicalDisk",
   "Instances": "*",
   "Counters": [
    "% Free Space", "Avg. Disk Queue Length",
    {
     "Counter": "Disk Reads/sec",
     "Unit": "Count/Second"
    },
    "Disk Writes/sec"
   ]
  },
  {
   "Category": "Network Adapter",
   "Instances": "^Local Area Connection\* \d$",
   "Counters": ["Bytes Received/sec", "Bytes Sent/sec"]
  }
 ]
}
```

Todas as declarações `WindowsPerformanceCounterSource` podem fornecer os seguintes pares de chave/valor:

`SourceType`

Deve ser a string literal "`WindowsPerformanceCounterSource`" (obrigatória).

`Categories`

Especifica um conjunto de grupos de métricas do contador de desempenho a serem reunidos do Windows. Cada grupo de métricas contém os seguintes pares de chave/valor:

`Category`

Especifica o conjunto de métricas do contador a serem coletadas (obrigatório).

`Instances`

Especifica o conjunto de objetos de interesse quando há um conjunto exclusivo de contadores de desempenho por objeto. Por exemplo, quando a categoria é `LogicalDisk`, há um conjunto de contadores de desempenho por unidade de disco. Esse par de chave/valor é opcional. Você pode usar os curingas `*` e `?` para fazer a correspondência com várias instâncias. Para valores agregados em todas as instâncias, especifique `_Total`.

Você também pode usar`InstanceRegex`, que aceita expressões regulares que contêm o`*`caractere curinga como parte do nome da ocorrência.

`Counters`

Especifica quais métricas a serem reunidas para a categoria especificada. Esse par de chave/valor é obrigatório. Você pode usar os curingas `*` e `?` para fazer a correspondência com vários contadores. Você pode especificar `Counters` usando apenas o nome e a unidade. Se as unidades não forem especificados, o Kinesis Agent para Windows tentará inferir as unidades a partir do nome. Se essas inferências estiverem incorretas, a unidade poderá ser especificada explicitamente. Se desejar, você poderá alterar os nomes `Counter`. A representação mais complexa de um contador é um objeto com os seguintes pares de chave/valor:

`Counter`

O nome do contador. Esse par de chave/valor é obrigatório.

`Rename`

O nome do contador a ser apresentado ao coletor. Esse par de chave/valor é opcional.

`Unit`

O significado do valor que está associado ao contador. Para obter uma lista completa de nomes de unidade válidos, consulte a documentação da unidade em[MetricDatum](#)noReferência de API do Amazon CloudWatch.

Veja a seguir um exemplo de uma especificação complexa de um contador:

```
{
   "Counter": "Disk Reads/sec,
   "Rename": "Disk Reads per second",
   "Unit": "Count/Second"
}
```

`WindowsPerformanceCounterSource`pode ser usado apenas com um pipe que especifique um coletor do Amazon CloudWatch. Use um coletor separado se também for feito streaming das métricas incorporadas do Kinesis Agent para Windows para o CloudWatch. Examine o log do Kinesis Agent para o Windows após a inicialização do serviço para determinar quais unidades foram inferidas para contadores quando as unidades não foram especificados no`WindowsPerformanceCounterSource`Declarações. Use o PowerShell para determinar os nomes válidos para categorias, instâncias e contadores.

Para ver as informações sobre todas as categorias, incluindo contadores associados a conjuntos de contadores, execute este comando em uma janela do PowerShell:

```
Get-Counter -ListSet * | Sort-Object
```

Para determinar quais instâncias estão disponíveis para cada um dos contadores do conjunto, execute um comando semelhante ao exemplo a seguir em uma janela do PowerShell:

```
Get-Counter -Counter "\Process(*)\% Processor Time"
```

O valor do parâmetro `Counter` deve ser um dos caminhos de um membro `PathsWithInstances` listado pela invocação do comando `Get-Counter -ListSet` anterior.

Origem de métricas incorporadas do Kinesis Agent para Windows

Além de fontes de métricas comuns, como o`WindowsPerformanceCounterSource`tipo (consulteConfiguração de WindowsPerformanceCounterSource), o tipo de coletor do CloudWatch do pode receber métricas de uma origem especial que reúne métricas sobre o próprio Kinesis Agent para Windows. As métricas do Kinesis Agent para Windows também estão disponíveis no`KinesisTap`categoria de contadores de desempenho do Windows.

O`MetricsFilter`O par de chave-valor para as declarações do coletor do CloudWatch do especifica de quais métricas é feito streaming do CloudWatch da origem de métricas incorporadas do Kinesis Agent para Windows. O valor é uma string que contém uma ou mais expressões de filtro separadas por ponto-e-vírgula; por exemplo:

`"MetricsFilter": "FilterExpression1;FilterExpression2"`

É feito streaming de uma métrica que corresponde a uma ou mais expressões de filtro para o CloudWatch.

As métricas de instância única são globais por natureza e não são vinculadas a uma origem ou coletor específico. Várias métricas de instância são dimensionais com base no `Id` da declaração de origem ou de coletor. Cada tipo de origem ou coletor pode ter um conjunto diferente de métricas.

Para obter uma lista de nomes de métricas incorporadas do Kinesis Agent para Windows, consulteLista de métricas do agente Kinesis para Windows.

Para métricas de instância única, a expressão de filtro é o nome da métrica, por exemplo:

```
"MetricsFilter": "SourcesFailedToStart;SinksFailedToStart"
```

Para várias métricas de instância, a expressão de filtro é o nome da métrica, um ponto (`.`) e depois o `Id` da origem ou da declaração de coletor que gerou essa métrica. Por exemplo, supondo-se que haja uma declaração de coletor com um `Id` de `MyFirehose`:

```
"MetricsFilter": "KinesisFirehoseRecordsFailedNonrecoverable.MyFirehose"
```

Você pode usar padrões de curinga especiais que são projetados para distinguir entre métricas de várias instâncias ou de uma instância única.

- O asterisco (`*`) corresponde a zero ou mais caracteres, exceto ponto (`.`).
- O ponto de interrogação (`?`) corresponde a um caractere exceto ponto.
- Qualquer outro caractere apenas corresponde a si mesmo.
- `_Total` é um token especial que causa a agregação de todos os valores de várias instâncias correspondentes em toda a dimensão.

O exemplo a seguir corresponde a todas as métricas de instância única:

```
"MetricsFilter": "*"
```

Como um asterisco não corresponde ao período de caractere, apenas as métricas de instância são incluídas.

O exemplo a seguir corresponde a várias métricas de instância:

```
"MetricsFilter": "*.*"
```

O exemplo a seguir corresponde a todas as métricas (instância única e várias instâncias):

```
"MetricsFilter": "*;*.*"
```

O exemplo a seguir agrega todas as métricas de várias instâncias em todas as origens e coletores:

```
"MetricsFilter": "*._Total"
```

O exemplo a seguir agrega todas as métricas do Kinesis Data Firehose para todos os coletores do Kinesis Data Firehose:

```
"MetricsFilter": "*Firehose*._Total"
```

O exemplo a seguir corresponde a todas as métricas de erro de instância única e de várias instâncias:

```
"MetricsFilter": "*Failed*;*Error*.*;*Failed*.*"
```

O exemplo a seguir corresponde a todas as métricas de erro não recuperável agregadas em todas as origens e coletores:

```
"MetricsFilter": "*Nonrecoverable*._Total"
```

Para obter informações sobre como especificar um pipe que use a origem de métricas incorporadas do Kinesis Agent para Windows, consulteConfiguração do Kinesis Agent para Pipes Métricos do Windows.

Lista de métricas do agente Kinesis para Windows

Veja a seguir uma lista de métricas de uma instância única e de várias instâncias que estão disponíveis para o Kinesis Agent para Windows.

Métricas de instância única

As seguintes métricas de instância única estão disponíveis:

`KinesisTapBuildNumber`

O número da versão do Kinesis Agent para Windows.

`PipesConnected`

Quantos pipes conectaram sua origem ao coletor com êxito.

`PipesFailedToConnect`

Quantos pipes conectaram sua origem ao coletor sem êxito.

`SinkFactoriesFailedToLoad`

Quantos tipos de coletores não foram carregados para o Kinesis Agent para Windows com êxito.

`SinkFactoriesLoaded`

Quantos tipos de coletores foram carregados para o Kinesis Agent para Windows com êxito.

`SinksFailedToStart`

Quantos coletores não foram iniciados com êxito, geralmente devido a declarações de coletor incorretas.

`SinksStarted`

Quantos coletores foram iniciados com êxito.

`SourcesFailedToStart`

Quantas origens não foram iniciadas com êxito, geralmente devido a declarações de origem incorretas.

`SourcesStarted`

Quantas origens foram iniciadas com êxito.

`SourceFactoriesFailedToLoad`

Quantos tipos de origens não foram carregados para o Kinesis Agent para Windows com êxito.

`SourceFactoriesLoaded`

Quantos tipos de origens foram carregados para o Kinesis Agent para Windows.

Métricas de várias instâncias

A métricas de várias instâncias a seguir estão disponíveis:

Métricas DirectorySource

`DirectorySourceBytesRead`

Quantos bytes foram lidos durante o intervalo para essa `DirectorySource`.

`DirectorySourceBytesToRead`

Quantos números conhecidos de bytes estão disponíveis para leitura que ainda não foram lidos pelo Kinesis Agent para Windows.

`DirectorySourceFilesToProcess`

Quantos arquivos conhecidos que ainda não foram examinados pelo Kinesis Agent para Windows.

`DirectorySourceRecordsRead`

Quantos registros foram lidos durante o intervalo para essa `DirectorySource`.

Métricas WindowsEventLogSource

`EventLogSourceEventsError`

Quantos eventos de log de eventos do Windows não foram lidos com êxito.

`EventLogSourceEventsRead`

Quantos eventos de log de eventos do Windows foram lidos com êxito.

Métricas do coletor KinesisFirehose

`KinesisFirehoseBytesAccepted`

Quantos bytes foram aceitos durante o intervalo.

`KinesisFirehoseClientLatency`

Quanto tempo decorreu entre a geração e o streaming dos registros para o serviço Kinesis Data Firehose.

`KinesisFirehoseLatency`

Quanto tempo decorreu entre o início e o fim do streaming dos registros para o serviço Kinesis Data Firehose.

`KinesisFirehoseNonrecoverableServiceErrors`

Quantas vezes não foi possível enviar os registros sem erro para o serviço Kinesis Data Firehose apesar das novas tentativas.

`KinesisFirehoseRecordsAttempted`

De quantos registros houve tentativa de streaming para o serviço de Data Firehose do Kinesis.

`KinesisFirehoseRecordsFailedNonrecoverable`

De quantos registros o streaming não foi feito com êxito para o serviço de Kinesis Data Firehose apesar das novas tentativas.

`KinesisFirehoseRecordsFailedRecoverable`

De quantos registros o streaming foi feito com êxito para o serviço de Kinesis Data Firehose, mas apenas com novas tentativas.

`KinesisFirehoseRecordsSuccess`

De quantos registros o streaming foi feito com êxito para o serviço de Kinesis Data Firehose sem novas tentativas.

`KinesisFirehoseRecoverableServiceErrors`

Quantas vezes foi possível enviar os registros com êxito para o serviço de Kinesis Data Firehose, mas apenas com novas tentativas.

Métricas KinesisStream

`KinesisStreamBytesAccepted`

Quantos bytes foram aceitos durante o intervalo.

`KinesisStreamClientLatency`

Quanto tempo decorreu entre a geração e o streaming dos registros para o serviço de Streams de Dados do Kinesis.

`KinesisStreamLatency`

Quanto tempo decorreu entre o início e o fim do streaming dos registros para o serviço de Streams de Dados do Kinesis.

`KinesisStreamNonrecoverableServiceErrors`

Quantas vezes não foi possível enviar os registros sem erro para o serviço de Streams de Dados do Kinesis apesar das novas tentativas.

`KinesisStreamRecordsAttempted`

De quantos registros houve tentativa de streaming para o serviço de Streams de dados do Kinesis.

`KinesisStreamRecordsFailedNonrecoverable`

De quantos registros o streaming não foi feito com êxito para o serviço de Kinesis Data Streams apesar das novas tentativas.

`KinesisStreamRecordsFailedRecoverable`

De quantos registros o streaming foi feito com êxito para o serviço de Streams de Dados do Kinesis, mas apenas com novas tentativas.

`KinesisStreamRecordsSuccess`

De quantos registros o streaming foi feito com êxito para o serviço de Streams de dados do Kinesis sem novas tentativas.

`KinesisStreamRecoverableServiceErrors`

Quantas vezes foi possível enviar os registros com êxito para o serviço de Kinesis Data Streams, mas apenas com novas tentativas.

Métricas do CloudWatchLog

`CloudWatchLogBytesAccepted`

Quantos bytes foram aceitos durante o intervalo.

`CloudWatchLogClientLatency`

Quanto tempo decorreu entre a geração e o streaming dos registros para o serviço CloudWatch Logs.

`CloudWatchLogLatency`

Quanto tempo decorreu entre o início e o fim do streaming dos registros para o serviço CloudWatch Logs.

`CloudWatchLogNonrecoverableServiceErrors`

Quantas vezes não foi possível enviar os registros sem erro para o serviço do CloudWatch Logs, apesar de repetidas tentativas.

`CloudWatchLogRecordsAttempted`

De quantos registros houve tentativa de streaming para o serviço CloudWatch Logs.

`CloudWatchLogRecordsFailedNonrecoverable`

De quantos registros o streaming não foi feito com êxito para o serviço CloudWatch Logs apesar das novas tentativas.

`CloudWatchLogRecordsFailedRecoverable`

De quantos registros o streaming foi feito com êxito para o serviço CloudWatch Logs, mas apenas com novas tentativas.

`CloudWatchLogRecordsSuccess`

De quantos registros o streaming foi feito com êxito para o serviço CloudWatch Logs sem novas tentativas.

`CloudWatchLogRecoverableServiceErrors`

Quantas vezes foi possível enviar os registros com êxito para o serviço do CloudWatch Logs, mas apenas com novas tentativas.

Métricas do CloudWatch

`CloudWatchLatency`

Quanto tempo em média decorreu entre o início e o fim do streaming de métricas para o serviço do CloudWatch.

`CloudWatchNonrecoverableServiceErrors`

Quantas vezes não foi possível enviar as métricas sem erro para o serviço CloudWatch apesar das novas tentativas.

`CloudWatchRecoverableServiceErrors`

Quantas vezes as métricas foram enviadas sem erro para o serviço CloudWatch, mas apenas com novas tentativas.

`CloudWatchServiceSuccess`

Quantas vezes as métricas foram enviadas sem erro para o serviço CloudWatch sem a necessidade de novas tentativas.

Configuração de marcador

Por padrão, o Kinesis Agent para Windows envia registros de log para coletores que são criados depois que o agente é iniciado. Às vezes é útil enviar registros de log anteriores, por exemplo, registros de log criados durante a interrupção do Agente do Kinesis para o Windows durante uma atualização automática. O recurso de marcador controla quais registros foram enviados para coletores. Quando o Kinesis Agent para Windows está no modo de marcador e é iniciado, ele envia todos os registros de log criados depois que o Kinesis Agent para Windows foi interrompido, junto com todos os registros de log criados posteriormente. Para controlar esse comportamento, as declarações de origem baseada em arquivo podem os seguintes pares de chave-valor:

`InitialPosition`

Especifica a situação inicial para o marcador. Os valores possíveis são:

`EOS`

Especifica o fim do streaming (EOS). Somente os registros de log criados enquanto o agente está em execução são enviados para coletores.

`0`

Todos os registros de log e eventos disponíveis são inicialmente enviados. Depois um marcador é criado para garantir que todos os novos registros de log e eventos criados depois que o marcador foi criado sejam enviados, esteja ou não o Kinesis Agent para Windows em execução.

`Bookmark`

O marcador é inicializado apenas após o último registro de log ou evento. Depois um marcador é criado para garantir que todos os novos registros de log e eventos criados depois que o marcador foi criado sejam enviados, esteja ou não o Kinesis Agent para Windows em execução.

Os marcadores estão habilitados por padrão. Os arquivos são armazenados no`%ProgramData%\Amazon\KinesisTap`diretório.

`Timestamp`

Os registros de log e eventos que são criados após o valor `InitialPositionTimestamp` (segue definição) são enviados. Depois um marcador é criado para garantir que todos os novos registros de log e eventos criados depois que o marcador foi criado sejam enviados, esteja ou não o Kinesis Agent para Windows em execução.

`InitialPositionTimestamp`

Especifica o primeiro registro de log ou timestamp de evento mais antigo que você desejar. Especifique esse par chave-valor somente quando `InitialPosition` tem um valor de `Timestamp`.

`BookmarkOnBufferFlush`

Essa configuração pode ser adicionada a qualquer fonte de favoritos. Quando definido como`true`, garante que as atualizações de marcadores ocorram somente quando um coletor envia com êxito um evento para a AWS. Você só pode inscrever um único coletor em uma fonte. Se você estiver enviando logs para vários destinos, duplique suas fontes para evitar possíveis problemas com a perda de dados.

Quando o Kinesis Agent para Windows foi interrompido por um longo período, pode ser necessário excluir esses marcadores porque os registros de log e eventos marcados podem não existir mais. Arquivos de marcador para um id de origem estão localizados em `%PROGRAMDATA%\Amazon\AWSKinesisTap\`*`source id`*`.bm`.

Os marcadores não funcionam em arquivos renomeados ou truncados. Devido à natureza dos eventos ETW e contadores de desempenho, eles não podem ser marcados.

Declarações de coletor

Declarações de coletor especificam onde e de que forma os logs, eventos e métricas devem ser enviados para vários serviços da AWS. As seções a seguir descrevem as configurações para os tipos de coletor integrados que estão disponíveis no Amazon Kinesis Agent para Microsoft Windows. Como o Kinesis Agent para Windows é extensível, você pode adicionar tipos de coletor personalizados. Cada tipo de coletor geralmente requer pares de chave-valor exclusivos nas declarações de configuração que são relevantes para esse tipo de coletor.

Todas as declarações de coletor podem conter os seguintes pares de chave/valor:

`Id`

Uma string exclusiva que identifica um determinado coletor no arquivo de configuração (obrigatório).

`SinkType`

O nome do tipo desse coletor (obrigatório). O tipo de coletor especifica o destino do log, dos dados de métricas ou eventos que estão sendo transmitidos por esse coletor.

`AccessKey`

Especifica a chave de acesso da AWS a ser usada ao autorizar o acesso ao serviço da AWS associado ao tipo de coletor. Esse par de chave/valor é opcional. Para obter mais informações, consulte Configuração de segurança do coletor.

`SecretKey`

Especifica a chave secreta da AWS para ser usada ao autorizar o acesso ao serviço da AWS associado ao tipo de coletor. Esse par de chave/valor é opcional. Para obter mais informações, consulte Configuração de segurança do coletor.

`Region`

Especifica qual região da AWS contém os recursos de destino para streaming. Esse par de chave/valor é opcional.

`ProfileName`

Especifica qual perfil da AWS a ser usado para autenticação. Esse par de chave-valor é opcional, mas se for especificado, ele qualquer chave de acesso e chave secreta especificadas. Para obter mais informações, consulte Configuração de segurança do coletor.

`RoleARN`

Especifica a função do IAM a ser usada ao acessar o serviço da AWS associado ao tipo de coletor. Essa opção é útil quando o Kinesis Agent para Windows está em execução em uma instância do EC2, mas uma função diferente seria mais apropriada do que a função referenciada pelo perfil de instância. Por exemplo, uma função entre contas pode ser usada para acessar recursos que não estão na mesma conta da AWS que a instância do EC2. Esse par de chave/valor é opcional.

`Format`

Especifica o tipo de serialização que é aplicada aos logs e aos dados de eventos antes do streaming. Os valores válidos são `json` e `xml`. Essa opção é útil quando a análise de

downstream no pipeline de dados exige ou prefere dados em um determinado formato. Esse par de chave-valor é opcional e, se não for especificado, será feito streaming do texto normal da origem do coletor para o serviço da AWS associado ao tipo de coletor.

`TextDecoration`

Quando nenhum `Format` está especificado, o `TextDecoration` determina o texto adicional que deverá ser incluído ao fazer streaming de logs ou de registros de eventos. Para obter mais informações, consulte Configuração de decorações de coletor. Esse par de chave/valor é opcional.

`ObjectDecoration`

Quando `Format` está especificado, `ObjectDecoration` determina quais dados adicionais estão incluídos no log ou no registro de evento antes da serialização e do streaming. Para obter mais informações, consulte Configuração de decorações de coletor. Esse par de chave/valor é opcional.

`BufferInterval`

Para minimizar chamadas de API para o serviço da AWS associado ao tipo de coletor, o Kinesis Agent para Windows armazena em buffer vários registros de log, eventos e métricas antes do streaming. Dessa forma, você pode economizar dinheiro em caso de serviços nos quais é feita cobrança por chamada de API. O `BufferInterval` especifica o tempo máximo (em segundos) durante o qual os registros devem ser armazenados em buffer antes do streaming para o serviço da AWS. Esse par de chave-valor é opcional e, se especificado, use uma string para representar o valor.

`BufferSize`

Para minimizar chamadas de API para o serviço da AWS associado ao tipo de coletor, o Kinesis Agent para Windows armazena em buffer vários registros de log, eventos e métricas antes do streaming. Dessa forma, você pode economizar dinheiro em caso de serviços nos quais é feita cobrança por chamada de API. O `BufferSize` especifica o número máximo de registros a serem armazenados em buffer antes do streaming para o serviço da AWS. Esse par de chave-valor é opcional e, se for especificado, use uma string para representar o valor.

`MaxAttempts`

Especifica o número máximo de vezes que o Kinesis Agent para Windows tenta fazer streaming de um conjunto de registros de log, eventos e métricas para um serviço da AWS se ocorrer uma falha de forma consistente no streaming. Esse par de chave/valor é opcional. Se ele for especificado, use uma string para representar o valor. O valor padrão é "3".

Para obter exemplos de arquivos de configuração completos que usam vários tipos de coletores, consulte Streaming do log de eventos de aplicativos do Windows para coletores.

Tópicos

Configuração do coletor **`KinesisStream`**

O`KinesisStream`O tipo de coletor faz streaming de registros de log e eventos para o serviço de Streams de dados do Kinesis. Normalmente, os dados dos quais é feito streaming para Kinesis Data Streams são processados por um ou mais aplicativos personalizados que são executados com vários serviços da AWS. É feito streaming dos dados para um determinado stream que é configurado com o Kinesis Data Streams. Para obter mais informações, consulte o .Guia do desenvolvedor do Amazon Kinesis Data Streams.

Veja a seguir um exemplo de declaração de coletor do Kinesis Data Streams:

```
{
    "Id": "TestKinesisStreamSink",
    "SinkType": "KinesisStream",
    "StreamName": "MyTestStream",
```

```
    "Region": "us-west-2"
}
```

Todas as declarações de coletor `KinesisStream` podem fornecer os seguintes pares de chave/valor adicionais:

`SinkType`

Deve ser especificado, e o valor deve ser a string literal `KinesisStream`.

`StreamName`

Especifica o nome do stream de dados do Kinesis que recebe os dados dos quais é feito streaming do`KinesisStream`tipo de pia (obrigatório). Antes do streaming dos dados, configure o stream no AWS Management Console, na CLI da AWS ou por meio de um aplicativo com a API do Kinesis Data Streams.

`RecordsPerSecond`

Especifica o número máximo de registros dos quais foi feito streaming para Kinesis Data Streams por segundo. Esse par de chave/valor é opcional. Se ele for especificado, use um valor inteiro para representar o valor. O valor padrão é 1.000 registros.

`BytesPerSecond`

Especifica o número máximo de bytes dos quais foi feito streaming para Kinesis Data Streams por segundo. Esse par de chave/valor é opcional. Se ele for especificado, use um valor inteiro para representar o valor. O valor padrão é 1 MB.

O `BufferInterval` padrão para esse tipo de coletor é 1 segundo, e o `BufferSize` padrão é 500 registros.

Configuração do coletor **`KinesisFirehose`**

O`KinesisFirehose`O tipo de coletor faz streaming de registros de log e eventos para o serviço Kinesis Data Firehose. O Kinesis Data Firehose fornece dados transmitidos para outros serviços para armazenamento. Normalmente, os dados armazenados são analisados em estágios subsequentes do pipeline de dados. É feito streaming dos dados para um stream de entrega que é configurado com o Kinesis Data Firehose. Para obter mais informações, consulte o .Guia do desenvolvedor do Amazon Kinesis Data Firehose.

Veja a seguir um exemplo de declaração de coletor do Kinesis Data Firehose do:

```
{
   "Id": "TestKinesisFirehoseSink",
   "SinkType": "KinesisFirehose",
   "StreamName": "MyTestFirehoseDeliveryStream",
   "Region": "us-east-1",
   "CombineRecords": "true"
}
```

Todas as declarações de coletor `KinesisFirehose` podem fornecer os seguintes pares de chave/valor adicionais:

`SinkType`

Deve ser especificado, e o valor deve ser a string literal `KinesisFirehose`.

`StreamName`

Especifica o nome do stream de entrega do Kinesis Data Firehose que recebe os dados dos quais é feito streaming do`KinesisStream`tipo de pia (obrigatório). Antes do streaming dos dados, configure o stream de entrega usando o AWS Management Console, a CLI da AWS ou por meio de um aplicativo com a API do Kinesis Data Firehose.

`CombineRecords`

Quando definido como`true`, especifica combinar vários registros pequenos em um registro grande com um tamanho máximo de 5 KB. Esse par de chave/valor é opcional. Os registros combinados usando esta função são separados por`\n`. Se você usar o AWS Lambda para transformar um registro do Kinesis Data Firehose, sua função do Lambda precisará considerar o caractere separador.

`RecordsPerSecond`

Especifica o número máximo de registros dos quais é feito streaming para Kinesis Data Streams por segundo. Esse par de chave/valor é opcional. Se ele for especificado, use um valor inteiro para representar o valor. O valor padrão é 5.000 registros.

`BytesPerSecond`

Especifica o número máximo de bytes dos quais é feito streaming para Kinesis Data Streams por segundo. Esse par de chave/valor é opcional. Se ele for especificado, use um valor inteiro para representar o valor. O valor padrão é 5 MB.

O `BufferInterval` padrão para esse tipo de coletor é 1 segundo, e o `BufferSize` padrão é 500 registros.

Configuração do CloudWatch

O`CloudWatch`O tipo de coletor faz streaming de métricas para o serviço do CloudWatch. É possível visualizar as métricas no AWS Management Console do. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch.

Veja a seguir um exemplo de declaração de coletor do `CloudWatch`:

```
{
   "Id": "CloudWatchSink",
   "SinkType": "CloudWatch"
}
```

Todas as declarações de coletor `CloudWatch` podem fornecer os seguintes pares de chave/valor adicionais:

`SinkType`

Deve ser especificado, e o valor deve ser a string literal `CloudWatch`.

`Interval`

Especifica com que frequência (em segundos) o Kinesis Agent for Windows relata métricas para o serviço do CloudWatch. Esse par de chave/valor é opcional. Se ele for especificado, use um valor inteiro para representar o valor. O valor padrão é de 60 segundos. Especifique 1 segundo se quiser métricas do CloudWatch de alta resolução.

`Namespace`

Especifica o namespace do CloudWatch no qual os dados das métricas são relatados. Os namespaces do CloudWatch agrupam um conjunto de métricas. Esse par de chave/valor é opcional. O valor padrão é `KinesisTap`.

`Dimensions`

Especifica as dimensões do CloudWatch usadas para isolar conjuntos de métricas em um namespace. Isso pode ser útil para fornecer conjuntos separados de dados de métricas para

cada desktop ou servidor, por exemplo. Esse par de chave-valor é opcional e, se for especificado, o valor deverá estar em conformidade com o seguinte formato: "key1=value1;key2=value2...". O valor padrão é "ComputerName={computername};InstanceId={instance_id}". Esse valor oferece suporte à substituição de variável de coletor. Para obter mais informações, consulte Configuração de substituições de variáveis de coletor.

`MetricsFilter`

Especifica de quais métricas é feito streaming para o CloudWatch da origem de métricas incorporadas do Kinesis Agent para Windows. Para obter mais informações sobre a origem de métricas integrada do Kinesis Agent para Windows, incluindo os detalhes da sintaxe do valor desse par de chave/valor, consulteOrigem de métricas incorporadas do Kinesis Agent para Windows.

Configuração do coletor **`CloudWatchLogs`**

O`CloudWatchLogs`O tipo de coletor faz streaming de registros de log e eventos para o Amazon CloudWatch Logs É possível visualizar os logs no AWS Management Console ou processá-los por meio de estágios adicionais de um pipeline de dados. É feito streaming dos dados para um stream de logs que é configurado no CloudWatch Logs. Os streams de log são organizados em grupos de logs. Para obter mais informações, consulte o .Amazon CloudWatch Logs.

Veja a seguir um exemplo de declaração de coletor do CloudWatch Logs:

```
{
   "Id": "MyCloudWatchLogsSink",
   "SinkType": "CloudWatchLogs",
   "BufferInterval": "60",
   "BufferSize": "100",
   "Region": "us-west-2",
   "LogGroup": "MyTestLogGroup",
   "LogStream": "MyTestStream"
}
```

Todas as declarações do coletor `CloudWatchLogs` devem fornecer os seguintes pares de chave/valor adicionais:

`SinkType`

Deve ser a string literal `CloudWatchLogs`.

`LogGroup`

Especifica o nome do grupo de logs do CloudWatch Logs que contém o stream de logs que recebe os registros de log e eventos dos quais o faz streaming pelo`CloudWatchLogs`tipo de pia. Se o grupo de logs especificado não existir, o Kinesis Agent para Windows tentará criá-lo.

`LogStream`

Especifica o nome do stream de logs do CloudWatch Logs que recebe o stream de registros de log e eventos pelo`CloudWatchLogs`tipo de pia. Esse valor oferece suporte à substituição de variável de coletor. Para obter mais informações, consulte Configuração de substituições de variáveis de coletor. Se o stream de logs especificado não existir, o Kinesis Agent para Windows tentará criá-lo.

O `BufferInterval` padrão para esse tipo de coletor é 1 segundo, e o `BufferSize` padrão é 500 registros. O tamanho máximo é 10.000 registros.

Local**`FileSystem`**Configuração do coletor

O tipo de pia`FileSystem`salva registros de log e eventos em um arquivo no sistema de arquivos local em vez de transmiti-los para os serviços da AWS.`FileSystem`são úteis para testes e diagnósticos. Por exemplo, você pode usar esse tipo de coletor para examinar registros antes de enviá-los para a AWS.

com`FileSystem`, você também pode usar parâmetros de configuração para simular lotes, limitação e retry-on-error para imitar o comportamento de pias reais da AWS.

Todos os registros de todas as fontes conectadas a um`FileSystem`são salvos no único arquivo especificado como`FilePath`. Se`FilePath`Não especificado, os registros serão salvos em um arquivo chamado*`SinkId`*`.txt`no`%TEMP%`diretório, que normalmente é`C:\Users\`*`UserName`*`\AppData\Local\Temp`, onde*`SinkId`*é o identificador exclusivo do coletor e*`UserName`*é o nome de usuário ativo do Windows.

Este tipo de coletor suporta atributos de decoração de texto. Para obter mais informações, consulte Configuração de decorações de coletor.

Um exemplo`FileSystem`A configuração do tipo de coletor é mostrada no exemplo a seguir.

```
{
    "Id": "LocalFileSink",
```

```
    "SinkType": "FileSystem",
    "FilePath": "C:\\ProgramData\\Amazon\\local_sink.txt",
    "Format": "json",
    "TextDecoration": "",
    "ObjectDecoration": ""
}
```

OFileSystemA configuração consiste nos seguintes pares de chave/valor.

`SinkType`

Deve ser a string literal `FileSystem`.

`FilePath`

Especifica o caminho e o arquivo onde os registros são salvos. Esse par de chave/valor é opcional. Se não especificado, o padrão será*TempPath*`\\`*SinkId*`.txt`, onde*TempPath*é a pasta armazenada no`%TEMP%`Variável e*SinkId*é o identificador exclusivo do coletor.

`Format`

Especifica o formato do evento a ser`json`ou`xml`. Esse par de valor de chave é opcional e não diferencia maiúsculas e minúsculas. Se omitidos, os eventos são gravados no arquivo em texto simples.

`TextDecoration`

Aplica-se apenas a eventos escritos em texto simples. Esse par de chave/valor é opcional.

`ObjectDecoration`

Aplica-se somente a eventos em que`Format`é definido como`json`. Esse par de chave/valor é opcional.

Uso Avançado — Controle de Registros e Simulação de Falha

`FileSystem`pode imitar o comportamento dos pias da AWS simulando a limitação de registros. É possível usar os seguintes pares de chave/valor para especificar atributos de simulação de falha e limitação de registros.

Ao adquirir um bloqueio no arquivo de destino e impedir gravações nele, você pode usar`FileSystem`para simular e examinar o comportamento dos pias da AWS quando a rede falha.

O exemplo a seguir mostra um`FileSystem`configuração com atributos de simulação.

```
{
    "Id": "LocalFileSink",
    "SinkType": "FileSystem",
    "FilePath": "C:\\ProgramData\\Amazon\\local_sink.txt",
    "TextDecoration": "",
    "RequestsPerSecond": "100",
    "BufferSize": "10",
    "MaxBatchSize": "1024"
}
```

`RequestsPerSecond`

Opcional e especificado como um tipo de string. Se omitido, o padrão será`"5"`. Controla a taxa de solicitações que o coletor processa, ou seja, grava no arquivo, não o número de registros. O Kinesis Agent para Windows faz solicitações em lote para endpoints da AWS, portanto, uma solicitação pode conter vários registros.

`BufferSize`

Opcional e especificado como tipo de string. Especifica o número máximo de registros de eventos que o coletor agrupa antes de salvar no arquivo.

`MaxBatchSize`

Opcional e especificado como um tipo de string. Especifica a quantidade máxima de dados de registro de evento em bytes que os lotes do coletor antes de salvar no arquivo.

O limite máximo de taxa de registro é uma função de`BufferSize`, o que determina o número máximo de registros por solicitação, e`RequestsPerSecond`. Você pode calcular o limite de taxa de registro por segundo usando a seguinte fórmula.

RecordRate=`BufferSize`*`RequestsPerSecond`

Dados os valores de configuração no exemplo acima, há uma taxa de registro máxima de 1000 registros por segundo.

Configuração de segurança do coletor

Como configurar a autenticação da

Para o Kinesis Agent for Windows fazer streaming de logs, eventos e métricas para os serviços da AWS, o acesso deve ser autenticado. Existem diversas formas de fornecer autenticação para o

Kinesis Agent para Windows. A forma como você faz isso depende da situação em que o Kinesis Agent para Windows está em execução e dos requisitos de segurança específicos para uma organização.

- Se o Kinesis Agent para Windows estiver em execução em um host do Amazon EC2, a maneira mais simples e segura de fornecer autenticação é criar uma função do IAM com acesso suficiente às operações necessárias para os serviços da AWS e um perfil de instância do EC2 que faça referência a essa função. Para obter informações sobre a criação de perfis da instância, consulte Uso de perfis de instância. Para obter informações sobre quais políticas associar à função do IAM, consulteConfiguração da autorização da.

  Depois de criar o perfil de instância, você pode associá-lo a todas as instâncias do EC2 que usam o Kinesis Agent para Windows. Se as instâncias já tiverem um perfil de instância associado, você poderá associar as políticas adequadas à função que estiver associada a esse perfil.
- Se o Kinesis Agent para Windows for executado em um host do EC2 em uma conta, mas os recursos que forem o destino do coletor residirem em uma conta diferente, você poderá criar uma função do IAM para acesso entre contas. Para obter mais informações, consulteTutorial: Delegar acesso entre contas da AWS usando funções do IAM. Depois de criar a função entre contas, especifique o nome de recurso da Amazon (ARN) para a função entre contas como o valor do`RoleARN` par de chave-valor na declaração de coletor. Depois, o Kinesis Agent para Windows tenta assumir a função entre contas especificada ao acessar os recursos da AWS associados ao tipo desse coletor.
- Se o Kinesis Agent para Windows estiver em execução fora do Amazon EC2 (por exemplo, no local), há várias opções:
  - Se for aceitável registrar o servidor no local ou uma máquina de desktop como uma instância gerenciada pelo Amazon EC2 Systems Manager, use o processo a seguir para configurar a autenticação:
    1. Use o processo descrito em Configuração do AWS Systems Manager em ambientes híbridos para criar uma função de serviço, criar uma ativação para uma instância gerenciada e instalar o agente do SSM.
    2. Associe as políticas adequadas à função de serviço para permitir que o Kinesis Agent para Windows acesse os recursos necessários para fazer streaming de dados dos coletores configurados. Para obter informações sobre quais políticas associar à função do IAM, consulteConfiguração da autorização da.

3. Use o processo descrito emConfigurar oProfileRefreshingAWSCredentialProviderAtualizar credenciais da AWSpara atualizar as credenciais da AWS.

Essa é a abordagem recomendada para instâncias que não sejam do EC2 porque as credenciais são gerenciadas de forma segura pelo SSM e pela AWS.

- Se for aceitável executar o serviço AWSKinesisTap para o Kinesis Agent para Windows em um usuário específico em vez da conta padrão do sistema, use o processo a seguir:

  1. Crie um usuário do IAM na conta da AWS na qual os serviços da AWS serão usados. Capture a chave de acesso e a chave secreta desse usuário durante o processo de criação. Você precisará dessas informações para etapas posteriores desse processo.
  2. Associe políticas ao usuário do IAM do que autorizem o acesso às operações necessárias para os serviços exigidos. Para obter informações sobre quais políticas associar ao usuário do IAM, consulteConfiguração da autorização da.
  3. Altere o serviço AWSKinesisTap em cada desktop ou servidor para que ele seja executado em um usuário específico em vez da conta do sistema padrão.
  4. Crie um perfil no armazenamento de SDKs usando a chave de acesso e a chave secreta registradas anteriormente. Para obter mais informações, consulte Configuração das credenciais da AWS.
  5. Atualize o arquivo `AWSKinesisTap.exe.config` no diretório `%PROGRAMFILES%\Amazon\AWSKinesisTap` para especificar o nome do perfil criado na etapa anterior. Para obter mais informações, consulte Configuração das credenciais da AWS.

  Essa é a abordagem recomendada para hosts que não sejam do EC2 que não podem ser instâncias gerenciadas, pois as credenciais são criptografadas para o host e o usuário específicos.

- Se for necessário executar o serviço AWSKinesisTap para o Kinesis Agent para Windows na conta padrão do sistema, use um arquivo de credenciais compartilhado. Isso ocorre porque a conta do sistema não tem perfil de usuário do Windows para habilitar o armazenamento de SDKs. Os arquivos de credenciais compartilhados não são criptografados, portanto, não recomendamos essa abordagem. Para obter informações sobre como usar arquivos de configuração compartilhados, consulteConfigurar as credenciais da AWSnoAWS SDK para .NET. Se você usar essa abordagem, recomendamos usar a criptografia NTFS e o acesso de arquivos restrito ao arquivo de configuração compartilhado. As chaves devem ser alteradas por uma plataforma de gerenciamento, e o arquivo de configuração compartilhado deve ser atualizado quando ocorre a alteração de chaves.

Embora seja possível fornecer diretamente as chaves de acesso e chaves secretas nas declarações de coletor, essa abordagem não é recomendada porque as declarações não são criptografadas.

Configuração da autorização da

Associe as políticas adequadas que se seguem ao usuário do IAM ou à função do que o Kinesis Agent para Windows usará para fazer streaming de dados para serviços da AWS:

Kinesis Data Streams

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord",
                "kinesis:PutRecords"
            ],
            "Resource": "arn:aws:kinesis:*:*:stream/*"
        }
    ]
}
```

Para limitar a autorização a uma região específica, a conta ou o nome do stream, substitua os asteriscos apropriados no ARN pelos valores específicos. Para obter mais informações, consulte "Nomes de recursos da Amazon (ARNs) para streams de dados do Kinesis" em Controle de acesso aos recursos de stream de dados do Amazon Kinesis usando o IAM.

Kinesis Data Firehose

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
```

```
            ],
            "Resource": "arn:aws:firehose:*:*:deliverystream/*"
        }
    ]
}
```

Para limitar a autorização a uma região, conta ou o nome do fluxo de entrega específico, substitua os asteriscos apropriados no ARN pelos valores específicos. Para obter mais informações, consulteControlar o acesso com o Amazon Kinesis Data FirehosenoGuia do desenvolvedor do Amazon Kinesis Data Firehose.

CloudWatch

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor2",
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*"
        }
    ]
}
```

Para obter mais informações, consulteVisão geral do gerenciamento de permissões de acesso aos recursos do CloudWatchnoAmazon CloudWatch Logs.

O CloudWatch Logs com um grupo e stream de logs existentes

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor3",
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
```

```
            "Resource": "arn:aws:logs:*:*:log-group:*"
        },
        {
            "Sid": "VisualEditor4",
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:*:*:log-group:*:*:*"
        }
    ]
}
```

Para restringir o acesso a uma região, conta, grupo de logs ou stream de logs específico, substitua os asteriscos apropriados nos ARNs pelos valores apropriados. Para obter mais informações, consulteVisão geral do gerenciamento de permissões de acesso aos recursos do CloudWatch LogsnoAmazon CloudWatch Logs.

CloudWatch Logs com permissões extras para o Kinesis Agent for Windows para criar grupos de logs e streams de logs

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor5",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:*"
        },
        {
            "Sid": "VisualEditor6",
            "Effect": "Allow",
            "Action": "logs:PutLogEvents",
            "Resource": "arn:aws:logs:*:*:log-group:*:*:*"
        },
        {
            "Sid": "VisualEditor7",
            "Effect": "Allow",
```

```
            "Action": "logs:CreateLogGroup",
            "Resource": "*"
        }
    ]
}
```

Para restringir o acesso a uma região, conta, grupo de logs ou stream de logs específico, substitua os asteriscos apropriados nos ARNs pelos valores apropriados. Para obter mais informações, consulteVisão geral do gerenciamento de permissões de acesso aos recursos do CloudWatch LogsnoAmazon CloudWatch Logs.

Permissões necessárias para a expansão de variáveis de tags do EC2

O uso da expansão de variáveis com o prefixo de variável `ec2tag` requer a permissão `ec2:Describe*`.

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Sid": "VisualEditor8",
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource": "*"
    }
   ]
}
```

> **Note**
>
> Você pode combinar várias instruções em uma única política, desde que o `Sid` de cada declaração seja exclusivo na política. Para obter informações sobre como criar políticas, consulteCriar políticas do IAMnoGuia do usuário do IAM.

Configurar o**`ProfileRefreshingAWSCredentialProvider`**Atualizar credenciais da AWS

Se você usar o AWS Systems Manager para ambientes híbridos para gerenciar credenciais da AWS, o Systems Manager girará as credenciais de sessão no`c:\Windows\System32\config`

`\systemprofile\.aws\credentials`. Para obter mais informações sobre o Systems Manager para ambientes híbridos, consulteConfigurando o AWS Systems Manager para ambientes híbridosnoGuia do usuário do AWS Systems Manager.

Como o AWS .net SDK não pega novas credenciais automaticamente, fornecemos o`ProfileRefreshingAWSCredentialProvider`para atualizar credenciais.

Você pode usar o`CredentialRef`de qualquer configuração de sincronização da AWS para fazer referência a um`Credentials`em que o`CredentialType`Atributo`ProfileRefreshingAWSCredentialProvider`Como mostrado no exemplo a seguir.

```
{
    "Sinks": [{
        "Id": "myCloudWatchLogsSink",
        "SinkType": "CloudWatchLogs",
        "CredentialRef": "ssmcred",
        "Region": "us-west-2",
        "LogGroup": "myLogGroup",
        "LogStream": "myLogStream"
    }],
    "Credentials": [{
        "Id": "ssmcred",
        "CredentialType": "ProfileRefreshingAWSCredentialProvider",
        "Profile": "default",
        "FilePath": "%USERPROFILE%//.aws//credentials",
        "RefreshingInterval": 300
    }]
}
```

Uma definição de credencial consiste nos seguintes atributos como pares de chave/valor.

`Id`

Define a string que as definições de coletor podem especificar usando`CredentialRef`para fazer referência a essa configuração de credencial.

`CredentialType`

Defina como a string literal`ProfileRefreshingAWSCredentialProvider`.

`Profile`

Optional. O padrão é `default`.

`FilePath`

Optional. Especifica o caminho para o arquivo de credenciais da AWS. Se omitido, `%USERPROFILE%/.aws/credentials` será o padrão.

`RefreshingInterval`

Optional. A frequência na qual as credenciais são atualizadas, em segundos. Se omitido, `300` será o padrão.

Configuração de decorações de coletor

As declarações de coletor podem incluir pares de chave-valor que especificam dados adicionais para fazer streaming para vários serviços da AWS a fim de aprimorar os registros reunidas da origem.

`TextDecoration`

Use esse par chave-valor quando nenhum `Format` estiver especificado na declaração de coletor. O valor é uma string de formato especial em que ocorre a substituição de variáveis. Por exemplo, suponha que uma `TextDecoration` de `"{ComputerName}:::{timestamp:yyyy-MM-dd HH:mm:ss}:::{_record}"` seja fornecida para um coletor. Quando uma origem emite um registro de log com o texto `The system has resumed from sleep.` e essa origem está conectada ao coletor por um pipe, é feito streaming do texto `MyComputer1:::2017-10-26 06:14:22:::The system has resumed from sleep.` para o serviço da AWS associado ao tipo de coletor. A variável `{_record}` faz referência ao registro de texto original entregue pela origem.

`ObjectDecoration`

Use esse par de chave-valor quando `Format` está especificado na declaração de coletor para adicionar dados extras antes da serialização de registros. Por exemplo, suponha que uma `ObjectDecoration` de `"ComputerName={ComputerName};DT={timestamp:yyyy-MM-dd HH:mm:ss}"` seja fornecida para um coletor que especifique JSON `Format`. O JSON resultante do qual foi feito streaming para o serviço da AWS associado ao tipo de coletor inclui os seguintes pares de chave-valor, além dos dados originais da origem:

```
{
    ComputerName: "MyComputer2",
    DT: "2017-10-17 21:09:04"
}
```

Para ver um exemplo de uso `ObjectDecoration`, consulte Tutorial: Transmitir arquivos de log JSON para o Amazon S3 usando o Kinesis Agent para Windows.

`ObjectDecorationEx`

Especifica uma expressão, que permite extração e formatação de dados mais flexíveis em comparação com`ObjectDecoration`. Este campo pode ser usado quando o formato do coletor é`json`. A sintaxe da expressão é mostrada a seguir.

```
"ObjectDecorationEx":
 "attribute1={expression1};attribute2={expression2};attribute3={expression3}(;...)"
```

Por exemplo, o seguinte`ObjectDecorationEx`Attribute:

```
"ObjectDecorationEx":
 "host={env:ComputerName};message={upper(_record)};time={format(_timestamp,
 'yyyyMMdd')}"
```

Transforma o registro literal:

```
System log message
```

Em um objeto JSON da seguinte forma, com os valores retornados pelas expressões:

```
{
    "host": "EC2AMAZ-1234",
    "message": "SYSTEM LOG MESSAGE",
    "time": "20210201"
}
```

Para obter mais informações sobre como formular expressões, consulteDicas para escrever expressões. A maioria dos`ObjectDecoration`deve funcionar usando a nova sintaxe com exceção das variáveis de carimbo de data/hora. A`{timestamp:yyyyMMdd}`em`ObjectDecoration`é expressa como`{format(_timestamp,'yyyyMMdd')}`em`ObjectDecorationEx`.

`TextDecorationEx`

Especifica uma expressão, que permite extração e formatação de dados mais flexíveis em comparação com`TextDecoration`, conforme mostrado no exemplo a seguir.

```
"TextDecorationEx": "Message '{lower(_record)}' at {format(_timestamp, 'yyyy-MM-
dd')}"
```

Você pode usar o`TextDecorationEx`para compor objetos JSON. Use '@ {'para escapar a chaves abertas, conforme mostrado no exemplo a seguir.

```
"TextDecorationEx": "@{ \"var\": \"{upper($myvar1)}\" }"
```

Se o tipo da origem conectada ao coletor for `DirectorySource`, o coletor poderá usar três variáveis adicionais:

`_FilePath`

O caminho completo para o arquivo de log.

`_FileName`

O nome e a extensão do nome do arquivo.

`_Position`

Um valor inteiro que representa onde o registro está localizado no arquivo de log.

Essas variáveis são úteis quando você usa uma origem que reúne os registros de log de vários arquivos conectados a um coletor que faz streaming de todos os registros para um único stream. A injeção dos valores dessas variáveis nos registros de streaming permite que a análise de downstream no pipeline de dados ordene os registros por arquivo e por local em cada arquivo.

Dicas para escrever expressões

Uma expressão pode ser qualquer uma destas:

- Uma expressão variável.
- Uma expressão constante, por exemplo,`'hello'`,`1`,`1.21`,`null`,`true`,`false`.
- Uma expressão de invocação que chama uma função, conforme mostrado no exemplo a seguir.

```
regexp_extract('Info: MID 118667291 ICID 197973259 RID 0 To: <jd@acme.com>', 'To: (\\
\\S+)', 1)
```

Caracteres especiais

Duas barras invertidas são necessárias para escapar caracteres especiais.

Nesting

As invocações de função podem ser aninhadas, conforme mostrado no exemplo a seguir.

```
format(date(2018, 11, 28), 'MMddyyyy')
```

Variables

Existem três tipos de variáveis: local, meta e global.

- Variáveis locaisComece com um$tais como`$message`. Eles são usados para resolver a propriedade do objeto de evento, uma entrada se o evento for um dicionário, ou um atributo se o evento for um objeto JSON. Se a variável local contiver espaços ou caracteres especiais, use uma variável local entre aspas, como`$'date created'`.
- Variáveis metaComece com um sublinhado (_) e são usados para resolver os metadados do evento. Todos os tipos de eventos suportam as seguintes variáveis meta.

  `_timestamp`

  O time stamp do evento.

  `_record`

  A representação de texto bruto do evento.

  Os eventos de log suportam as seguintes meta variáveis adicionais.

  `_filepath`

  `_filename`

  `_position`

  `_linenumber`

- Variáveis globaisresolver para variáveis de ambiente, metadados de instância do EC2 ou EC2tag. Para obter o melhor desempenho, recomendamos usar o prefixo para limitar o escopo da pesquisa, como`{env:ComputerName}`,`{ec2:InstanceId}`, e`{ec2tag:Name}`.

Funções incorporadas

O Kinesis Agent para Windows oferece suporte às seguintes funções incorporadas do. Se algum dos argumentos forNULLe a função não é projetada para lidar comNULL, umNULLobjeto é retornado.

```
//string functions
int length(string input)
string lower(string input)
string lpad(string input, int size, string padstring)
string ltrim(string input)
string rpad(string input, int size, string padstring)
string rtrim(string input)
string substr(string input, int start)
string substr(string input, int start, int length)
string trim(string input)
string upper(string str)

//regular expression functions
string regexp_extract(string input, string pattern)
string regexp_extract(string input, string pattern, int group)

//date functions
DateTime date(int year, int month, int day)
DateTime date(int year, int month, int day, int hour, int minute, int second)
DateTime date(int year, int month, int day, int hour, int minute, int second, int
 millisecond)

//conversion functions
int? parse_int(string input)
decimal? parse_decimal(string input)
DateTime? parse_date(string input, string format)
string format(object o, string format)

//coalesce functions
object coalesce(object obj1, object obj2)
object coalesce(object obj1, object obj2, object obj3)
object coalesce(object obj1, object obj2, object obj3, object obj4)
object coalesce(object obj1, object obj2, object obj3, object obj4, object obj5)
object coalesce(object obj1, object obj2, object obj3, object obj4, object obj5, object
 obj6)
```

Configuração de substituições de variáveis de coletor

As declarações de coletor `KinesisStream`, `KinesisFirehose` e `CloudWatchLogs` exigem um par de chave-valor `LogStream` ou `StreamName`. Os valores desses pares de chaves-valores podem conter referências a variáveis que são automaticamente resolvidas pelo Kinesis Agent para Windows. para o`CloudWatchLogs`, o`LogGroup`Também é necessário e pode conter referências a variáveis que são automaticamente resolvidas pelo Kinesis Agent para Windows. As variáveis são especificadas usando o modelo {*`prefix`*`:`*`variablename`*`}` em que *`prefix`*`:` é opcional. Os prefixos compatíveis são os seguintes:

- `env`— A referência a variáveis é resolvida para o valor da variável de ambiente com o mesmo nome.
- `ec2`— A referência a variáveis é resolvida para os metadados da instância do EC2 com o mesmo nome.
- `ec2tag`— A referência a variáveis é resolvida para o valor da tag de instância do EC2 com o mesmo nome. A permissão `ec2:Describe*` é necessária para acessar as tags de instância. Para obter mais informações, consulte Permissões necessárias para a expansão de variáveis de tags do EC2.

Se o prefixo não for especificado, se houver uma variável de ambiente com o mesmo nome que `variablename`, a referência a variáveis será resolvida para o valor da variável de ambiente. Caso contrário, se `variablename` for `instance_id` ou `hostname`, a referência a variáveis será resolvida para o valor dos metadados do EC2 com o mesmo nome. Caso contrário, a referência a variáveis não será resolvida.

Veja a seguir exemplos de pares de chave-valor válidos usando referências a variáveis:

```
"LogStream": "LogStream_{instance_id}"
"LogStream": "LogStream_{hostname}"
"LogStream": "LogStream_{ec2:local-hostname}"
"LogStream": "LogStream_{computername}"
"LogStream": "LogStream_{env:computername}"
```

As declarações de coletor do `CloudWatchLogs` oferecem suporte a uma variável de timestamp de formato especial que permite que o timestamp do registro de evento ou log original da origem altere o nome do stream de logs. O formato é `{timestamp:timeformat}`. Veja o exemplo a seguir:

```
"LogStream": "LogStream_{timestamp:yyyyMMdd}"
```

Se o registro de evento ou log foi gerado em 5 de junho de 2017, o valor do par de chave-valor `LogStream` no exemplo anterior seria resolvido para `"LogStream_20170605"`.

Se autorizado, o tipo de coletor `CloudWatchLogs` pode criar automaticamente novos streams de log quando necessários com base nos nomes gerados. Você não pode fazer isso para outros tipos de coletor, pois eles exigem configuração adicional além do nome do stream.

Há substituições de variáveis especiais que ocorrem na decoração de objeto e texto. Para obter mais informações, consulte Configuração de decorações de coletor.

Configuração do enfileiramento do coletor

As declarações dos coletores `KinesisStream`, `KinesisFirehose` e `CloudWatchLogs` também poderão ativar o enfileiramento de registros cujo streaming para o serviço da AWS associado falhou com esses tipos de coletor devido a problemas de conectividade temporários. Para ativar o enfileiramento e novas tentativas de streaming automático quando a conectividade for restaurada, use os seguintes pares de chave-valor nas declarações de coletor:

`QueueType`

Especifica o tipo de mecanismo de enfileiramento a ser usado. O único valor com suporte é `file`, que indica que os registros devem ser colocados na fila em um arquivo. Esse par de chave-valor é obrigatório para ativar o recurso de enfileiramento do Kinesis Agent para Windows. Se não for especificado, o comportamento padrão será apenas colocar na fila na memória e não efetuar o streaming quando os limites de enfileiramento na memória forem atingidos.

`QueuePath`

Especifica o caminho para a pasta que contém os arquivos de registros em fila. Esse par de chave/valor é opcional. O valor padrão é `%PROGRAMDATA%\KinesisTap\Queue\SinkId` em que SinkId é o identificador atribuído como o valor do `Id` para a declaração de coletor.

`QueueMaxBatches`

Limita a quantidade total de espaço que o Kinesis Agent para Windows pode consumir ao colocar registros na fila para streaming. A quantidade de espaço é limitada ao valor desse par de chave/valor multiplicado pelo número máximo de bytes por lote. O número máximo de bytes por lote para os tipos de coletor `KinesisStream`, `KinesisFirehose` e `CloudWatchLogs` são 5 MB, 4

MB e 1 MB, respectivamente. Quando esse limite for atingido, nenhuma das falhas de streaming será colocada em fila e as falhas serão relatadas como não recuperáveis. Esse par de chave/valor é opcional. O valor padrão é 10.000 lotes.

Configuração de um proxy para coletores

Para configurar um proxy para todos os tipos de coletor do Kinesis Agent para Windows que acessam os serviços da AWS, edite o arquivo de configuração do Kinesis Agent para Windows localizado em`%Program Files%\Amazon\KinesisTap\AWSKinesisTap.exe.config`. Para obter instruções, consulte o`proxy`seção emReferência de arquivos de configuração do AWS SDK for .NETnoGuia do desenvolvedor do AWS SDK para .NET.

Configurando variáveis de resolução em mais atributos de coletor

O exemplo a seguir mostra uma configuração de coletor do que usa o`Region`Variável de ambiente para o valor do`Region`Attribute chave-valor. para o`RoleARN`, ele especifica a chave de tag EC2`MyRoleARN`, que é avaliado para o valor associado a essa chave.

```
"Id": "myCloudWatchLogsSink",
"SinkType": "CloudWatchLogs",
"LogGroup": "EC2Logs",
"LogStream": "logs-{instance_id}"
"Region": "{env:Region}"
"RoleARN": "{ec2tag:MyRoleARN}"
```

Configurando endpoints regionais do AWS STS ao usar a propriedade RoleARN nos pias da AWS

Esse recurso só se aplicará se você estiver usando o KinesISTAP no Amazon EC2 e usando o`RoleARN`dos coletores da AWS para assumir uma função externa do IAM para autenticar com os serviços da AWS de destino.

Definindo`UseSTSRegionalEndpoints`para`true`, você pode especificar que um agente use o endpoint regional (por exemplo,`https://sts.us-east-1.amazonaws.com`) em vez do endpoint global (por exemplo,`https://sts.amazonaws.com`). O uso de um endpoint STS Regional reduz a latência de ida e volta para a operação e limita o impacto de falhas no serviço de endpoint global.

Configurando o ponto final da VPC para pias da AWS

Você pode especificar um endpoint da VPC na configuração do coletor para`CloudWatchLogs`,`CloudWatch`,`KinesisStreams`, e`KinesisFirehose`tipos de pia. Um VPC endpoint permite que você conecte de forma privada a VPC aos serviços da AWS compatíveis e aos serviços do VPC endpoint desenvolvidos pelo AWS PrivateLink sem exigir um gateway da Internet, um dispositivo NAT, uma conexão VPN ou uma conexão do AWS Direct Connect. As instâncias na sua VPC não exigem que endereços IP públicos se comuniquem com recursos no serviço. O tráfego entre a sua VPC e os outros serviços não deixa a rede da Amazon. Para obter mais informações, consulteVPC endpointsnoGuia do usuário da Amazon VPC.

Você especifica o endpoint da VPC usando o parâmetro`ServiceURL`, conforme mostrado no exemplo a seguir de um`CloudWatchLogs`configuração do coletor. Defina o valor de`ServiceURL`para o valor mostrado na guiaDetalhes do VPC endpointusando o console da Amazon VPC.

```
{
    "Id": "myCloudWatchLogsSink",
    "SinkType": "CloudWatchLogs",
    "LogGroup": "EC2Logs",
    "LogStream": "logs-{instance_id}",
    "ServiceURL":"https://vpce-ab1c234de56-ab7cdefg.logs.us-east-1.vpce.amazonaws.com"
}
```

Configurando um meio alternativo de proxy

Esse recurso permite que você configure um servidor proxy em uma configuração de coletor usando o suporte de proxy integrado ao AWS SDK em vez do .NET. Anteriormente, a única maneira de configurar o agente para usar um proxy era usar um recurso nativo do .NET, que roteava automaticamente todas as solicitações HTTP/S através do proxy definido no arquivo proxy.

Se você estiver usando o agente com um servidor proxy no momento, não será necessário alterar para usar esse método.

Você pode usar o`ProxyHost`e`ProxyPort`Para configurar um proxy alternativo, conforme mostrado no exemplo a seguir.

```
{
    "Id": "myCloudWatchLogsSink",
    "SinkType": "CloudWatchLogs",
```

```
    "LogGroup": "EC2Logs",
    "LogStream": "logs-{instance_id}",
    "Region": "us-west-2",
    "ProxyHost": "myproxy.mydnsdomain.com",
    "ProxyPort": "8080"
}
```

Declarações de pipe

Usar oDeclarações de pipepara conectar uma fonte (consulteDeclarações de origem) para um lavatório (verDeclarações de coletor) no Amazon Kinesis Agent para Microsoft Windows. Uma declaração de pipe é expressa como um objeto JSON. Depois que o Kinesis Agent para Windows é iniciado, os logs, os eventos ou as métricas são coletados da origem para um pipe. É feito streaming deles para vários serviços da AWS usando o coletor associado a esse pipe.

Veja a seguir um exemplo de declaração de pipe :

```
{
   "Id": "MyAppLogToCloudWatchLogs",
   "SourceRef": "MyAppLog",
   "SinkRef": "MyCloudWatchLogsSink"
}
```

Tópicos

- Configuração de pipes
- Configuração do Kinesis Agent para Pipes Métricos do Windows

Configuração de pipes

Todas as declarações de pipe podem conter os seguintes pares de chave/valor:

`Id`

Especifica o nome do pipe (obrigatório). Deve ser exclusivo no arquivo de configuração.

`Type`

Especifica o tipo de transformação (se houver) que é aplicada pelo pipe quando os dados de log são transferidos da origem para o coletor. O único valor suportado é `RegexFilterPipe`. Esse valor habilita a filtragem de expressões regulares da representação textual subjacente do registro

de log. O uso da filtragem pode reduzir os custos de armazenamento e transmissão enviando somente os registros de log relevantes downstream para o pipeline de dados. Esse par de chave/valor é opcional. O valor padrão é não fornecer nenhuma transformação.

`FilterPattern`

Especifica a expressão regular para pipelines `RegexFilterPipe` que são usados para filtrar os registros de log coletados pela origem antes de serem transferidos para o coletor. Os registros de log são transferidos por pipes do tipo `RegexFilterPipe` quando a expressão regular corresponde à representação textual subjacente do registro. Os registros de log estruturados que são gerados, por exemplo, ao usar o par de chave-valor `ExtractionPattern` em uma declaração `DirectorySource`, ainda podem ser filtrados usando o mecanismo `RegexFilterPipe`. Isso ocorre porque esse mecanismo opera na representação textual original antes da análise. Esse par de chave-valor é opcional, mas deverá ser fornecido se o pipe especificar o tipo `RegexFilterPipe`.

Veja a seguir um exemplo de declaração de pipe `RegexFilterPipe`:

```
{
 "Id": "MyAppLog2ToFirehose",
 "Type": "RegexFilterPipe",
 "SourceRef": "MyAppLog2",
 "SinkRef": "MyFirehose",
 "FilterPattern": "^(10|11),.*",
 "IgnoreCase": false,
 "Negate": false
}
```

`SourceRef`

Especifica o nome (o valor do par de chave-valor `Id`) da declaração de origem que define a origem que está coletando dados de log, eventos e métricas para o pipe (obrigatório).

`SinkRef`

Especifica o nome (o valor do par de chave-valor `Id`) da declaração do coletor que define o coletor que está recebendo os dados de log, eventos e métricas para o pipe (obrigatório).

`IgnoreCase`

Optional. Aceita valores de`true`ou`false`. Quando definido como`true`, o Regex corresponderá aos registros de forma insensível a maiúsculas e minúsculas.

`Negate`

Optional. Aceita valores de`true`ou`false`. Quando definido como`true`, o pipe encaminhará os registros queNãoA expressão regular.

Para obter um exemplo de um arquivo de configuração completo que usa o tipo de pipe `RegexFilterPipe`, consulte Uso de pipes.

Configuração do Kinesis Agent para Pipes Métricos do Windows

Há uma origem de métrica integrada chamada`_KinesisTapMetricsSource`que produz métricas sobre o Kinesis Agent para Windows. Se houver um`CloudWatch`declaração de dissipador com um`IddeMyCloudWatchSink`O exemplo de declaração de pipeline a seguir transferirá o Kinesis Agent para as métricas geradas pelo Windows para esse coletor:

```
{
   "Id": "KinesisAgentMetricsToCloudWatch",
   "SourceRef": "_KinesisTapMetricsSource",
   "SinkRef": "MyCloudWatchSink"
}
```

Para obter mais informações sobre a origem de métricas incorporadas do Kinesis Agent para Windows, consulteOrigem de métricas incorporadas do Kinesis Agent para Windows.

Se o arquivo de configuração também fizer streaming das métricas de contador de desempenho do Windows, recomendamos que você use um pipe e um coletor separados em vez de usar o mesmo coletor para as métricas do Kinesis Agent para as métricas do Windows e as métricas de contador de desempenho do Windows.

Configuração de atualizações automáticas

Usar `aappsettings.json`Para habilitar a atualização automática do Amazon Kinesis Agent para Microsoft Windows e do arquivo de configuração do Kinesis Agent para Windows. Para controlar o comportamento de atualização, especifique o par de chave/valor `Plugins` no mesmo nível no arquivo de configuração que `Sources`, `Sinks` e `Pipes`.

O par de chave/valor `Plugins` especifica a funcionalidade geral adicional a ser usada que não se insere especificamente nas categorias de origens, coletores e pipes. Por exemplo, há um plug-in para atualizar o Kinesis Agent para Windows e um plug-in para atualizar

`oappsettings.json`Arquivo de configuração. Os plug-ins são representados como objetos JSON e sempre têm um par de chave/valor `Type`. O `Type` determina quais outros pares de chave/valor podem ser especificados para o plug-in. Há suporte para os seguintes tipos de plug-in:

`PackageUpdate`

Especifica que o Kinesis Agent para Windows deve verificar periodicamente um arquivo de configuração da versão do pacote. Se o arquivo de versão do pacote indicar que uma versão diferente do Kinesis Agent para Windows deve ser instalada, o Kinesis Agent para Windows fará download dessa versão e a instalará. Os pares de chave/valor do plug-in `PackageUpdate` incluem:

`Type`

O valor deve ser a string `PackageUpdate`, e ele é obrigatório.

`Interval`

Especifica com que frequência é verificada a existência de alterações em minutos no arquivo de versão do pacote representadas como uma string. Esse par de chave/valor é opcional. Se não for especificado, o valor padrão será 60 minutos. Se o valor for inferior a 1, não ocorrerá nenhuma verificação de atualização.

`PackageVersion`

Especifica o local do arquivo JSON de versão do pacote. O arquivo pode residir em um compartilhamento de arquivos (`file://`), um site (`http://`) ou Amazon S3 (`s3://`). Por exemplo, um valor de`s3://mycompany/config/agent-package-version.json`indica que o Kinesis Agent para Windows deve verificar o conteúdo do`config/agent-package-version.json`arquivo`mycompany`Bucket do Amazon S3. Ele deve realizar atualizações com base no conteúdo desse arquivo.

> **Note**
>
> O valor da propriedade`PackageVersion`O par de chave/valor faz distinção de maiúsculas Amazon S3 minúsculas para

Veja a seguir um exemplo do conteúdo de um arquivo de versão do pacote:

```
{
```

```
    "Name": "AWSKinesisTap",
    "Version": "1.0.0.106",
    "PackageUrl": "https://s3-us-west-2.amazonaws.com/kinesis-agent-windows/
downloads/AWSKinesisTap.{Version}.nupkg"
}
```

O`Version`Especifica qual versão do Kinesis Agent para Windows deve ser instalada se ainda não estiver instalada. A referência de variável `{Version}` no `PackageUrl` resolve o valor que você especificar para o par de chave/valor `Version`. Neste exemplo, a variável é resolvida para a string `1.0.0.106`. Essa resolução de variável é fornecida para que possa haver um único lugar no arquivo de versão do pacote no qual a versão desejada específica é armazenada. Você pode usar vários arquivos de versão do pacote para controlar o ritmo da implementação de novas versões do Kinesis Agent para Windows para validar uma nova versão antes de uma implantação maior. Para reverter uma implantação do Kinesis Agent para Windows, altere um ou mais arquivos de versão do pacote para especificar uma versão anterior do Kinesis Agent para Windows que funcione em seu ambiente.

O valor do par de chave/valor `PackageVersion` é afetado pela substituição de variáveis para facilitar a seleção automática de diferentes arquivos de versão do pacote. Para obter mais informações sobre substituição de variáveis, consulte Configuração de substituições de variáveis de coletor.

`AccessKey`

Especifica qual chave de acesso usar ao autenticar o acesso ao arquivo de versão do pacote no Amazon S3. Esse par de chave/valor é opcional. Não é recomendável usar esse par de chave/valor. Para saber as abordagens de autenticação alternativas que são recomendadas, consulte Como configurar a autenticação da.

`SecretKey`

Especifica qual chave secreta usar ao autenticar o acesso ao arquivo de versão do pacote no Amazon S3. Esse par de chave/valor é opcional. Não é recomendável usar esse par de chave/valor. Para saber as abordagens de autenticação alternativas que são recomendadas, consulte Como configurar a autenticação da.

`Region`

Especifica o endpoint de região a ser usado ao acessar o arquivo de versão do pacote do Amazon S3. Esse par de chave/valor é opcional.

`ProfileName`

Especifica qual perfil de segurança usar ao autenticar o acesso ao arquivo de versão do pacote no Amazon S3. Para obter mais informações, consulte Como configurar a autenticação da. Esse par de chave/valor é opcional.

`RoleARN`

Especifica qual função assumir ao autenticar o acesso ao arquivo de versão do pacote no Amazon S3 em um cenário entre contas. Para obter mais informações, consulte Como configurar a autenticação da. Esse par de chave/valor é opcional.

Se nenhum plug-in `PackageUpdate` for especificado, nenhum arquivo de versão do pacote será verificado para determinar se uma atualização é obrigatória.

`ConfigUpdate`

Especifica que o Kinesis Agent para Windows deve verificar periodicamente se há um`appsettings.json`Armazenado em um compartilhamento de arquivos, no site ou no Amazon S3. Se houver um arquivo de configuração atualizado, ele será baixado e instalado pelo Kinesis Agent para Windows.`ConfigUpdate`Os pares de chave/valor incluem o seguinte:

`Type`

O valor deve ser a string `ConfigUpdate`, e ele é obrigatório.

`Interval`

Especifica com que frequência é verificada a existência de um novo arquivo de configuração representado como uma string. Esse par de chave/valor é opcional e, se não for especificado, o padrão será 5 minutos. Se o valor for inferior a 1, a atualização do arquivo de configuração não será verificada.

`Source`

Especifica onde procurar um arquivo de configuração atualizado. O arquivo pode residir em um compartilhamento de arquivos (`file://`), um site (`http://`) ou Amazon S3 (`s3://`). Por exemplo, um valor des`3://mycompany/config/appsettings.json`indica que o Kinesis Agent para Windows deve verificar se há atualizações para o`config/appsettings.json`arquivo`mycompany`Bucket do Amazon S3.

> **Note**
>
> O valor da propriedade`Source`O par de chave-valor faz distinção de maiúsculas e minúsculas no Amazon S3.

O valor do par de chave/valor `Source` é afetado pela substituição de variáveis para facilitar a seleção automática de diferentes arquivos de configuração. Para obter mais informações sobre substituição de variáveis, consulte Configuração de substituições de variáveis de coletor.

`Destination`

Especifica onde armazenar o arquivo de configuração no computador local. Pode ser um caminho relativo, um caminho absoluto ou um caminho que contenha referências de variável de ambiente, como `%PROGRAMDATA%`. Se o caminho for relativo, ele será relativo ao local onde o Kinesis Agent para Windows está instalado. Em geral, o valor deve ser `.\appsettings.json`. Esse par de chave/valor é obrigatório.

`AccessKey`

Especifica qual chave de acesso usar ao autenticar o acesso ao arquivo de configuração no Amazon S3. Esse par de chave/valor é opcional. Não é recomendável usar esse par de chave/valor. Para saber as abordagens de autenticação alternativas que são recomendadas, consulte Como configurar a autenticação da.

`SecretKey`

Especifica qual chave secreta usar ao autenticar o acesso ao arquivo de configuração no Amazon S3. Esse par de chave/valor é opcional. Não é recomendável usar esse par de chave/valor. Para saber as abordagens de autenticação alternativas que são recomendadas, consulte Como configurar a autenticação da.

`Region`

Especifica o endpoint de região a ser usado ao acessar o arquivo de configuração do Amazon S3. Esse par de chave/valor é opcional.

`ProfileName`

Especifica qual perfil de segurança usar ao autenticar o acesso ao arquivo de configuração no Amazon S3. Para obter mais informações, consulte Como configurar a autenticação da. Esse par de chave/valor é opcional.

`RoleARN`

Especifica qual função assumir ao autenticar o acesso ao arquivo de configuração no Amazon S3 em um cenário entre contas. Para obter mais informações, consulte Como configurar a autenticação da. Esse par de chave/valor é opcional.

Se nenhum plug-in `ConfigUpdate` for especificado, nenhum arquivo de configuração será verificado para determinar se uma atualização é obrigatória.

Veja a seguir um exemplo de arquivo de configuração `appsettings.json` que demonstra o uso dos plug-ins `PackageUpdate` e `ConfigUpdate`. Neste exemplo, há um arquivo de versão do pacote localizado no`mycompany`Bucket do Amazon S3 nomeado como`config/agent-package-version.json`. É verificada a existência de alterações nesse arquivo a cada 2 horas aproximadamente. Se uma versão diferente do Kinesis Agent para Windows for especificada no arquivo de versão do pacote, a versão especificada do agente será instalada do local especificado no arquivo de versão do pacote.

Além disso, há um`appsettings.json`armazenado no arquivo de configuração`mycompany`Bucket do Amazon S3 nomeado como`config/appsettings.json`. Aproximadamente a cada 30 minutos, esse arquivo é comparado com o arquivo de configuração atual. Se eles forem diferentes, o arquivo de configuração atualizado será baixado do Amazon S3 e instalado no local normal do`appsettings.json`Arquivo de configuração.

```
{
  "Sources": [
    {
      "Id": "ApplicationLogSource",
      "SourceType": "DirectorySource",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "RecordParser": "SingleLine"
    }
  ],
  "Sinks": [
    {
       "Id": "ApplicationLogKinesisFirehoseSink",
       "SinkType": "KinesisFirehose",
       "StreamName": "ApplicationLogFirehoseDeliveryStream",
       "Region": "us-east-1"
    }
```

```
    ],
  "Pipes": [
    {
      "Id": "ApplicationLogSourceToApplicationLogKinesisFirehoseSink",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "ApplicationLogKinesisFirehoseSink"
    }
  ],
  "Plugins": [
    {
      "Type": "PackageUpdate"
      "Interval": "120",
      "PackageVersion": "s3://mycompany/config/agent-package-version.json"
    },
    {
      "Type": "ConfigUpdate",
      "Interval": "30",
      "Source": "s3://mycompany/config/appsettings.json",
      "Destination": ".\appSettings.json"
    }
  ]
}
```

Exemplos de configuração do Kinesis Agent para Windows

O`appsettings.json`é um documento JSON que controla como o Amazon Kinesis Agent para Microsoft Windows coleta logs, eventos e métricas. Ele também controla como o Kinesis Agent para Windows transforma esses dados e faz streaming dele para vários serviços da AWS. Para obter detalhes sobre declarações de origem, coletor e pipe no arquivo de configuração, consulte Declarações de origem, Declarações de coletor e Declarações de pipe.

As seções a seguir contêm exemplos de arquivos de configuração para vários tipos diferentes de cenários.

Tópicos

- Streaming de várias origens para o Kinesis Data Streams
- Streaming do log de eventos de aplicativos do Windows para coletores
- Uso de pipes
- Uso de várias origens e pipes

Streaming de várias origens para o Kinesis Data Streams

O exemplo a seguirappsettings.jsonDemonstram o streaming de logs e eventos de várias origens para o Kinesis Data Streams e de contadores de desempenho do Windows para as métricas do Amazon CloudWatch.

Analisador de registros **`DirectorySource`**, **`SysLog`**

O seguinte arquivo faz streaming de registros de log de formato syslog de todos os arquivos com um`.log`na extensãoC:\LogSource\para o`SyslogKinesisDataStream`Kinesis Data Streams fazem streaming na região us-east-1. Um marcador é estabelecido para garantir que todos os dados dos arquivos de log sejam enviados, mesmo que o agente seja desligado e reiniciado posteriormente. Um aplicativo personalizado pode ler e processar os registros do stream `SyslogKinesisDataStream`.

```
{
  "Sources": [
    {
      "Id": "SyslogDirectorySource",
      "SourceType": "DirectorySource",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "RecordParser": "SysLog",
      "TimeZoneKind": "UTC",
      "InitialPosition": "Bookmark"
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "SyslogKinesisDataStream",
      "Region": "us-east-1"
    }
  ],
  "Pipes": [
    {
      "Id": "SyslogDS2KSSink",
      "SourceRef": "SyslogDirectorySource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
```

```
}
```

Analisador de registros **`DirectorySource`**, **`SingleLineJson`**

O seguinte arquivo faz streaming de registros de log de formato JSON de todos os arquivos com um `.log` na extensão `C:\LogSource\` para o `JsonKinesisDataStream` Kinesis Data Streams fazem streaming na região us-east-1. Antes do streaming, pares de chave-valor para as chaves `DT` e `ComputerName` são adicionados a cada objeto JSON, com valores para o nome do computador e a data e a hora em que o registro é processado. Um aplicativo personalizado pode ler e processar os registros do stream `JsonKinesisDataStream`.

```
{
  "Sources": [
    {
      "Id": "JsonLogSource",
      "SourceType": "DirectorySource",
      "RecordParser": "SingleLineJson",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "InitialPosition": 0
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "JsonKinesisDataStream",
      "Region": "us-east-1",
      "Format": "json",
      "ObjectDecoration": "ComputerName={ComputerName};DT={timestamp:yyyy-MM-dd
 HH:mm:ss}"
    }
  ],
  "Pipes": [
    {
      "Id": "JsonLogSourceToKinesisStreamSink",
      "SourceRef": "JsonLogSource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
}
```

ExchangeLogSource

O seguinte arquivo faz streaming de registros de log gerados pelo Microsoft Exchange e armazenados em arquivos com o `.log`na extensão`C:\temp\ExchangeLog\`para o`ExchangeKinesisDataStream`Stream de dados do Kinesis na região us-east-1 no formato JSON. Embora os logs do Exchange não estejam no formato JSON, o Kinesis Agent para Windows pode analisar os logs e transformá-los em JSON. Antes do streaming, pares de chave-valor para as chaves `DT` e `ComputerName` são adicionados a cada objeto JSON que contém valores para o nome do computador e a data e a hora em que o registro é processado. Um aplicativo personalizado pode ler e processar os registros do stream `ExchangeKinesisDataStream`.

```
{
  "Sources": [
    {
       "Id": "ExchangeSource",
       "SourceType": "ExchangeLogSource",
       "Directory": "C:\\temp\\ExchangeLog\",
       "FileNameFilter": "*.log"
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "ExchangeKinesisDataStream",
      "Region": "us-east-1",
      "Format": "json",
      "ObjectDecoration": "ComputerName={ComputerName};DT={timestamp:yyyy-MM-dd
 HH:mm:ss}"
    }
  ],
  "Pipes": [
    {
      "Id": "ExchangeSourceToKinesisStreamSink",
      "SourceRef": "ExchangeSource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
}
```

`W3SVCLogSource`

O seguinte arquivo faz streaming de serviços de informações da Internet (IIS) para registros de log do Windows armazenados no local padrão para esses arquivos para o`IISKinesisDataStream`Kinesis Data Streams fazem streaming na região us-east-1. Um aplicativo personalizado pode ler e processar os registros do stream `IISKinesisDataStream`. IIS é um servidor web para Windows.

```
{
  "Sources": [
    {
       "Id": "IISLogSource",
       "SourceType": "W3SVCLogSource",
       "Directory": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
       "FileNameFilter": "*.log"
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "IISKinesisDataStream",
      "Region": "us-east-1"
    }
  ],
  "Pipes": [
    {
      "Id": "IISLogSourceToKinesisStreamSink",
      "SourceRef": "IISLogSource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
}
```

`WindowsEventLogSource` com consulta

Os seguintes fluxos de arquivo de log eventos do log de eventos do sistema do Windows que têm um nível de`Critical`ou`Error`(menores ou iguais a 2) para o`SystemKinesisDataStream`Stream de dados do Kinesis na região us-east-1 no formato JSON. Um aplicativo personalizado pode ler e processar os registros do stream `SystemKinesisDataStream`.

```
{
```

```
  "Sources": [
    {
         "Id": "SystemLogSource",
         "SourceType": "WindowsEventLogSource",
         "LogName": "System",
         "Query": "*[System/Level<=2]"
    }
  ],
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "SystemKinesisDataStream",
      "Region": "us-east-1",
      "Format": "json"
    }
  ],
  "Pipes": [
    {
      "Id": "SLSourceToKSSink",
      "SourceRef": "SystemLogSource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
}
```

`WindowsETWEventSource`

O seguinte arquivo faz streaming de eventos de segurança e exceção do Microsoft Common Language Runtime (CLR) para o`ClrKinesisDataStream`Stream de dados do Kinesis na região us-east-1 no formato JSON. Um aplicativo personalizado pode ler e processar os registros do stream `ClrKinesisDataStream`.

```
{
  "Sources": [
    {
       "Id": "ClrETWEventSource",
       "SourceType": "WindowsETWEventSource",
       "ProviderName": "Microsoft-Windows-DotNETRuntime",
       "TraceLevel": "Verbose",
       "MatchAnyKeyword": "0x00008000, 0x00000400"
    }
  ],
```

```
  "Sinks": [
    {
      "Id": "KinesisStreamSink",
      "SinkType": "KinesisStream",
      "StreamName": "ClrKinesisDataStream",
      "Region": "us-east-1",
      "Format": "json"
    }
  ],
  "Pipes": [
    {
      "Id": "ETWSourceToKSSink",
      "SourceRef": "ClrETWEventSource",
      "SinkRef": "KinesisStreamSink"
    }
  ]
}
```

`WindowsPerformanceCounterSource`

O seguinte arquivo faz streaming de contadores de desempenho para o total de arquivos abertos, total de tentativas de login desde a reinicialização, o número de leituras de disco por segundo e a porcentagem de espaço livre em disco para as métricas do CloudWatch na região us-east-1. Você pode representar essas métricas no CloudWatch, criar painéis a partir dos gráficos e definir alarmes que enviam notificações quando os limites são excedidos.

```
{
  "Sources": [
    {
      "Id": "PerformanceCounter",
      "SourceType": "WindowsPerformanceCounterSource",
      "Categories": [
        {
          "Category": "Server",
          "Counters": [
            "Files Open",
            "Logon Total"
          ]
        },
        {
          "Category": "LogicalDisk",
          "Instances": "*",
          "Counters": [
```

```
            "% Free Space",
            {
              "Counter": "Disk Reads/sec",
              "Unit": "Count/Second"
            }
          ]
        }
      ],
    }
  ],
  "Sinks": [
    {
      "Namespace": "MyServiceMetrics",
      "Region": "us-east-1",
      "Id": "CloudWatchSink",
      "SinkType": "CloudWatch"
    }
  ],
  "Pipes": [
    {
      "Id": "PerformanceCounterToCloudWatch",
      "SourceRef": "PerformanceCounter",
      "SinkRef": "CloudWatchSink"
    }
  ]
}
```

Streaming do log de eventos de aplicativos do Windows para coletores

O exemplo a seguir `appsettings.json` Demonstram o streaming de logs de eventos de aplicativos do Windows para vários coletores no Amazon Kinesis Agent para Microsoft Windows. Para obter exemplos de como usar os tipos de coletores `KinesisStream` e `CloudWatch`, consulte Streaming de várias origens para o Kinesis Data Streams.

`KinesisFirehose`

Os seguintes fluxos de arquivos `Critical` ou `Error` Eventos de log de aplicativos do Windows para o `WindowsLogFirehoseDeliveryStream` Stream de entrega do Kinesis Data Firehose na região us-east-1. Se a conectividade com o Kinesis Data Firehose for interrompida, os eventos serão primeiro colocados em fila na memória. Se necessário, eles serão colocados em fila em um arquivo no disco até que a conectividade seja restaurada. Os eventos serão retirados da fila e enviados apos novos eventos.

Você pode configurar o Kinesis Data Firehose para armazenar dados em streaming para vários tipos diferentes de serviços de armazenamento e análise baseados em requisitos de pipeline de dados.

```
{
  "Sources": [
    {
         "Id": "ApplicationLogSource",
         "SourceType": "WindowsEventLogSource",
         "LogName": "Application",
         "Query": "*[System/Level<=2]"
    }
  ],
  "Sinks": [
    {
       "Id": "WindowsLogKinesisFirehoseSink",
       "SinkType": "KinesisFirehose",
       "StreamName": "WindowsLogFirehoseDeliveryStream",
       "Region": "us-east-1",
       "QueueType": "file"
    }
    ],
  "Pipes": [
    {
      "Id": "ALSource2ALKFSink",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "WindowsLogKinesisFirehoseSink"
    }
  ]
}
```

`CloudWatchLogs`

Os seguintes fluxos de arquivos`Critical`ou`Error`Eventos de logs de aplicativos do Windows para o CloudWatch Logs faz streaming de logs do no`MyServiceApplicationLog-Group`. O nome de cada stream começa com `Stream-`. Ele termina com o ano de quatro dígitos, o mês de dois dígitos e o dia de dois dígitos em que o stream foi criado, todos concatenados (por exemplo, `Stream-20180501` é o stream criado em 1º de maio de 2018).

```
{
  "Sources": [
    {
         "Id": "ApplicationLogSource",
```

```
        "SourceType": "WindowsEventLogSource",
        "LogName": "Application",
        "Query": "*[System/Level<=2]"
    }
  ],
  "Sinks": [
    {
      "Id": "CloudWatchLogsSink",
      "SinkType": "CloudWatchLogs",
      "LogGroup": "MyServiceApplicationLog-Group",
      "LogStream": "Stream-{timestamp:yyyyMMdd}",
      "Region": "us-east-1",
      "Format": "json"
    }
  ],
  "Pipes": [
    {
      "Id": "ALSource2CWLSink",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "CloudWatchLogsSink"
    }
  ]
}
```

Uso de pipes

O exemplo de arquivo de configuração `appsettings.json` a seguir demonstra o uso de recursos relacionados ao pipe.

Este exemplo faz streaming de entradas de log doc`:\LogSource\`para o`ApplicationLogFirehoseDeliveryStream`Streaming de entrega do Kinesis Data Firehose. Ele inclui apenas as linhas que correspondem à expressão regular especificada pelo par de chave-valor `FilterPattern`. Especificamente, somente as linhas do arquivo de log que começam com o`10`ou`11`são transmitidos para o Kinesis Data Firehose.

```
{
  "Sources": [
    {
      "Id": "ApplicationLogSource",
      "SourceType": "DirectorySource",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
```

```
      "RecordParser": "SingleLine"
    }
  ],
  "Sinks": [
    {
       "Id": "ApplicationLogKinesisFirehoseSink",
       "SinkType": "KinesisFirehose",
       "StreamName": "ApplicationLogFirehoseDeliveryStream",
       "Region": "us-east-1"
    }
    ],
  "Pipes": [
    {
      "Id": "ALSourceToALKFSink",
      "Type": "RegexFilterPipe",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "ApplicationLogKinesisFirehoseSink",
      "FilterPattern": "^(10|11),.*"
    }
  ]
}
```

Uso de várias origens e pipes

O exemplo de arquivo de configuração `appsettings.json` a seguir demonstra o uso de várias origens e pipes.

Este exemplo faz streaming do aplicativo, da segurança e do sistema do Windows Event Logs para o`EventLogStream`Stream de entrega do Kinesis Data Firehose usando três origens, três pipes e um único coletor.

```
{
    "Sources": [
  {
    "Id": "ApplicationLog",
    "SourceType": "WindowsEventLogSource",
    "LogName": "Application"
  },
  {
    "Id": "SecurityLog",
    "SourceType": "WindowsEventLogSource",
    "LogName": "Security"
```

```
  },
  {
    "Id": "SystemLog",
    "SourceType": "WindowsEventLogSource",
    "LogName": "System"
  }
    ],
    "Sinks": [
  {
    "Id": "EventLogSink",
    "SinkType": "KinesisFirehose",
    "StreamName": "EventLogStream",
    "Format": "json"
  },
    ],
    "Pipes": [
  {
    "Id": "ApplicationLogToFirehose",
    "SourceRef": "ApplicationLog",
    "SinkRef": "EventLogSink"
  },
  {
    "Id": "SecurityLogToFirehose",
    "SourceRef": "SecurityLog",
    "SinkRef": "EventLogSink"
  },
  {
    "Id": "SystemLogToFirehose",
    "SourceRef": "SystemLog",
    "SinkRef": "EventLogSink"
  }
    ]
}
```

Configuração de telemetria

Para permitir o melhor suporte, por padrão, o Amazon Kinesis Agent para Microsoft Windows coleta estatísticas sobre a operação do agente e as envia para a AWS. Essas informações não contêm informações de identificação pessoal. Elas não incluem dados que você coleta ou dos quais faz streaming para os serviços da AWS. Coletamos aproximadamente 1 a 2 KB desses dados de métrica a cada 60 minutos.

Você pode cancelar a coleta e a transmissão dessas estatísticas. Para fazer isso, adicione o seguinte par de chave/valor ao arquivo de configuração `appsettings.json` no mesmo nível que origens, coletores e pipes:

```
"Telemetrics":
    { "off": "true" }
```

Por exemplo, o arquivo de configuração a seguir configura uma origem, um coletor e um pipe e também desativa a telemetria:

```
{
  "Sources": [
    {
      "Id": "ApplicationLogSource",
      "SourceType": "DirectorySource",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "RecordParser": "SingleLine"
    }
  ],
  "Sinks": [
    {
       "Id": "ApplicationLogKinesisFirehoseSink",
       "SinkType": "KinesisFirehose",
       "StreamName": "ApplicationLogFirehoseDeliveryStream",
       "Region": "us-east-1"
    }
    ],
  "Pipes": [
    {
      "Id": "ApplicationLogSourceToApplicationLogKinesisFirehoseSink",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "ApplicationLogKinesisFirehoseSink"
    }
  ],
  "Telemetrics":
    {
      "off": "true"
    }
}
```

Coletamos as seguintes métricas quando a telemetria está ativada:

`ClientId`

O ID exclusivo atribuído automaticamente quando o software é instalado.

`ClientTimestamp`

A data e a hora em que a telemetria é coletada.

`OSDescription`

Uma descrição do sistema operacional.

`DotnetFramework`

A versão da estrutura dotnet atual.

`MemoryUsage`

A quantidade de memória consumida pelo Kinesis Agent para Windows (MB).

`CPUUsage`

A quantidade de porcentagem de uso de CPU do Kinesis Agent para Windows do em valores decimais. Por exemplo, 0,01 significa 1%.

`InstanceId`

O ID da instância do Amazon EC2 se o Kinesis Agent for Windows estiver sendo executado em uma instância do Amazon EC2.

`InstanceType (string)`

O tipo de instância do Amazon EC2 se o Kinesis Agent for Windows estiver sendo executado em uma instância do Amazon EC2.

Além disso, coletamos as métricas listadas em Lista de métricas do agente Kinesis para Windows.

Tutorial: Transmitir arquivos de log JSON para o Amazon S3 usando o Kinesis Agent para Windows

Este tutorial apresenta etapas detalhadas para configurar um pipeline de dados usando o Amazon Kinesis Agent para Microsoft Windows (Kinesis Agent para Windows).

O tutorial inclui as seguintes etapas:

- Usar o Kinesis Agent for Windows para fazer streaming de arquivos de log no formato JSON para oAmazon Simple Storage Service (Amazon S3)ViaAmazon Kinesis Data Firehose. Para obter informações sobre o Kinesis Agent para Windows, consulteO que é o Amazon Kinesis Agent para o Microsoft Windows?.
- Como aprimorar os dados de log antes do streaming usando decoração de objeto. Para obter mais informações, consulte Configuração de decorações de coletor.
- O uso doAmazon AthenaPara procurar determinados tipos de registros de log.

Prerequisites

Se você ainda não tem uma conta da AWS, siga as instruções emConfiguração de uma conta da AWSpara conseguir um.

Tópicos

Etapa 1: Configurar os Serviços da AWS

Siga estas etapas para preparar seu ambiente para streaming de dados de log para o Amazon Simple Storage Service (Amazon S3) usando o Amazon Kinesis Agent para Microsoft Windows. Para obter mais informações e pré-requisitos, consulte Tutorial: Fazer streaming de arquivos de log JSON para o Amazon S3.

Use o AWS Management Console para configurar o AWS Identity and Access Management (IAM), o Amazon S3, o Kinesis Data Firehose e o Amazon Elastic Compute Cloud (Amazon EC2) para se preparar para streaming de dados de log de uma instância do EC2 para o Amazon S3.

Tópicos

Configurar políticas e funções do IAM

Crie a seguinte política, que autoriza o Kinesis Agent para Windows a fazer streaming de registros para um fluxo de entrega específico do Kinesis Data Firehose:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Resource": "arn:aws:firehose:region:account-id:deliverystream/log-
delivery-stream"
        }
    ]
}
```

Substituir*region*Com o nome da região da AWS onde o fluxo de entrega do Kinesis Data Firehose será criado (`us-east-1`, por exemplo). Substitua *`account-id`* pelo ID da conta de 12 dígitos da AWS na qual o fluxo de entrega será criado.

Na barra de navegação, selecioneSuporte parae, em seguida,Support Center. Seu número (ID) de conta de 12 dígitos da conectada no momento aparece noSupport CenterPainel de navegação.

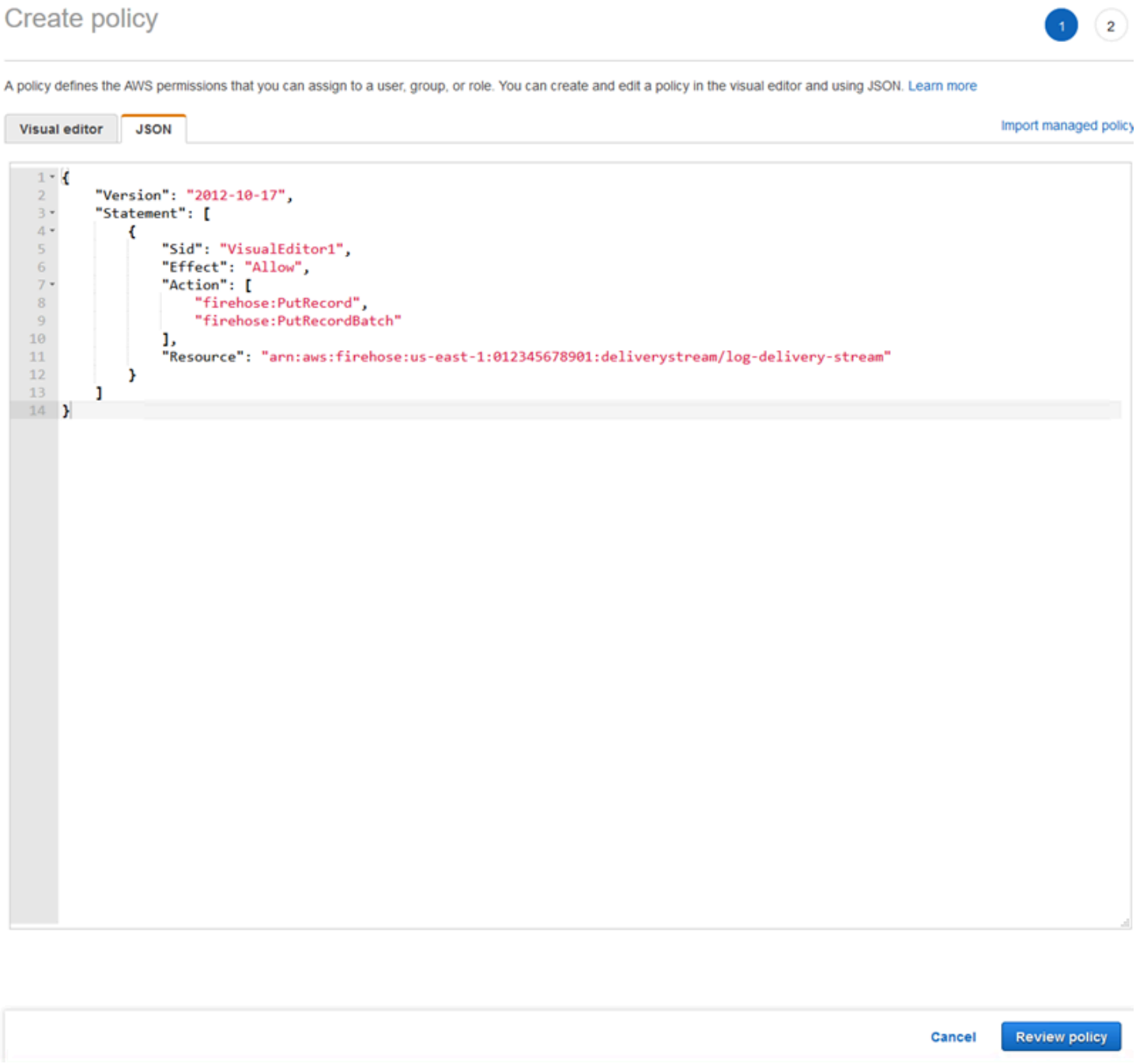
Crie a política usando procedimento a seguir. fornece um nome para a política `log-delivery-stream-access-policy`.

Para criar uma política usando o editor de políticas JSON

1. Faça login no Console de Gerenciamento da AWS e abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação no lado esquerdo, selecione Policies (Políticas).

   Se essa for a primeira vez que escolhe Políticas, a página Bem-vindo às políticas gerenciadas será exibida. Escolha Get Started.
3. Na parte superior da página, escolha Create policy (Criar política).
4. Selecione a guia JSON.
5. Insira um documento de política JSON. Para obter detalhes sobre a linguagem de políticas do IAM, consulteReferência de políticas JSON do InoGuia do usuário do IAM.
6. Ao concluir, selecione Revisar política. O Validador de política indica se há qualquer erro de sintaxe.

   > **Note**
   >
   > Você pode alternar entre as guias Editor visual e JSON sempre que quiser. No entanto, se você fizer alterações ou escolherReview policynoEditor visual, o IAM poderá reestruturar sua política de forma a otimizá-la para o editor visual. Para obter mais informações, consulteReestruturação da políticanoGuia do usuário do IAM.
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.



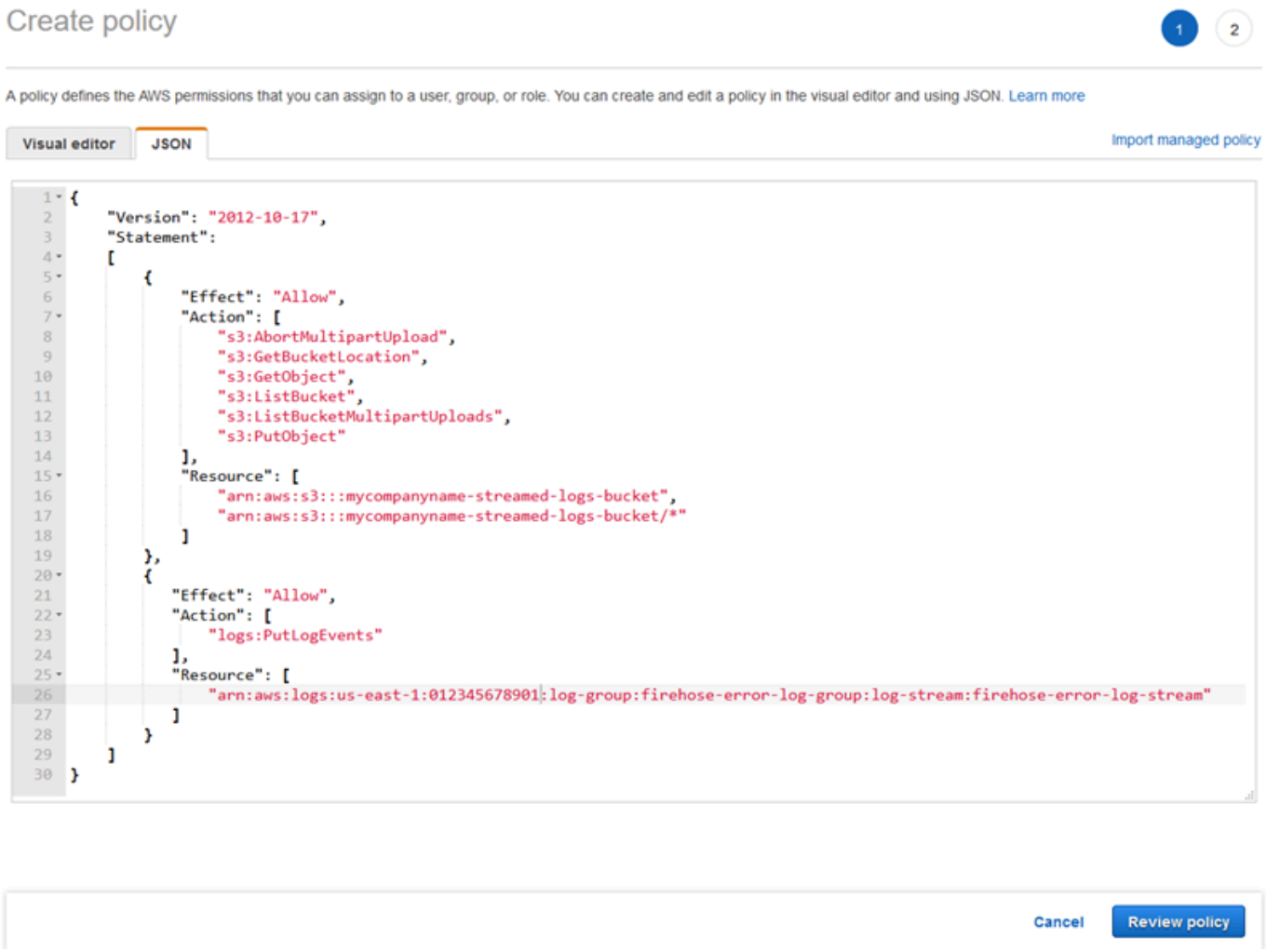
Para criar a função que concede ao Kinesis Data Firehose ao acesso a um bucket do S3

1. Usando o procedimento anterior, crie uma política denominada `firehose-s3-access-policy` que é definida usando o seguinte JSON:

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

```
    "Statement":
    [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*"
            ]
        },
        {
           "Effect": "Allow",
           "Action": [
               "logs:PutLogEvents"
           ],
           "Resource": [
               "arn:aws:logs:region:account-id:log-group:firehose-error-log-
group:log-stream:firehose-error-log-stream"
           ]
        }
    ]
}
```

Substitua *bucket-name* por um único nome de bucket onde os logs serão armazenados. Substituir*region*Com a região da AWS em que o grupo de logs do CloudWatch Logs e o stream de logs serão criados. Eles servem para registrar todos os erros que ocorrerem durante o streaming dos dados para o Amazon S3 via Kinesis Data Firehose. Substitua *account-id* pelo ID da conta de 12 dígitos na qual o grupo de logs e o stream de logs serão criados.



2. No painel de navegação do console do IAM, selecione Roles e, em seguida, Create role.
3. Selecione oServiço da AWSTipo de função do e depois escolha oKinesisserviçoServiço do
4. SelecioneKinesis Data FirehosePara o caso de uso e depois escolhaPróximo: Permissões
5. Na caixa de pesquisa, digite**`firehose-s3-access-policy`**, escolha essa política e depois escolhaPróximo: Review (Revisar).
6. Na caixa Role name (Nome da função), digite **`firehose-s3-access-role`**.
7. Selecione Create role.

Para criar a função a ser associada ao perfil de instância do EC2 que executará o Kinesis Agent para Windows

1. No painel de navegação do console do IAM, selecione Roles e, em seguida, Create role.
2. Selecione oServiço da AWSTipo de função do e depois escolhaEC2.

3. SelecionePróximo: Permissões
4. Na caixa de pesquisa, insira **`log-delivery-stream-access-policy`**.
5. Escolha a política e depoisPróximo: Review (Revisar).
6. Na caixa Role name (Nome da função), digite **`kinesis-agent-instance-role`**.
7. Selecione Create role.

Crie o bucket do Amazon S3

Crie o bucket do S3 em que o Kinesis Data Firehose faz streaming dos logs.

Para criar o bucket do S3 para armazenamento de logs

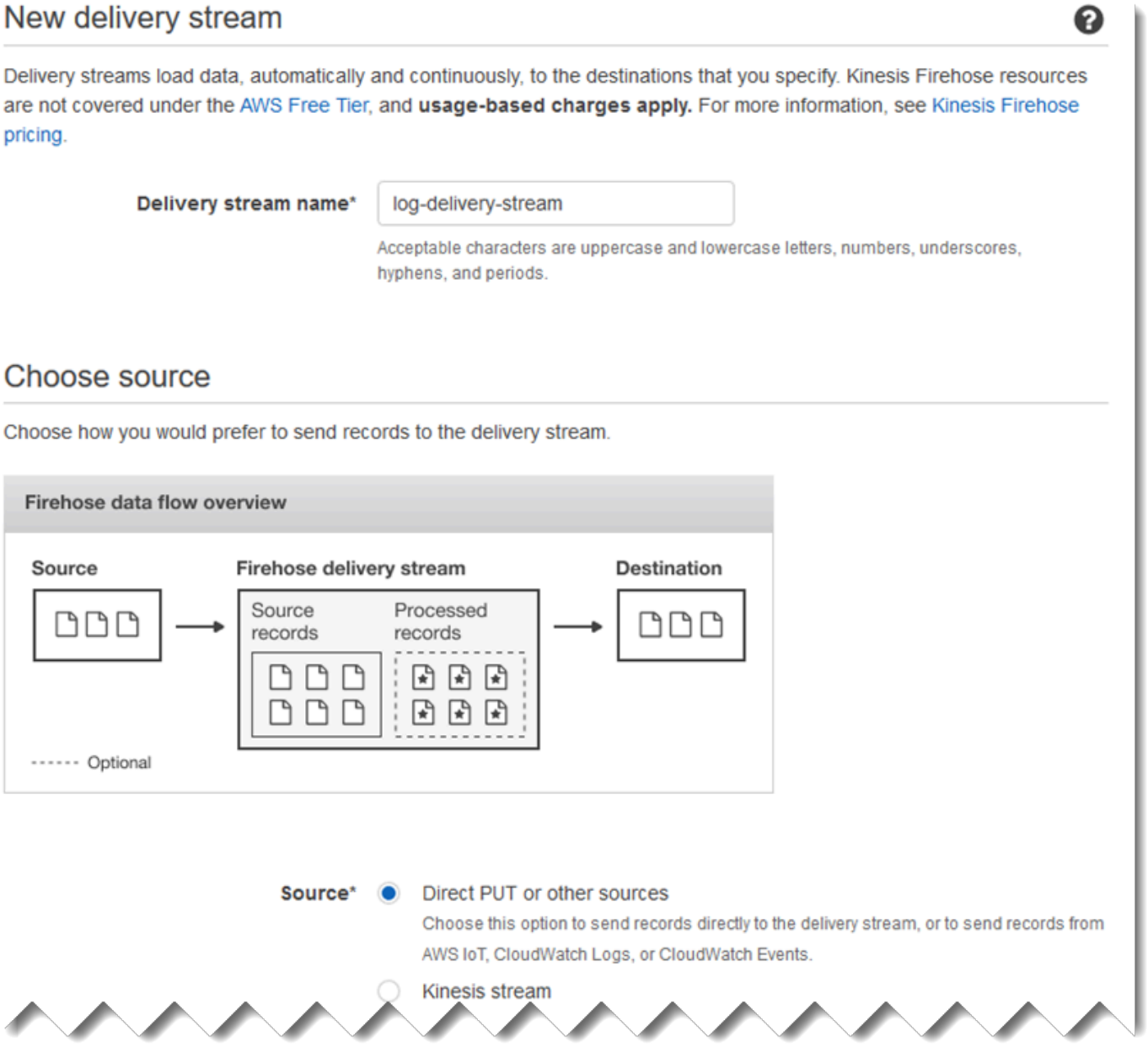
1. Abra o console do Amazon S3 em https://console.aws.amazon.com/s3/.
2. Selecione Create bucket (Criar bucket).
3. Na caixa Bucket name (Nome do bucket), insira o nome do bucket do S3 exclusivo que você escolheu em Configurar políticas e funções do IAM.
4. Escolha a região onde o bucket deve ser criado. Normalmente é a mesma região onde você pretende criar o fluxo de entrega do Kinesis Data Firehose e a instância do Amazon EC2.
5. Escolha Create (Criar).

Criar o fluxo de entrega do Kinesis Data Firehose

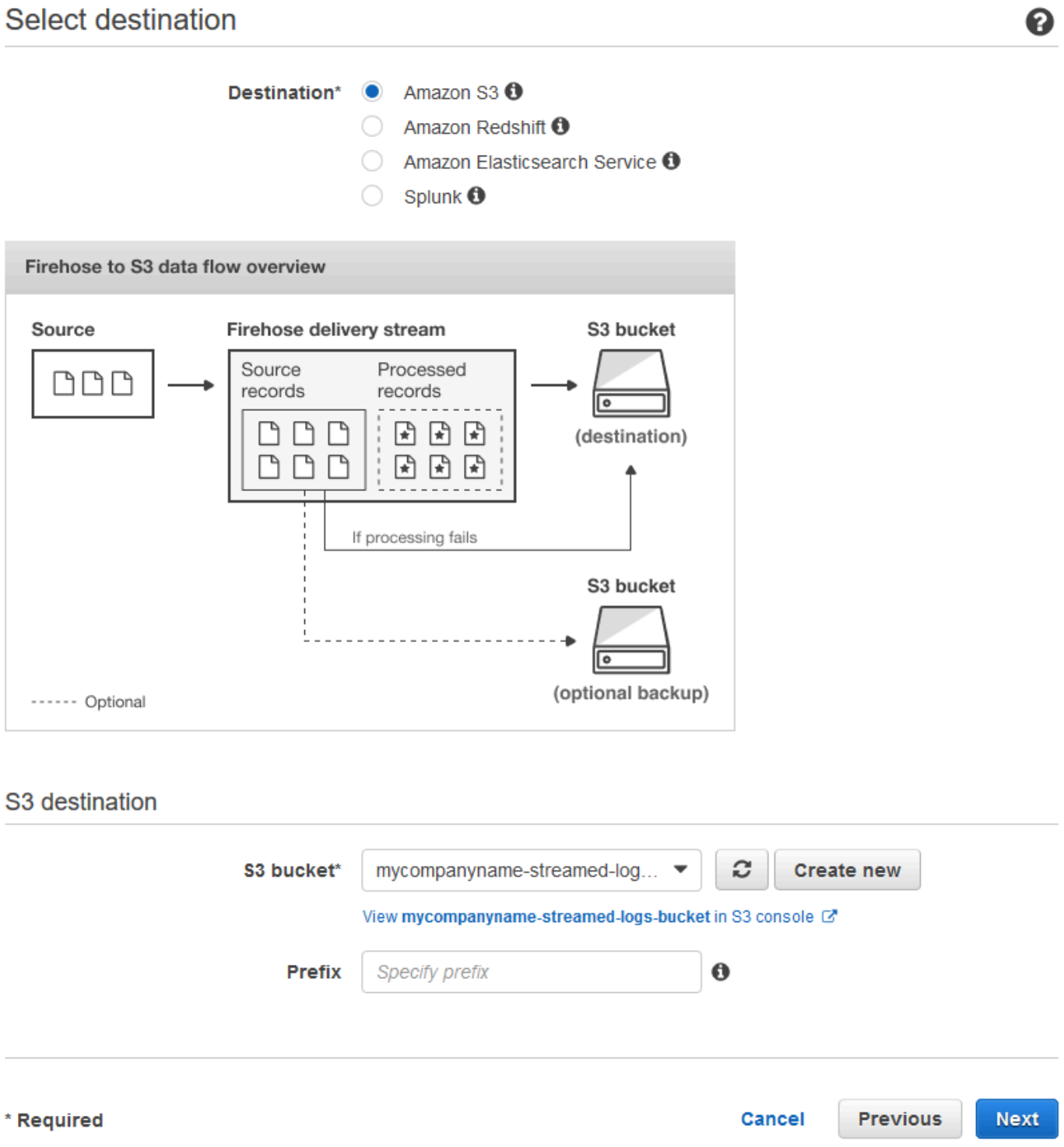
Crie o fluxo de entrega do Kinesis Data Firehose que armazenará registros dos quais foi feito streaming no Amazon S3.

Para criar o fluxo de entrega do Kinesis Data Firehose

1. Abra o console Kinesis Data Firehose emhttps://console.aws.amazon.com/firehose/.
2. Escolha Create Delivery Stream.
3. Na caixa Delivery stream name (Nome do fluxo de entrega), digite **`log-delivery-stream`**.
4. Para a Source (Origem), escolha Direct PUT or other sources (Direct PUT ou outras origens).



5. Escolha Next (Próximo).
6. Escolha Next (Próximo) novamente.
7. Para o destino, escolhaAmazon S3.
8. Para o S3 bucket (Bucket do S3), escolha o nome do bucket criado em Crie o bucket do Amazon S3.



9. Escolha Next (Próximo).
10. Na caixa Buffer interval (intervalo de buffer), digite **`60`**.
11. Em IAM role (Função do IAM), selecione Create new or choose (Criar novo ou escolher).
12. Em IAM role (Função do IAM), escolha `firehose-s3-access-role`.

13. Selecione Permitir.

Configure settings

Configure buffer, compression, logging, and IAM role settings for your delivery stream.

S3 buffer conditions

Firehose buffers incoming records before delivering them to your S3 bucket. Record delivery will be triggered once either of these conditions has been satisfied. Learn more

Buffer size* 5 MB

Specify a buffer size between 1-128 MB

Buffer interval* 60 seconds

Specify a buffer interval between 60-900 seconds

S3 compression and encryption

Firehose can compress records before delivering them to your S3 bucket. Compressed records can also be encrypted in the S3 bucket using a KMS master key. Learn more

S3 compression* (•) Disabled ( ) GZIP ( ) Snappy ( ) Zip

S3 encryption* (•) Disabled ( ) Enabled

Error logging

Firehose can log record delivery errors to CloudWatch Logs. If enabled, a CloudWatch log group and corresponding log streams are created on your behalf. Learn more

Error logging* ( ) Disabled (•) Enabled

IAM role

Firehose uses an IAM role to access your specified resources, such as the S3 bucket and KMS key. Learn more

IAM role* firehose-s3-access-role

Create new or choose

14. Escolha Next (Próximo).
15. Escolha Create delivery stream (Criar fluxo de entrega).

Criar a instância do Amazon EC2 para executar o Kinesis Agent para Windows

Crie a instância do EC2 que usa o Kinesis Agent para Windows para fazer streaming de registros de log por meio do Kinesis Data Firehose.

Para criar a instância do EC2

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. Siga as instruções em Conceitos básicos das instâncias do Windows do Amazon EC2, usando as seguintes etapas adicionais:
   - Para a IAM role (Função do IAM) para a instância, escolha `kinesis-agent-instance-role`.
   - Se você ainda não tiver uma nuvem privada virtual (VPC) pública, conectada à Internet, siga as instruções emConfiguração com o Amazon EC2noGuia do usuário do Amazon EC2 para instâncias do Windows.
   - Crie ou use um grupo de segurança que restrinja o acesso apenas à instância de seu computador ou apenas aos computadores de sua organização. Para obter mais informações, consulteConfiguração com o Amazon EC2noGuia do usuário do Amazon EC2 para instâncias do Windows.
   - Se você especificar um par de chaves existente, certifique-se de ter acesso à chave privada do par de chaves. Ou crie um novo par de chaves e salve a chave privada em um lugar seguro.
   - Antes de continuar, aguarde até que a instância esteja em execução e tenha concluído as duas verificações de integridade.
   - Sua instância requer um endereço IP público. Se um ainda não tiver sido alocado, siga as instruções emEndereços Elastic IPnoGuia do usuário do Amazon EC2 para instâncias do Windows.

Próximas etapas

Etapa 2: Instalar, configurar e executar o Kinesis Agent para Windows

Etapa 2: Instalar, configurar e executar o Kinesis Agent para Windows

Nesta etapa, você usa o AWS Management Console para se conectar remotamente à instância que você executou emCriar a instância do Amazon EC2 para executar o Kinesis Agent para Windows. Depois, instale o Amazon Kinesis Agent para Microsoft Windows na instância, crie e implante o arquivo de configuração do Kinesis Agent para Windows e inicie oAWSkineSistapserviçoServiço do

1. Conecte-se remotamente à instância via Remore Desktop Protocol (RDP), seguindo as instruções emEtapa 2: Conecte-se à sua instânciaonoGuia do usuário do Amazon EC2 para instâncias do Windows.
2. Na instância, use o Windows Server Manager para desativar a configuração de segurança reforçada do Microsoft Internet Explorer para usuários e administradores. Para obter mais informações, consulte Como desativar a configuração de segurança reforçada do Internet Explorer no site do Microsoft TechNet.
3. Na instância, instale e configure o Kinesis Agent para Windows. Para obter mais informações, consulte Instalando o Kinesis Agent para Windows.
4. Na instância, use o Bloco de notas para criar um arquivo de configuração do Kinesis Agent for Windows. Salve o arquivo em `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`. Adicione o seguinte conteúdo ao arquivo de configuração:

```
{
  "Sources": [
    {
      "Id": "JsonLogSource",
      "SourceType": "DirectorySource",
      "RecordParser": "SingleLineJson",
      "Directory": "C:\\LogSource\\",
      "FileNameFilter": "*.log",
      "InitialPosition": 0
    }
  ],
  "Sinks": [
    {
```

```
      "Id": "FirehoseLogStream",
      "SinkType": "KinesisFirehose",
      "StreamName": "log-delivery-stream",
      "Region": "us-east-1",
      "Format": "json",
      "ObjectDecoration": "ComputerName={ComputerName};DT={timestamp:yyyy-MM-dd
 HH:mm:ss}"
    }
  ],
  "Pipes": [
    {
      "Id": "JsonLogSourceToFirehoseLogStream",
      "SourceRef": "JsonLogSource",
      "SinkRef": "FirehoseLogStream"
    }
  ]
}
```

Esse arquivo configura o Kinesis Agent para Windows para enviar registros de log em formato JSON de arquivos no`c:\logsource\`(osource) para um fluxo de entrega do Kinesis Data Firehose chamado`log-delivery-stream`(osink). Antes de cada registro de log ser transmitido ao Kinesis Data Firehose, ele é aprimorado com dois pares de chave-valor extras que contêm o nome do computador e um timestamp.

5. Crie o diretório `c:\LogSource\` e use o Bloco de Notas para criar um arquivo `test.log` nesse diretório com o seguinte conteúdo:

```
{ "Message": "Copasetic message 1", "Severity": "Information" }
{ "Message": "Copasetic message 2", "Severity": "Information" }
{ "Message": "Problem message 2", "Severity": "Error" }
{ "Message": "Copasetic message 3", "Severity": "Information" }
```

6. Em uma sessão privilegiada do PowerShell, use o comando a seguir para iniciar o serviço AWSKinesisTap:

```
Start-Service -ServiceName AWSKinesisTap
```

7. Usando o Explorador de Arquivos, navegue até o diretório `%PROGRAMDATA%\Amazon \AWSKinesisTap\logs`. Abra o arquivo de log mais recente. O arquivo de log deve ser semelhante ao seguinte:

```
2018-09-28 23:51:02.2472 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.AWS.AWSEventSinkFactory.
2018-09-28 23:51:02.2784 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Windows.PerformanceCounterSinkFactory.
2018-09-28 23:51:02.5753 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Core.DirectorySourceFactory.
2018-09-28 23:51:02.5909 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.ExchangeSource.ExchangeSourceFactory.
2018-09-28 23:51:02.5909 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Uls.UlsSourceFactory.
2018-09-28 23:51:02.5909 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Windows.WindowsSourceFactory.
2018-09-28 23:51:02.9347 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Core.Pipes.PipeFactory.
2018-09-28 23:51:03.5128 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.AutoUpdate.AutoUpdateFactory.
2018-09-28 23:51:03.5440 Amazon.KinesisTap.Hosting.LogManager INFO Performance
 counter sink  started.
2018-09-28 23:51:03.7628 Amazon.KinesisTap.Hosting.LogManager INFO
 KinesisFirehoseSink id FirehoseLogStream for StreamName log-delivery-stream
 started.
2018-09-28 23:51:03.7784 Amazon.KinesisTap.Hosting.LogManager INFO Connected source
 JsonLogSource to sink FirehoseLogStream
2018-09-28 23:51:03.7940 Amazon.KinesisTap.Hosting.LogManager INFO DirectorySource
 id JsonLogSource watching directory C:\LogSource\ with filter *.log started.
```

Esse arquivo de log indica que o serviço foi iniciado e os registros de log agora estão sendo coletados do diretório `c:\LogSource\`. Cada linha é analisada como um único objeto JSON. Os pares de chave-valor para o nome do computador e o timestamp são adicionados a cada objeto. Em seguida, ele é transmitido para o Kinesis Data Firehose.

8. Em um ou dois minutos, navegue até o bucket do Amazon S3 que você criou noCrie o bucket do Amazon S3Usando o Console de Gerenciamento da AWS. Certifique-se de que você tenha escolhido a região correta no console.

   Nesse bucket, existe uma pasta para o ano atual. Abra essa pasta para revelar uma pasta para o mês atual. Abra essa pasta para revelar uma pasta para o dia atual. Abra essa pasta para revelar uma pasta para a hora atual (em UTC). Abra essa pasta para revelar um ou mais itens que começam com o nome `log-delivery-stream`.

9. Abra o conteúdo do item mais recente para confirmar se os registros de log foram armazenados com êxito no Amazon S3 com as melhorias desejadas. Se tudo estiver configurado corretamente, o conteúdo será semelhante ao seguinte:

```
{"Message":"Copasetic message 1","Severity":"Information","ComputerName":"EC2AMAZ-
ABCDEFGH","DT":"2018-09-28 23:51:04"}
{"Message":"Copasetic message 2","Severity":"Information","ComputerName":"EC2AMAZ-
ABCDEFGH","DT":"2018-09-28 23:51:04"}
{"Message":"Problem message 2","Severity":"Error","ComputerName":"EC2AMAZ-
ABCDEFGH","DT":"2018-09-28 23:51:04"}
{"Message":"Copasetic message 3","Severity":"Information","ComputerName":"EC2AMAZ-
ABCDEFGH","DT":"2018-09-28 23:51:04"}
```

10. Para obter informações sobre como resolver qualquer um dos problemas a seguir, consulte Solução de problemas do Amazon Kinesis Agent para Microsoft Windows:

    - O arquivo de log do Kinesis Agent para Windows contém erros.
    - As pastas ou itens esperados no Amazon S3 não existem.
    - O conteúdo de um item do Amazon S3 está incorreto.

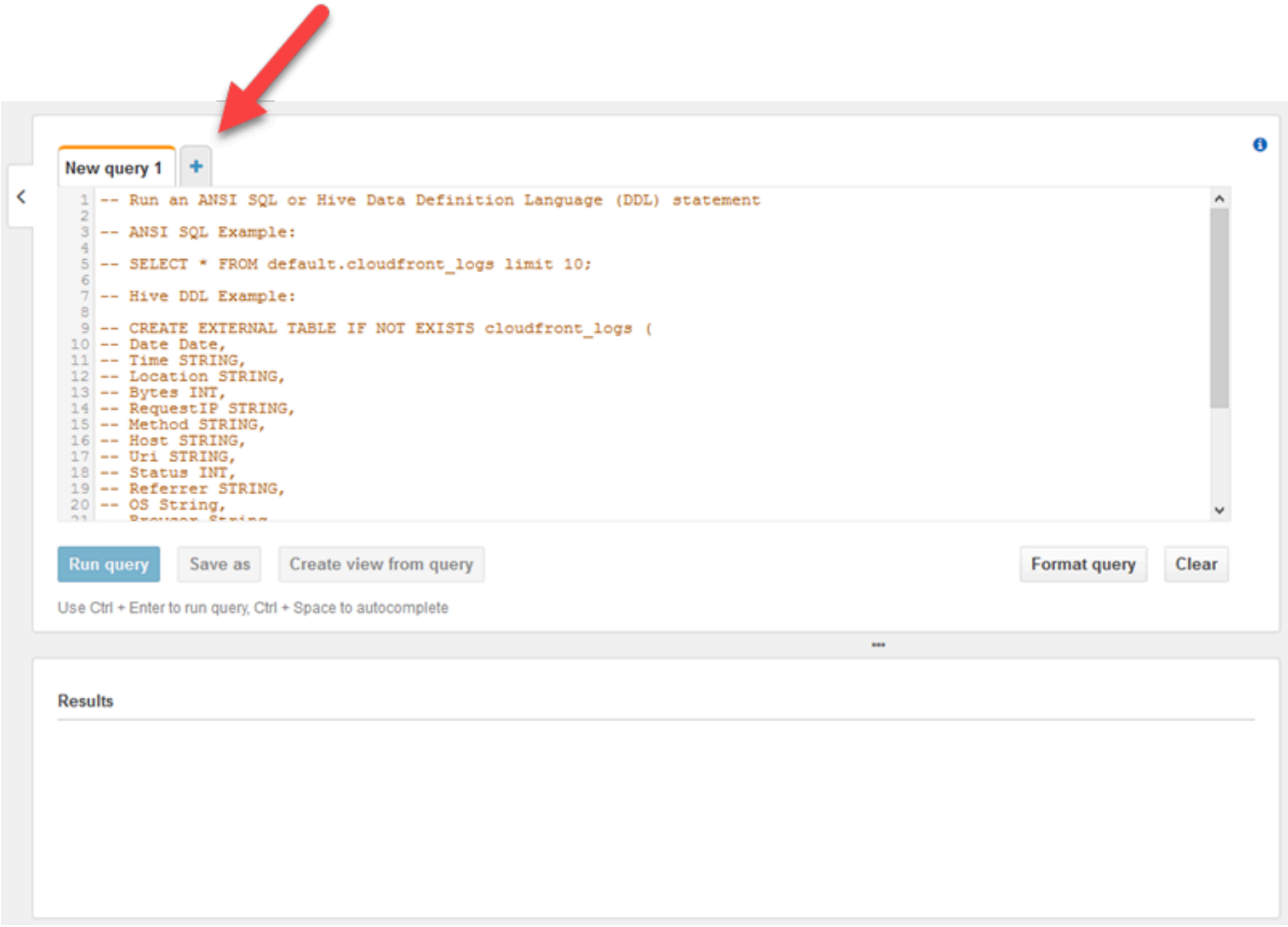
Próximas etapas

Etapa 3: Consulte os dados de log no Amazon S3

Etapa 3: Consulte os dados de log no Amazon S3

Na etapa final deste Amazon Kinesis Agent para Microsoft WindowsTutorial do, você usa o Amazon Athena para consultar os dados de log armazenados no Amazon Simple Storage Service (Amazon S3).

1. Abra o console do Athena em https://console.aws.amazon.com/athena/.
2. Escolha o sinal de adição (+) na janela de consulta do Athena para criar uma nova janela de consulta.



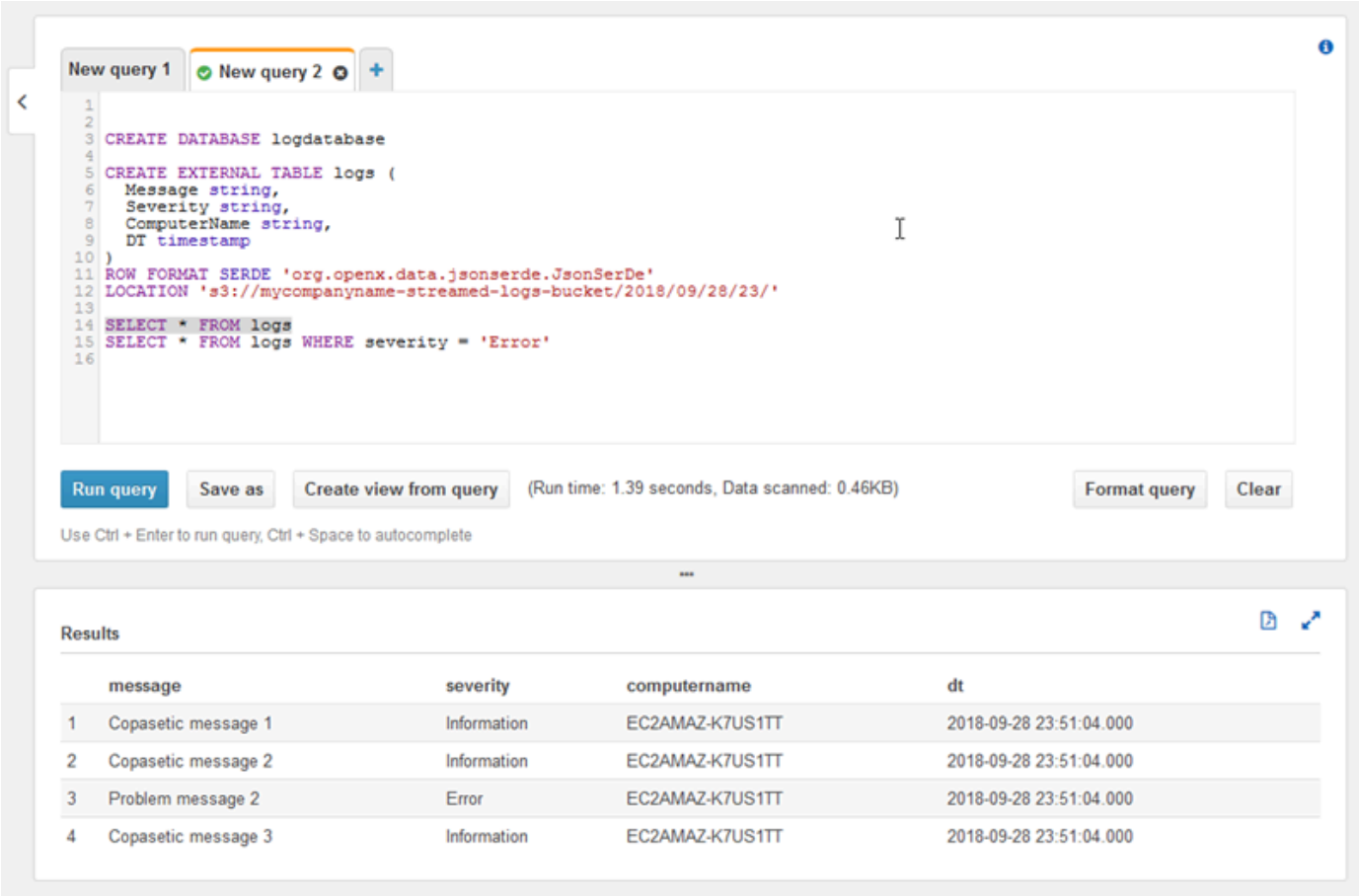
3. Insira o seguinte texto na janela de consulta:

```
CREATE DATABASE logdatabase

CREATE EXTERNAL TABLE logs (
  Message string,
  Severity string,
```

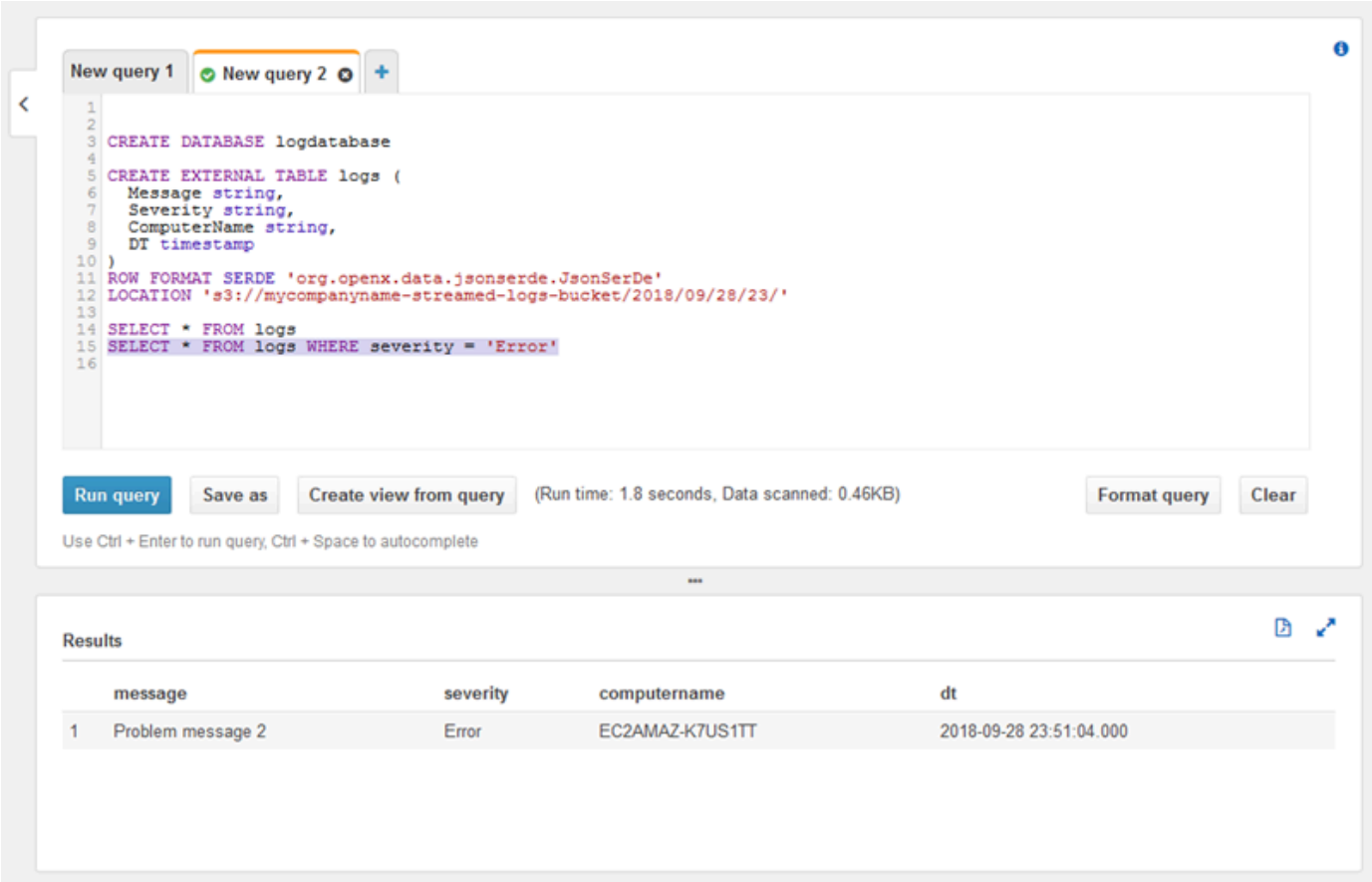
```
  ComputerName string,
  DT timestamp
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://bucket/year/month/day/hour/'

SELECT * FROM logs
SELECT * FROM logs WHERE severity = 'Error'
```

Substitua *bucket* pelo nome do bucket criado em Crie o bucket do Amazon S3. Substituir*year*,*month*,*day*e*hour*Com o ano, mês, dia e hora em que o arquivo de log do Amazon S3 foi criado em UTC.

4. Selecione o texto para a instrução `CREATE DATABASE` e Run query (Executar consulta). Isso cria o banco de dados de logs no Athena.
5. Selecione o texto para a instrução `CREATE EXTERNAL TABLE` e Run query (Executar consulta). Isso cria uma tabela do Athena que faz referência ao bucket do S3 com os dados de log, mapeando o esquema para o JSON ao esquema para a tabela do Athena.
6. Selecione o texto para a primeira instrução `SELECT` e Run query (Executar consulta). Isso exibe todas as linhas na tabela.



7. Selecione o texto para a segunda instrução `SELECT` e Run query (Executar consulta). Isso exibe somente as linhas na tabela que representam os registros de log com uma gravidade no nível de `Error`. Esse tipo de consulta localiza registros de log interessantes em um conjunto de registros de log potencialmente grandes.



Próximas etapas

Use o Console de Gerenciamento da AWS para limpar os recursos criados durante o tutorial:

1. Encerre a instância do EC2 (consulte a etapa 3 em Conceitos básicos das instâncias do Windows do Amazon EC2).

   > **Important**
   >
   > Se você tiver iniciado uma instância que não estava dentro doNível gratuito da AWS, você será cobrado pela instância até que a encerre.

2. Exclua o fluxo de entrega do Kinesis Data Firehose.

   a. Abra o console Kinesis Data Firehose emhttps://console.aws.amazon.com/firehose/.

   b. Selecione o fluxo de entrega que você criou.

   c. Escolha Delete (Excluir).

d. Escolha Delete delivery stream (Excluir fluxo de entrega).

3. Exclua o bucket do S3. Para obter instruções, consulte Como eu faço para excluir um bucket do S3? no Guia do usuário do Amazon Simple Storage Service Console.

Para obter mais informações, consulte os tópicos a seguir:

- Configurando o Amazon Kinesis Agent para Microsoft Windows
- O que é o Amazon Kinesis Data Firehose?
- O que é o Amazon S3?
- O que é o Amazon Athena?

Solução de problemas do Amazon Kinesis Agent para Microsoft Windows

Use as instruções a seguir para diagnosticar e corrigir problemas ao usar o Amazon Kinesis Agent para Microsoft Windows.

Tópicos

Não é feito streaming de dados de desktops nem de servidores para os serviços esperados da AWS

Symptoms

Ao analisar logs, eventos e métricas hospedados por vários serviços da AWS que estão configurados para receber streams de dados do Kinesis Agent para Windows, você observa que o streaming de dados não está sendo realizado para o Kinesis Agent para Windows.

Causes

Há várias causas possíveis para esse problema:

- Uma origem, um coletor ou um pipe está configurado incorretamente.
- A autenticação para o Kinesis Agent para Windows do está configurada incorretamente.
- A autorização para o Kinesis Agent para Windows do está configurada incorretamente.
- Há um erro em uma expressão regular fornecida em uma declaração `DirectorySource`.
- Um diretório inexistente está especificado para uma declaração `DirectorySource`.

- Valores inválidos estão sendo fornecidos para os serviços da AWS que, por sua vez, rejeitam solicitações do Kinesis Agent para Windows.
- Um coletor está fazendo referência a um recurso que não existe na região da AWS especificada ou implícita da.
- Uma consulta inválida está especificada para uma declaração `WindowsEventLogSource`.
- Um valor inválido está especificado para o par de chave-valor `InitialPosition` para uma origem.
- O arquivo de configuração `appsettings.json` não está em conformidade com o esquema JSON para esse arquivo.
- Os dados estão fazendo streaming para uma região diferente da selecionada no AWS Management Console.
- Kinesis Agent para Windows não está instalado corretamente ou não está em execução.

Resolutions

Para resolver problemas de streaming de dados, execute as seguintes etapas:

1. Examine os logs do Kinesis Agent para Windows no`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`Diretório. Procure a string `ERROR`.
   a. Se uma origem ou um coletor não foi carregado, faça o seguinte:
      i. Examine a mensagem de erro e encontre o `Id` da origem ou do coletor.
      ii. Verifique a declaração de origem ou de coletor que corresponde a esse `Id` no arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` para todos os erros relacionados à mensagem de erro encontrada. Para obter mais detalhes, consulte Configurando o Amazon Kinesis Agent para Microsoft Windows.
      iii. Corrija todos os problemas do arquivo de configuração relacionados ao erro.
      iv. Interrompa e inicie o serviço `AWSKinesisTap`. Depois, verifique o arquivo de log mais recente para ver se os problemas de configuração foram resolvidos.
   b. Se a mensagem de erro indicar que um `SourceRef` ou `SinkRef` não foi encontrado para um pipe, faça o seguinte:
      i. Anote o `Id` do pipe.
      ii. Examine a declaração de pipe no arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` correspondente ao `Id` anotado. Certifique-se de que os valores dos pares de chave/valor `SourceRef` e `SinkRef` sejam `Id`s corretamente

digitados para as declarações de origem e de coletor que você pretendia referenciar. Corrija todos os erros de digitação ou ortografia. Se uma declaração de origem ou coletor estiver ausente do arquivo de configuração, adicione a declaração. Para obter mais informações, consulte Configurando o Amazon Kinesis Agent para Microsoft Windows.

iii. Interrompa e inicie o serviço `AWSKinesisTap`. Depois, verifique o arquivo de log mais recente para ver se os problemas de configuração foram resolvidos.

c. Se a mensagem de erro indicar que um usuário ou função do IAM do não está autorizado a executar determinadas operações, faça o seguinte:

i. Verifique se o usuário ou a função correta do IAM está sendo usado pelo Kinesis Agent para Windows. Se não for, reviseConfiguração de segurança do coletore ajuste o modo como o Kinesis Agent para Windows é autenticado para garantir que o usuário ou a função do IAM correto esteja sendo usado.

ii. Se a função ou o usuário correto do IAM estiver sendo usado, com o AWS Management Console, examine as políticas que estão associadas ao usuário ou à função. Verifique se o usuário ou a função tem todas as permissões mencionadas na mensagem de erro para todos os recursos da AWS que o Kinesis Agent para Windows acessa. Para obter mais informações, consulte Configuração da autorização da.

iii. Interrompa e inicie o serviço `AWSKinesisTap`. Depois verifique o arquivo de log mais recente para ver se os problemas de segurança foram resolvidos.

d. Se a mensagem de erro indicar que há um erro de argumento ao analisar uma expressão regular que está contida no arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`, faça o seguinte:

i. Examine a expressão regular no arquivo de configuração.

ii. Verifique a sintaxe da expressão regular. Há vários sites que você pode usar para verificar expressões regulares ou use as seguintes linhas de comando para verificar expressões regulares para uma declaração de origem `DirectorySource`:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
ktdiag.exe /r sourceId
```

Substitua *`sourceId`* pelo valor do par de chave/valor `Id` da declaração de origem `DirectorySource` com uma expressão regular incorreta.

iii. Faça as correções necessárias na expressão regular no arquivo de configuração para que ela seja válida.

iv. Interrompa e inicie o serviço `AWSKinesisTap`. Depois, verifique o arquivo de log mais recente para ver se os problemas de configuração foram resolvidos.

e. Se a mensagem de erro indicar que há um erro de argumento ao analisar uma expressão regular que não está contida no arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` e está relacionada a um coletor específico, faça o seguinte:

i. Localize a declaração do coletor no arquivo de configuração.

ii. Verifique se os pares de chave-valor que são especificamente relacionados a um serviço da AWS estão usando nomes em conformidade com as regras de validação desse serviço. Por exemplo, nomes de grupos do CloudWatch Logs devem conter apenas um conjunto de caracteres especificado usando a expressão regular `[\.\-_/#A-Za-z0-9]+`.

iii. Corrija os nomes inválidos nos pares de chave-valor para a declaração do coletor e garanta que esses recursos estejam configurados corretamente na AWS.

iv. Interrompa e inicie o serviço `AWSKinesisTap`. Depois, verifique o arquivo de log mais recente para ver se os problemas de configuração foram resolvidos.

f. Se a mensagem de erro indicar que uma origem ou um coletor não pode ser carregado devido a um parâmetro nulo ou ausente, faça o seguinte:

i. Anote o `Id` da origem ou do coletor.

ii. Localize a declaração de origem ou de coletor que corresponda ao `Id` anotado no arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

iii. Analise os pares de chave-valor que são fornecidos na declaração de origem ou coletor em comparação com os requisitos de tipo de origem ou coletor na documentação Configurando o Amazon Kinesis Agent para Microsoft Windows relevante para o tipo de coletor. Adicione todos os pares chave-valor ausentes obrigatórios para a declaração de origem ou coletor.

iv. Interrompa e inicie o serviço `AWSKinesisTap`. Depois, verifique o arquivo de log mais recente para ver se os problemas de configuração foram resolvidos.

g. Se a mensagem de erro indicar que um nome de diretório é inválido, faça o seguinte:

i. Localize o nome de diretório inválido no arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

ii. Verifique se esse diretório existe e contém os arquivos de log dos quais deve ser feito streaming.

iii. Corrija todos os erros de digitação ou erros no nome do diretório especificado no arquivo de configuração.

iv. Interrompa e inicie o serviço `AWSKinesisTap`. Depois, verifique o arquivo de log mais recente para ver se os problemas de configuração foram resolvidos.

h. Se a mensagem de erro indicar que um recurso não existe:

i. Localize a referência do recurso que não existe em uma declaração de coletor no arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

ii. Use o AWS Management Console para localizar o recurso na região correta da AWS que deve ser usada na declaração de coletor. Compare-o com o que foi especificado no arquivo de configuração.

iii. Altere a declaração do coletor no arquivo de configuração para ter o nome de recurso e a região corretos.

iv. Interrompa e inicie o serviço `AWSKinesisTap`. Depois, verifique o arquivo de log mais recente para ver se os problemas de configuração foram resolvidos.

i. Se a mensagem de erro indicar que uma consulta é inválida para uma `WindowsEventLogSource`, faça o seguinte:

i. No arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap \appsettings.json`, localize a declaração `WindowsEventLogSource` com o mesmo `Id` que na mensagem de erro.

ii. Verifique se o valor do par de chave-valor `Query` na declaração de origem está em conformidade com as consultas de eventos e Event XML.

iii. Faça as alterações na consulta para que ela fique em conformidade.

iv. Interrompa e inicie o serviço `AWSKinesisTap`. Depois, verifique o arquivo de log mais recente para ver se os problemas de configuração foram resolvidos.

j. Se a mensagem de erro indicar que há uma posição inicial inválida, faça o seguinte:

i. No arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap \appsettings.json`, localize a declaração de origem com o mesmo `Id` que na mensagem de erro.

ii. Altere o valor do par de chave-valor `InitialPosition` na declaração de origem para estar em conformidade com os valores permitidos, conforme descrito em Configuração de marcador.

iii. Interrompa e inicie o serviço `AWSKinesisTap`. Depois, verifique o arquivo de log mais recente para ver se os problemas de configuração foram resolvidos.

2. Garanta que o arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap \appsettings.json` cumpra o esquema JSON.

a. Em uma janela do prompt de comando, invoque as seguintes linhas:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
%PROGRAMFILES%\Amazon\AWSKinesisTap\ktdiag.exe /c
```

b. Corrija todos os problemas detectados com o arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

c. Interrompa e inicie o serviço `AWSKinesisTap`. Depois, verifique o arquivo de log mais recente para ver se os problemas de configuração foram resolvidos.

3. Altere o nível de registro em log para obter informações de registro em log mais detalhadas.

a. Substitua o arquivo de configuração `%PROGRAMFILES%\Amazon\AWSKinesisTap\nlog.xml` pelo seguinte conteúdo:

```
<?xml version="1.0" encoding="utf-8" ?>
<nlog xmlns="http://www.nlog-project.org/schemas/NLog.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.nlog-project.org/schemas/NLog.xsd NLog.xsd"
      autoReload="true"
      throwExceptions="false"
      internalLogLevel="Off" internalLogFile="c:\temp\nlog-internal.log" >

  <!--
  See https://github.com/nlog/nlog/wiki/Configuration-file
  for information on customizing logging rules and outputs.
   -->
  <targets>
    <!--
    add your targets here
    See https://github.com/nlog/NLog/wiki/Targets for possible targets.
    See https://github.com/nlog/NLog/wiki/Layout-Renderers for the possible layout
 renderers.
    -->

    <target name="logfile"
            xsi:type="File"
            layout="${longdate} ${logger} ${uppercase:${level}} ${message}"
            fileName="${specialfolder:folder=CommonApplicationData}/Amazon/
KinesisTap/logs/KinesisTap.log"
     maxArchiveFiles="90"
     archiveFileName="${specialfolder:folder=CommonApplicationData}/Amazon/
KinesisTap/logs/Archive-{################}.log"
```

```
      archiveNumbering="Date"
      archiveDateFormat="yyyy-MM-dd"
      archiveEvery="Day"
      />
  </targets>

  <rules>
    <logger name="*" minlevel="Debug" writeTo="logfile" />
  </rules>
</nlog>
```

b. Interrompa e inicie o serviço `AWSKinesisTap`. Depois verifique o arquivo de log mais recente para ver se há mensagens adicionais no log que podem ajudar a diagnosticar e resolver o problema.

4. Verifique se você está examinando os recursos na região correta do no AWS Management Console.
5. Verifique se o Kinesis Agent para Windows está instalado e em execução.
   a. No Windows, escolha Start (Iniciar) e navegue até Control Panel (Painel de controle), Administrative Tools (Ferramentas administrativas), Services (Serviço).
   b. Encontre o serviço AWSKinesisTap.
   c. Se o serviço AWSKinesisTap não estiver visível, instale o Kinesis Agent para Windows usando as instruções em[Conceitos básicos do Amazon Kinesis Agent para Microsoft Windows](#).
   d. Se o serviço estiver visível, determine se ele está em execução. Se não estiver em execução, abra o menu de contexto (clique com o botão direito do mouse) do serviço e escolha Start (Iniciar).
   e. Verifique se o serviço foi iniciado examinando o arquivo de log mais recente no diretório `%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`.

Aplica-se a

Essas informações se aplicam ao Kinesis Agent para Windows versão 1.0.0.115 e posterior.

Às vezes os dados esperados estão ausentes

Symptoms

O Kinesis Agent para Windows faz streaming dos dados com êxito na maior parte do tempo, mas ocasionalmente alguns dados ficam ausentes.

Causes

Há várias causas possíveis para esse problema:

- O recurso de marcação não está sendo usado.
- Os limites de taxa de dados para os serviços da AWS da foram excedidos com base na configuração atual desses serviços.
- Os limites de taxas de chamada de API para serviços da AWS da foram excedidos com base no atual`appsettings.json`e os limites da conta da AWS.

Resolutions

Para resolver problemas de ausência de dados, execute as seguintes etapas:

1. Considere usar o recurso de marcação documentado em Configuração de marcador. Ele ajuda a garantir que todos os dados sejam enviados, mesmo quando o Kinesis Agent para Windows é interrompido e iniciado.
2. Use as métricas incorporadas do Kinesis Agent para o Windows para descobrir problemas:
   a. Habilite o streaming do Kinesis Agent para Windows, conforme descrito emConfiguração do Kinesis Agent para Pipes Métricos do Windows.
   b. Se houver um número significativo de erros não recuperáveis para um ou mais coletores, determine quantos bytes ou registros estão sendo enviados por segundo. Depois determine se está dentro dos limites configurados para esses serviços da AWS na região e na conta onde está sendo feito streaming dos dados.
   c. Quando os limites são excedidos, reduza a taxa ou a quantidade de dados que estão sendo enviados, solicite aumentos de limite ou aumente o estilhaçamento, se aplicável.
   d. Depois de fazer ajustes, continue a monitorar as métricas incorporadas do Kinesis Agent para Windows para garantir que a situação tenha sido resolvida.

Para obter mais informações sobre os limites do Kinesis Data Streams, consulteLimites do Kinesis Data StreamsnoKinesis Data Streams. Para obter mais informações sobre os limites do Kinesis Data Firehose, consulteLimites do Amazon Kinesis Data Firehose.

Aplica-se a

Essas informações se aplicam ao Kinesis Agent para Windows versão 1.0.0.115 e posterior.

Os dados chegam em um formato incorreto

Symptoms

Os dados chegam ao serviço da AWS no formato incorreto.

Causes

Há várias causas possíveis para esse problema:

- O valor do par de chave-valor `Format` para uma declaração de coletor no arquivo de configuração `appsettings.json` está incorreto.
- O valor para o par de chave-valor `RecordParser` em uma declaração `DirectorySource` está incorreto.
- As expressões regulares em uma declaração `DirectorySource` que usa o analisador de registros `Regex` estão incorretas.

Resolutions

Para resolver problemas de formatação incorreta, execute as seguintes etapas:

1. Revise as declarações do coletor no arquivo de configuração `%PROGRAMFILES%\Amazon \AWSKinesisTap\appsettings.json`.
2. Certifique-se de que o valor correto do par de chave-valor `Format` esteja especificado para cada declaração de coletor. Para obter mais informações, consulte Declarações de coletor.
3. Se as origens com declarações `DirectorySource` estão conectadas por pipes a coletores que especificam valores `xml` ou `json` para o par de chave-valor `Format`, certifique-se de que essas origens especifiquem um dos seguintes valores para o par de chave-valor `RecordParser`:
   - `SingleLineJson`

- `Regex`
- `SysLog`
- `Delimited`

Outros analisadores de registros são apenas baseados em texto e não funcionam corretamente com coletores que exijam a formatação JSON ou XML.

4. Se os registros de log não estiverem sendo corretamente analisados pelo tipo de origem `DirectorySource`, invoque as seguintes linhas em uma janela de prompt de comando para verificar os pares de chave-valor de timestamp e expressão regular especificados na declaração `DirectorySource`:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
ktdiag.exe /r sourceID
```

Substitua *`sourceID`* pelo valor do par de chave-valor `Id` da declaração de origem `DirectorySource` que parece não estar funcionando corretamente. Corrija todos os problemas relatados pelo `ktdiag.exe`.

Aplica-se a

Essas informações se aplicam ao Kinesis Agent para Windows versão 1.0.0.115 e posterior.

Problemas de desempenho

Symptoms

Aplicativos e serviços têm aumento de latência depois que o Kinesis Agent para Windows é instalado e iniciado.

Causes

Há várias causas possíveis para esse problema:

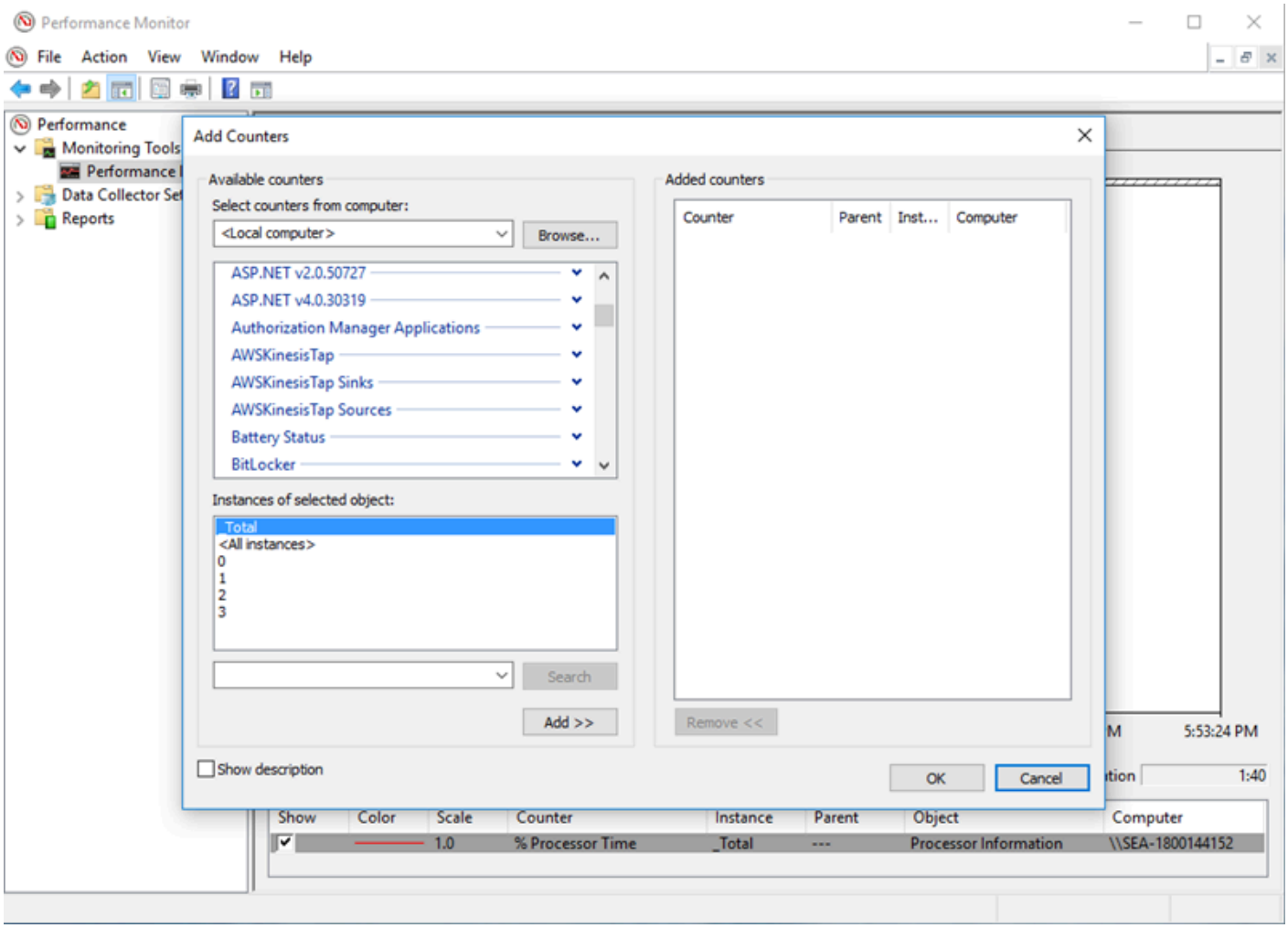
- O computador no qual o Kinesis Agent para Windows é executado não tem capacidade suficiente para fazer streaming do volume de dados desejado.
- Está sendo feito streaming de dados desnecessários para um ou mais serviços da AWS.

- Kinesis Agent para Windows está fazendo streaming de dados para serviços da AWS que não estão configurados para uma taxa de dados tão alta.
- O Kinesis Agent para Windows está invocando operações em serviços da AWS em uma conta na qual o limite da taxa de chamadas de APIs é muito baixo.

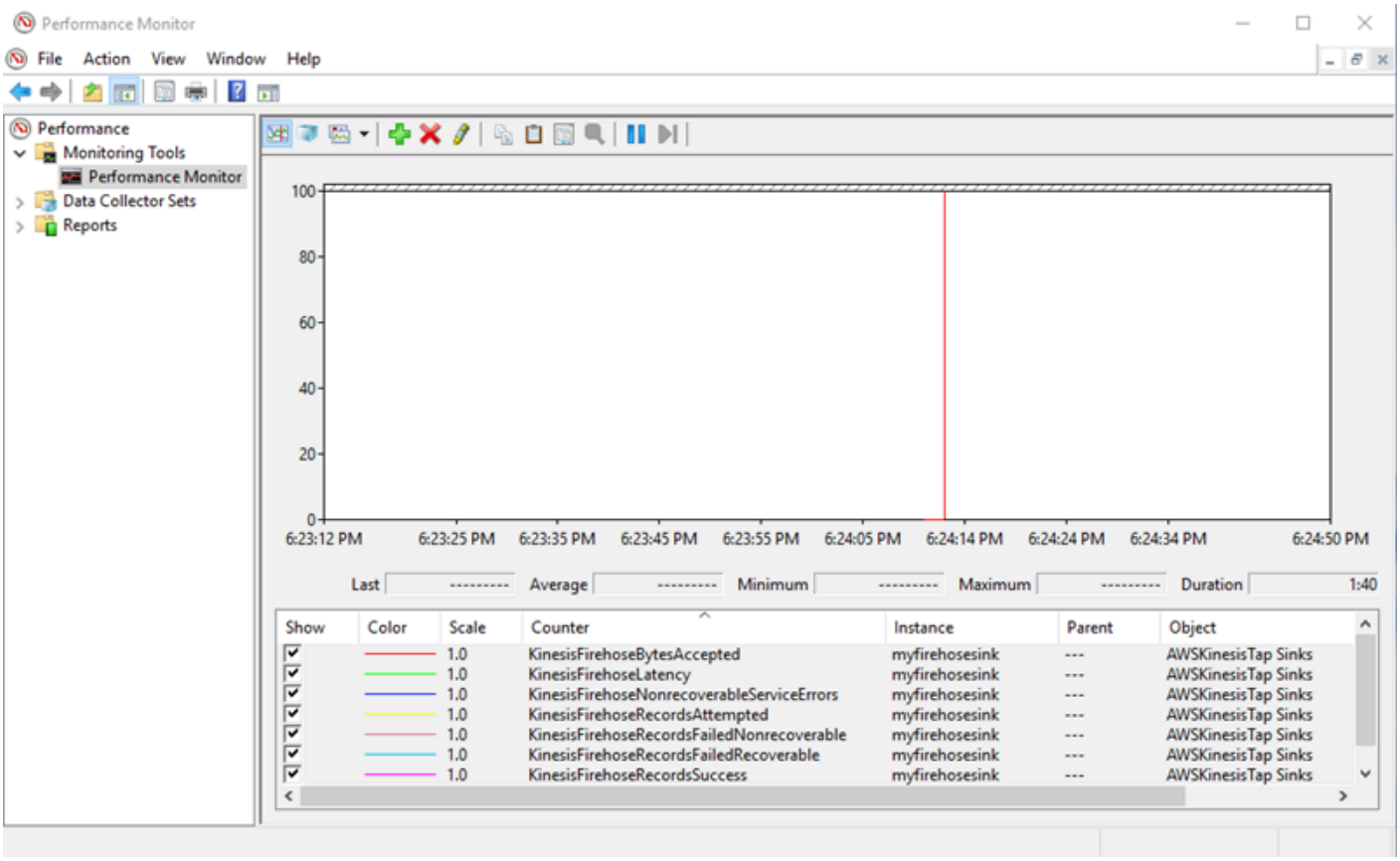
Resolutions

Para resolver problemas de desempenho, execute as seguintes etapas:

1. Use o aplicativo de monitor de recursos do Windows para verificar o uso de memória, CPU, disco e rede. Se você precisa fazer streaming de grandes quantidades de dados com o Kinesis Agent para Windows, pode ser necessário provisionar um computador com capacidades mais elevadas em algumas dessas áreas, dependendo da configuração.
2. Talvez você consiga reduzir o volume de dados registrados usando a filtragem:
   - Consulte o par de chave-valor `Query` em Configuração de WindowsEventLogSource.
   - Consulte a filtragem de pipeline em Configuração de pipes.
   - Consulte a filtragem de métricas do Amazon CloudWatchConfiguração do CloudWatch).
3. Use o aplicativo de monitor de desempenho do Windows para visualizar as métricas do Kinesis Agent para Windows do ou faça streaming dessas métricas para o CloudWatch (consulteOrigem de métricas incorporadas do Kinesis Agent para Windows). No aplicativo de monitor de desempenho do Windows, é possível adicionar contadores para coletores e origens do Kinesis Agent para Windows. Eles estão listados nas categorias de contador AWSKinesisTap Sinks (Coletores do AWSKinesisTap) e AWSKinesisTap Sources (Origens do AWSKinesisTap).



Por exemplo, para diagnosticar problemas de desempenho do Kinesis Data Firehose, adicione oKinesis FirehoseContadores de desempenho.



Se houver um grande número de erros recuperáveis, inspecione os logs mais recentes do Kinesis Agent para Windows no`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`Diretório. Se a limitação estiver ocorrendo para os coletores `KinesisStream` ou `KinesisFirehose`, faça o seguinte:

- Se ocorrer a limitação devido ao streaming muito rápido dos dados, considere aumentar o número de estilhaços para o stream de dados do Kinesis. Para obter mais informações, consulteReestilhaçamento, escalabilidade e processamento paralelonoKinesis Data Streams.
- Considere o aumento do limite de chamadas de API para Kinesis Data Streams ou o aumento do tamanho do buffer para o coletor se as chamadas de API estiverem sendo limitadas. Para obter mais informações, consulteLimites do Kinesis Data StreamsnoKinesis Data Streams.
- Se o streaming dos dados for feito com muita rapidez, considere solicitar um aumento no limite da taxa para o stream de entrega do Kinesis Data Firehose. Ou se as chamadas de API estiverem sendo limitadas, solicite um aumento do limite de chamadas de API (consulte Limites do Amazon Kinesis Firehose) ou aumente o tamanho do buffer para o coletor.
- Depois de aumentar o número de estilhaços em um stream de Kinesis Data Streams ou aumentar o limite de taxa para um stream de entrega do Kinesis Data Firehose, revise o Kinesis Agent para Windows`appsettings.json`Arquivo de configuração para aumentar os registros

ou bytes por segundo para o coletor. Caso contrário, o Kinesis Agent para Windows não pode aproveitar o aumento de limites.

Aplica-se a

Essas informações se aplicam ao Kinesis Agent para Windows versão 1.0.0.115 e posterior.

Sem espaço em disco

Symptoms

O Kinesis Agent para Windows está em execução em uma máquina que tem muito pouco espaço em disco em uma ou mais unidades de disco.

Causes

Há várias causas possíveis para esse problema:

- O Kinesis Agent para o Windows Log de configuração do está incorreto.
- A fila persistente do Kinesis Agent para Windows do está configurada incorretamente.
- Algum outro aplicativo ou serviço está consumindo espaço em disco.

Resolutions

Para resolver problemas de espaço em disco, execute as seguintes etapas:

- Se o espaço em disco estiver baixo no disco que contém os arquivos de log do Kinesis Agent para Windows, examine o diretório de arquivos de log (normalmente`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`). Certifique-se de que um número razoável de arquivos de log esteja sendo mantido e que os arquivos de log tenha um tamanho razoável. Você pode controlar o local, a retenção e o detalhamento dos logs do Kinesis Agent para Windows editando o`%PROGRAMFILES%\Amazon\AWSKinesisTap\Nlog.xml`Arquivo de configuração.
- Quando o recurso de enfileiramento do coletor estiver ativado, examine as declarações de coletor que usam esse recurso. Certifique-se de que o par de chave-valor `QueuePath` faça referência a uma unidade de disco com espaço suficiente para conter o número máximo de lotes especificado usando o par de chave-valor `QueueMaxBatches`. Se isso não for possível, reduza o valor do par

de chave-valor `QueueMaxBatches` para que os dados se encaixem facilmente no espaço em disco restante da unidade especificada.

- Use o Explorador de Arquivos do Windows para localizar os arquivos que consomem o espaço em disco e transfira ou exclua os arquivos em excesso. Altere a configuração do aplicativo ou do serviço que consome grandes quantidades de espaço em disco.

Aplica-se a

Essas informações se aplicam ao Kinesis Agent para Windows versão 1.0.0.115 e posterior.

Ferramentas de solução de problemas

Além de verificar o arquivo de configuração do, é possível usar `oktdiag.exe`, que fornece vários outros recursos para diagnosticar e resolver problemas ao configurar e usar o Kinesis Agent para Windows. A ferramenta `ktdiag.exe` está localizada no diretório `%PROGRAMFILES%\Amazon\AWSKinesisTap`.

- Se você acredita que os arquivos de log com um determinado padrão estão sendo gravados em um diretório, mas não estão sendo processados pelo Kinesis Agent para Windows, use a opção`/w`para verificar se essas alterações estão sendo detectadas. Por exemplo, suponha que os arquivos de log com o nome `*.log` devam ser gravados no diretório `c:\foo`. Você pode usar a opção `/w` ao executar a ferramenta `ktdiag.exe` especificando o diretório e o padrão de arquivo:

  ```
  cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
  ktdiag /w c:\foo *.log
  ```

  Se os arquivos de log estiverem sendo gravados, você poderá ver uma saída semelhante à seguinte:

  ```
  Type any key to exit this program...
  File: c:\foo\log1.log ChangeType: Created
  File: c:\foo\log1.log ChangeType: Deleted
  File: c:\foo\log1.log ChangeType: Created
  File: c:\foo\log1.log ChangeType: Changed
  File: c:\foo\log1.log ChangeType: Changed
  File: c:\foo\log1.log ChangeType: Changed
  File: c:\foo\log1.log ChangeType: Changed
  ```

Se nenhuma saída estiver ocorrendo, há um problema de gravação de logs no aplicativo ou no serviço ou há um problema de configuração de segurança e não um problema com o Kinesis Agent para Windows. Se essa saída estiver ocorrendo, mas o Kinesis Agent para Windows ainda não estiver processando os logs, consulteNão é feito streaming de dados de desktops nem de servidores para os serviços esperados da AWS.

- Às vezes, os logs são gravados apenas ocasionalmente, mas seria útil verificar se o Kinesis Agent para Windows está funcionando corretamente. Use a opção `/log4net` para simular um aplicativo gravando logs com a biblioteca `Log4net`; por exemplo:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
KTDiag.exe /log4net c:\foo\log2.log
```

Isso grava um arquivo de log no estilo `Log4net` no arquivo de log `c:\foo\log2.log` e continua incluindo novas entradas de log até uma tecla ser pressionada. Você pode configurar várias opções usando opções adicionais que podem ser especificadas após o nome do arquivo:

Bloqueio: `-lm`, `-li` ou `-le`

Você pode especificar uma das seguintes opções de bloqueio que controlam como o arquivo de log é bloqueado:

`-lm`

A quantidade mínima de bloqueio é usada no arquivo de log permitindo o acesso máximo ao arquivo de log.

`-li`

Somente threads dentro do mesmo processo podem acessar o log ao mesmo tempo.

`-le`

Apenas um thread de cada vez pode acessar o log. Esse é o padrão.

`-tn:`*`milissegundos`*

Especifica o número de *`milissegundos`* entre a gravação de entradas de log. O padrão é 1.000 milissegundos (1 segundo).

`-sm:`*`bytes`*

Especifica o número de *`bytes`* para cada entrada de log. O padrão é 1.000 bytes.

`-bk:`*`número`*

Especifica o *`número`* de entradas de log a serem gravadas por vez. O padrão é 1.

- Às vezes, é útil simular um aplicativo que grave no log de eventos do Windows. Use a opção `/e` para gravar entradas em um log de eventos do Windows, por exemplo:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
KTDiag.exe /e Application
```

Isso grava entradas no log de eventos do aplicativo do Windows até uma tecla ser pressionada. Você também pode especificar as seguintes opções adicionais após o nome do log:

`-tn:`*`milissegundos`*

Especifica o número de *`milissegundos`* entre a gravação de entradas de log. O padrão é 1.000 milissegundos (1 segundo).

`-sm:`*`bytes`*

Especifica o número de *`bytes`* para cada entrada de log. O padrão é 1.000 bytes.

`-bk:`*`número`*

Especifica o *`número`* de entradas de log a serem gravadas por vez. O padrão é 1.

Criação de plug-ins do Kinesis Agent para Windows

Para a maioria das situações, não é necessário criar um plug-in do Amazon Kinesis Agent para Microsoft Windows. O Kinesis Agent para Windows é altamente configurável e contém origens e coletores avançados, como`DirectorySource`e`KinesisStream`, que são suficientes para a maioria dos cenários. Para obter detalhes sobre as origens e os coletores existentes, consulte Configurando o Amazon Kinesis Agent para Microsoft Windows.

Para cenários incomuns, pode ser necessário estender o Kinesis Agent para Windows usando um plug-in personalizado. Alguns desses cenários incluem o seguinte:

- Empacotar uma declaração `DirectorySource` complexa usando os analisadores de registros `Delimited` ou `Regex` para que seja fácil aplicar em muitos tipos diferentes de arquivos de configuração.
- Criar uma nova origem que não seja baseada em arquivos ou que exceda os recursos de análise fornecidos pelo analisadores de registros existentes.
- Criar um coletor para um serviço da AWS que atualmente não seja compatível.

Tópicos

Introdução ao Kinesis Agent para plug-ins do Windows

Não há nada de especial com plug-ins personalizados. Todas as origens e coletores existentes usam os mesmos mecanismos que os plug-ins personalizados usam para serem carregados quando o Kinesis Agent para Windows é iniciado, e eles instanciam plug-ins relevantes depois de ler a`appsettings.json`Arquivo de configuração.

Quando o Kinesis Agent para Windows é iniciado, ocorre a seguinte sequência:

1. O Kinesis Agent para Windows examina as montagens no diretório de instalação (`%PROGRAMFILES%\Amazon\AWSKinesisTap`) para classes que implementam

oIFactory<T>definida na interfaceAmazon.KinesisTap.CoreAssembly. Essa interface é definida emAmazon.KinesisTap.Core\Infrastructure\IFactory.csno código-fonte do Kinesis Agent para Windows.

2. O Kinesis Agent para Windows carrega as montagens que contêm essas classes e invoca a propriedadeRegisterFactorynessas classes.
3. O Kinesis Agent para Windows carrega oappsettings.jsonArquivo de configuração. Para cada origem e coletor no arquivo de configuração, os pares de chave-valor SourceType e SinkType são examinados. Se houver fábricas registradas com o mesmo nome que os valores dos pares de chave-valor SourceType e SinkType, o método CreateInstance é invocado nessas fábricas. O método CreateInstance transmite a configuração e outras informações como um objeto IPluginContext. O método CreateInstance é responsável por configurar e inicializar o plug-in.

Para um plug-in funcionar corretamente, deve haver uma classe de fábrica registrada que crie o plug-in, e a classe do plug-in deve ser definida.

O código-fonte do Kinesis Agent para Windows está localizado emhttps://github.com/awslabs/kinesis-agent-windows.

Implementando fábricas de plugins do Kinesis Agent para Windows

Siga estas etapas para implementar uma fábrica de plug-ins do Kinesis Agent para Windows.

Para criar uma fábrica de plugins do Kinesis Agent para Windows

1. Crie um projeto de biblioteca C# direcionado para .NET Framework 4.6.
2. Adicione uma referência ao conjunto Amazon.KinesisTap.Core. Esse conjunto está localizado no%PROGRAMFILES%\Amazon\AWSKinesisTapapós a instalação do Kinesis Agent para Windows.
3. Use NuGet para instalar o pacote Microsoft.Extensions.Configuration.Abstractions.
4. Use NuGet para instalar o pacote System.Reactive.
5. Use NuGet para instalar o pacote Microsoft.Extensions.Logging.
6. Crie uma classe de fábrica que implemente IFactory<IEventSource> para origens ou IFactory<IEventSink> para coletores. Adicione os métodos RegisterFactory e CreateInstance.

Por exemplo, o código a seguir cria uma fábrica de plug-ins do Kinesis Agent para Windows que cria uma origem que gera dados aleatórios:

```
using System;
using Amazon.KinesisTap.Core;
using Microsoft.Extensions.Configuration;

namespace MyCompany.MySources
{
    public class RandomSourceFactory : IFactory<ISource>
    {
        public void RegisterFactory(IFactoryCatalog<ISource> catalog)
        {
            catalog.RegisterFactory("randomsource", this);
        }

        public ISource CreateInstance(string entry, IPlugInContext context)
        {
            IConfiguration config = context.Configuration;

            switch (entry.ToLower())
            {
                case "randomsource":
                    string rateString = config["Rate"];
                    string maxString = config["Max"];
                    TimeSpan rate;
                    int max;

                    if (string.IsNullOrWhiteSpace(rateString))
                    {
                        rate = TimeSpan.FromSeconds(30);
                    }
                    else
                    {
                        if (!TimeSpan.TryParse(rateString, out rate))
                        {
                            throw new Exception($"Rate {rateString} is invalid for
 RandomSource.");
                        }
                    }

                    if (string.IsNullOrWhiteSpace(maxString))
```

```
                    {
                        max = 1000;
                    }
                    else
                    {
                        if (!int.TryParse(maxString, out max))
                        {
                            throw new Exception($"Max {maxString} is invalid for
 RandomSource.");
                        }
                    }

                    return new RandomSource(rate, max, context);
                default:
                    throw new ArgumentException($"Source {entry} is not
 recognized.", entry);
            }
        }
    }
}
```

A instrução `switch` é usada no método `CreateInstance` caso você deseje aprimorar a fábrica para criar diferentes tipos de instâncias.

Para criar uma fábrica de coletores que crie um coletor que não faz nada, use uma classe semelhante à seguinte:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Amazon.KinesisTap.Core;
using Microsoft.Extensions.Configuration;

namespace MyCompany.MySinks
{
    public class NullSinkFactory : IFactory<IEventSink>
    {
        public void RegisterFactory(IFactoryCatalog<IEventSink> catalog)
        {
            catalog.RegisterFactory("nullsink", this);
        }
```

```
        public IEventSink CreateInstance(string entry, IPlugInContext context)
        {
            IConfiguration config = context.Configuration;

            switch (entry.ToLower())
            {
                case "nullsink":
                    return new NullSink(context);
                default:
                    throw new Exception("Unrecognized sink type {entry}.");
            }
        }
    }
}
```

Implementando origens de plug-in do Kinesis Agent para Windows

Siga estas etapas para implementar uma origem de plug-ins do Kinesis Agent para Windows.

Para criar uma fonte de plug-in do Kinesis Agent para Windows

1. Adicione uma classe que implemente a interface `IEventSource<out T>` ao projeto criado anteriormente para a origem.

   Por exemplo, use o código a seguir para definir uma origem que gere dados aleatórios:

```
using System;
using System.Reactive.Subjects;
using System.Timers;
using Amazon.KinesisTap.Core;
using Microsoft.Extensions.Logging;

namespace MyCompany.MySources
{
    public class RandomSource : EventSource<RandomData>, IDisposable
    {
        private TimeSpan _rate;
        private int _max;
        private Timer _timer = null;
        private Random _random = new Random();
```

```
        private ISubject<IEnvelope<RandomData>> _recordSubject = new
Subject<IEnvelope<RandomData>>();



        public RandomSource(TimeSpan rate, int max, IPlugInContext context) :
base(context)
        {
            _rate = rate;
            _max = max;
        }

        public override void Start()
        {
            try
            {
                CleanupTimer();
                _timer = new Timer(_rate.TotalMilliseconds);
                _timer.Elapsed += (Object source, ElapsedEventArgs args) =>
                {
                    var data = new RandomData()
                    {
                        RandomValue = _random.Next(_max)
                    };
                    _recordSubject.OnNext(new Envelope<RandomData>(data));
                };
                _timer.AutoReset = true;
                _timer.Enabled = true;
                _logger?.LogInformation($"Random source id {this.Id} started with
rate {_rate.TotalMilliseconds}.");
            }
            catch (Exception e)
            {
                _logger?.LogError($"Exception during start of RandomSource id
{this.Id}: {e}");
            }
        }

        public override void Stop()
        {
            try
            {
                CleanupTimer();
```

```
                _logger?.LogInformation($"Random source id {this.Id} stopped.");

            }
            catch (Exception e)
            {
                _logger?.LogError($"Exception during stop of RandomSource id
 {this.Id}: {e}");
            }
        }

        private void CleanupTimer()
        {
            if (_timer != null)
            {
                _timer.Enabled = false;
                _timer?.Dispose();
                _timer = null;
            }
        }

        public override IDisposable Subscribe(IObserver<IEnvelope<RandomData>>
 observer)
        {
            return this._recordSubject.Subscribe(observer);
        }

        public void Dispose()
        {
            CleanupTimer();
        }
    }
}
```

Neste exemplo, a classe `RandomSource` é herdada da classe `EventSource<T>`, pois ela fornece a propriedade `Id`. Embora este exemplo não ofereça suporte a marcadores, essa classe básica também é útil para implementar essa funcionalidade. Os envelopes fornecem uma maneira de armazenar metadados e encapsular dados arbitrários para o streaming para coletores. A classe `RandomData` é definida na próxima etapa e representa o tipo de objeto de saída a partir dessa origem.

2. Adicione uma classe ao projeto definido anteriormente que contém os dados que são transmitidos por streaming a partir da origem.

Por exemplo, um contêiner de dados aleatórios poderia ser definido da seguinte forma:

```
namespace MyCompany.MySources
{
    public class RandomData
    {
        public int RandomValue { get; set; }
    }
}
```

3. Compile o projeto definido anteriormente.
4. Copie o conjunto no diretório de instalação do Kinesis Agent para Windows.
5. Crie ou atualize um`appsettings.json`que use a nova origem e coloque-a no diretório de instalação do Kinesis Agent para Windows.
6. Interrompa e inicie o Kinesis Agent para Windows.
7. Verifique o arquivo de log atual do Kinesis Agent para Windows (geralmente localizado na seção`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`) para garantir que não haja problemas com o plug-in de origem personalizado.
8. Certifique-se de que os dados estejam chegando ao serviço da AWS desejado.

Para obter um exemplo de como estender o`DirectorySource`Para implementar a análise de um formato de log, consulte`Amazon.KinesisTap.Uls\UlsSourceFactory.cs`e`Amazon.KinesisTap.Uls\UlsLogParser.cs`no código-fonte do Kinesis Agent para Windows.

Para obter um exemplo de como criar uma origem que forneça a funcionalidade de marcação, consulte`Amazon.KinesisTap.Windows\WindowsSourceFactory.cs`e`Amazon.KinesisTap.Windows\EventLogSource.cs`no código-fonte do Kinesis Agent para Windows.

Implementação de pias de plug-in do Kinesis Agent para Windows

Siga estas etapas para implementar um coletor de plug-ins do Kinesis Agent para Windows.

Para criar um coletor de plugins do Kinesis Agent para Windows

1. Adicione uma classe ao projeto definido anteriormente que implemente a interface `IEventSink`.

Por exemplo, o código a seguir implementa um coletor que não faz nada que não seja registrar a chegada de registros que depois são descartados.

```
using Amazon.KinesisTap.Core;
using Microsoft.Extensions.Logging;

namespace MyCompany.MySinks
{
    public class NullSink : EventSink
    {
        public NullSink(IPlugInContext context) : base(context)
        {
        }

        public override void OnNext(IEnvelope envelope)
        {
            _logger.LogInformation($"Null sink {Id} received
 {GetRecord(envelope)}.");
        }

        public override void Start()
        {
            _logger.LogInformation($"Null sink {Id} starting.");
        }

        public override void Stop()
        {
            _logger.LogInformation($"Null sink {Id} stopped.");
        }
    }
}
```

Nesse exemplo, a classe de coletor `NullSink` é herdada da classe `EventSink`, pois ela fornece a capacidade de transformar registros em diferentes formatos de serialização, como JSON e XML.

2. Compile o projeto definido anteriormente.
3. Copie o conjunto no diretório de instalação do Kinesis Agent para Windows.
4. Crie ou atualize um`appsettings.json`que use o novo coletor e coloque-o no diretório de instalação do Kinesis Agent para Windows. Por exemplo, para usar os plug-ins personalizados

`RandomSource` e `NullSink`, você pode usar o seguinte arquivo de configuração `appsettings.json`:

```
{
  "Sources": [
  {
 "Id": "MyRandomSource",
 "SourceType": "RandomSource",
 "Rate": "00:00:10",
 "Max": 50
  }

  ],
  "Sinks": [
  {
    "Id": "MyNullSink",
 "SinkType": "NullSink",
 "Format": "json"
  }
  ],
  "Pipes": [
    {
   "Id": "MyRandomToNullPipe",
   "SourceRef": "MyRandomSource",
   "SinkRef": "MyNullSink"
 }
  ]
}
```

Essa configuração cria uma origem que envia uma instância de `RandomData` com um `RandomValue` definido como um número aleatório entre 0 e 50 a cada 10 segundos. Ele cria um coletor que transforma as instâncias `RandomData` de entrada em JSON, registra esse JSON e descarta as instâncias. Certifique-se de incluir os dois exemplos de fábricas, a classe de origem `RandomSource` e a classe de coletor `NullSink` no projeto definido anteriormente para usar o arquivo de configuração de exemplo.

5. Interrompa e inicie o Kinesis Agent para Windows.
6. Verifique o arquivo de log atual do Kinesis Agent para Windows (geralmente localizado na seção`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`) para garantir que não haja problemas com o plug-in de coletor personalizado.

7. Certifique-se de que os dados estejam chegando ao serviço da AWS desejado. Como o exemplo de `NullSink` não faz streaming para um serviço da AWS, você pode verificar o funcionamento correto do coletor procurando as mensagens de log que indicam que os registros foram recebidos.

   Por exemplo, você pode ver um arquivo de log semelhante ao seguinte:

```
2018-10-18 12:36:36.3647 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.AWS.AWSEventSinkFactory.
2018-10-18 12:36:36.4018 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Windows.PerformanceCounterSinkFactory.
2018-10-18 12:36:36.4018 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory MyCompany.MySinks.NullSinkFactory.
2018-10-18 12:36:36.6926 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Core.DirectorySourceFactory.
2018-10-18 12:36:36.6926 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.ExchangeSource.ExchangeSourceFactory.
2018-10-18 12:36:36.6926 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Uls.UlsSourceFactory.
2018-10-18 12:36:36.6926 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Windows.WindowsSourceFactory.
2018-10-18 12:36:36.6926 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory MyCompany.MySources.RandomSourceFactory.
2018-10-18 12:36:36.9601 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.Core.Pipes.PipeFactory.
2018-10-18 12:36:37.4694 Amazon.KinesisTap.Hosting.LogManager INFO Registered
 factory Amazon.KinesisTap.AutoUpdate.AutoUpdateFactory.
2018-10-18 12:36:37.4807 Amazon.KinesisTap.Hosting.LogManager INFO Performance
 counter sink  started.
2018-10-18 12:36:37.6250 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink starting.
2018-10-18 12:36:37.6250 Amazon.KinesisTap.Hosting.LogManager INFO Connected source
 MyRandomSource to sink MyNullSink
2018-10-18 12:36:37.6333 Amazon.KinesisTap.Hosting.LogManager INFO Random source id
 MyRandomSource started with rate 10000.
2018-10-18 12:36:47.8084 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":14}.
2018-10-18 12:36:57.6339 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":5}.
2018-10-18 12:37:07.6490 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":9}.
2018-10-18 12:37:17.6494 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":47}.
```

```
2018-10-18 12:37:27.6520 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":25}.
2018-10-18 12:37:37.6676 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":21}.
2018-10-18 12:37:47.6688 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":29}.
2018-10-18 12:37:57.6700 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":22}.
2018-10-18 12:38:07.6838 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":32}.
2018-10-18 12:38:17.6848 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":12}.
2018-10-18 12:38:27.6866 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":46}.
2018-10-18 12:38:37.6880 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":48}.
2018-10-18 12:38:47.6893 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":39}.
2018-10-18 12:38:57.6906 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":18}.
2018-10-18 12:39:07.6995 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":6}.
2018-10-18 12:39:17.7004 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":0}.
2018-10-18 12:39:27.7021 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":3}.
2018-10-18 12:39:37.7023 Amazon.KinesisTap.Hosting.LogManager INFO Null sink
 MyNullSink received {"RandomValue":19}.
```

Se você estiver criando um coletor que acesse os serviços da AWS, há classes básicas que podem ser úteis. Para um dissipador que usa o`AWSBufferedEventSink`classe base, consulte`Amazon.KinesisTap.AWS\CloudWatchLogsSink.cs`no código-fonte do Kinesis Agent para Windows.

Guia do Usuário do Histórico do Documento do Amazon Kinesis Agent para Microsoft Windows

Versão da API: 15/10/2018

A tabela a seguir descreve alterações noGuia do usuário do Amazon Kinesis Agent para Microsoft Windows(este documento).

update-history-change	update-history-description	update-history-date
Atualização importante da documentação	Adicionadas instruções para instalação do MSI. Configuração do Directory Source atualizada e adicionou WindowsEventLogPollingSource. Para configuração do coletor, adicionou configuração de sincronização do sistema de arquivos local; profileReshingawscredentialProvider; informações sobre decorações de texto, resolução de variáveis em atributos de coletor, configuração de endpoints regionais STS para coletores, configuração de endpoints de VPC e configuração de servidores proxy alternativos. Para pipes, atributos de configuração adicionados.	23 de fevereiro de 2021
Atualizar para a documentação	Tópico atualizado para indicar que as especificações do local do Amazon S3 fazem	7 de novembro de 2018

	distinção de maiúsculas e minúsculas.	
Lançamento inicial, versão 1.0.0.115	Primeira versão do Guia do Usuário do Kinesis Agent para Windows.	5 de novembro de 2018

Glossário da AWS

Para obter a terminologia mais recente da AWS, consulte o Glossário da AWS na Referência geral da AWS.

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.