aws

Guida per l'utente

Amazon Kinesis Agent per Microsoft Windows

Amazon Kinesis Agent per Microsoft Windows: Guida per l'utente

Table of Contents

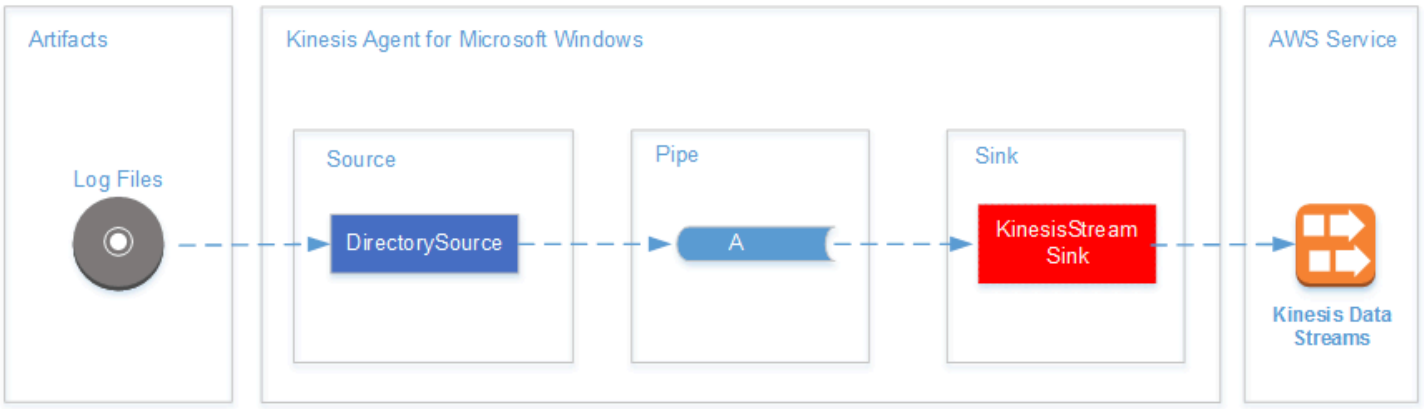
Cos'è Amazon Kinesis Agent per Microsoft Windows?

Amazon Kinesis Agent per Microsoft Windows (Kinesis Agent for Windows) è un agente ampliabile e configurabile. Questo servizio viene eseguito su parchi istanze di computer desktop e server Windows, in locale o nel Cloud AAWS. Kinesis Agent per Windows raccoglie, analizza, trasforma e invia log, eventi e parametri ai vari servizi AWS, inclusiKinesis Data Streams,Kinesis Data Firehose,Amazon CloudWatch, eLog di CloudWatch: .

Da questi servizi, è possibile archiviare, analizzare e visualizzare i dati utilizzando un'ampia gamma di altri servizi AWS, tra cui i seguenti:

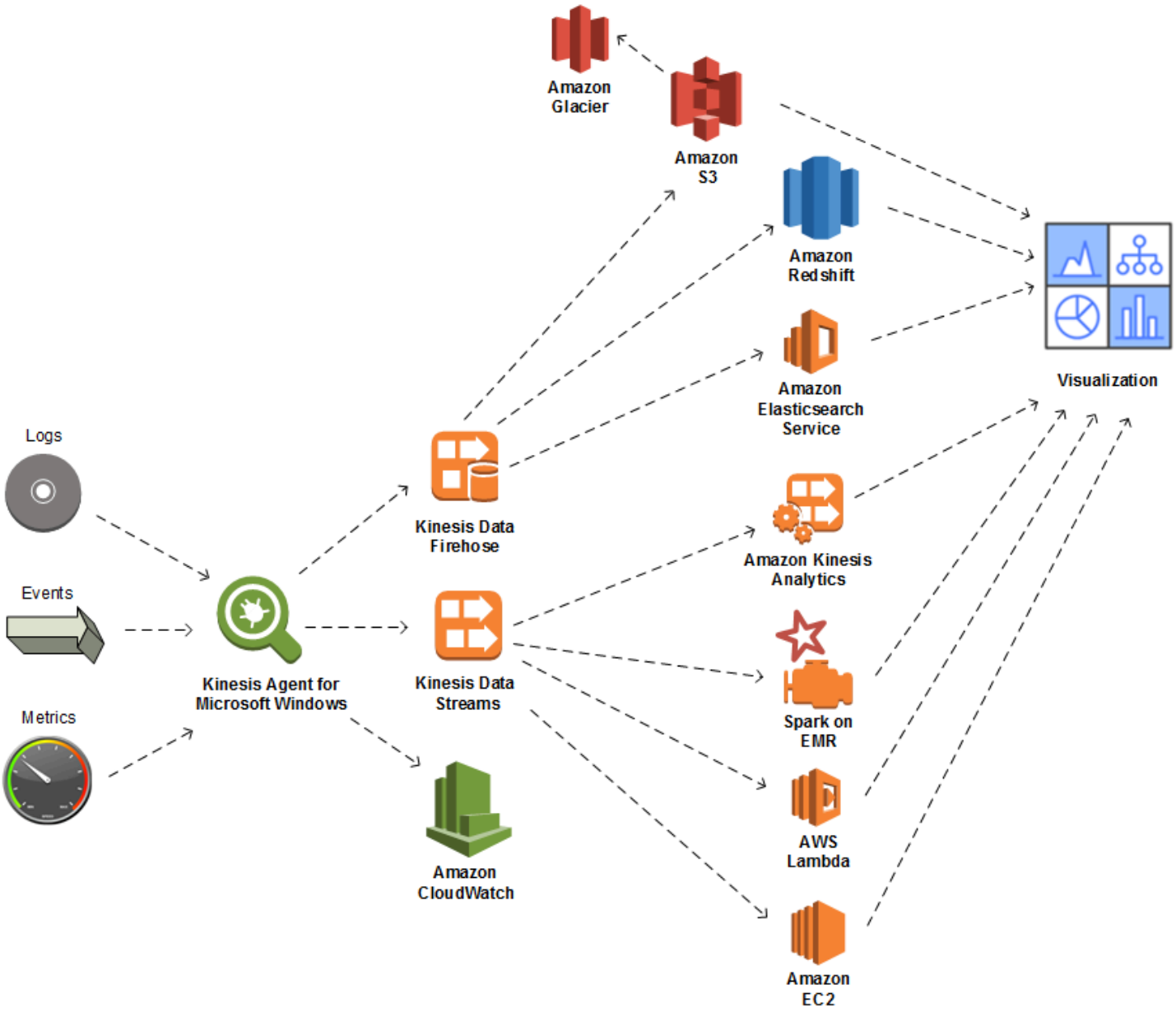
- Amazon Simple Storage Service (Amazon S3)
- Amazon Redshift
- Amazon Elasticsearch Service (Amazon ES)
- Analisi di Kinesis Data Analytics
- Amazon QuickSight
- Amazon Athena
- Kibana

Il seguente diagramma illustra una semplice configurazione di Kinesis Agent per Windows che invia file di log ai Kinesis Data Streams.



Per ulteriori informazioni su sorgenti, pipe e sink, consulta Amazon Kinesis Agent per Microsoft Windows Concetti.

Il seguente diagramma illustra alcuni dei modi in cui è possibile creare pipeline personalizzate di dati in tempo reale tramite framework di elaborazione dello streaming. Questi framework includono Kinesis Data Analytics, Apache Spark su Amazon EMR e AWS Lambda.



Argomenti

- Informazioni su AWS
- Cosa è possibile fare con Kinesis Agent per Windows?
- Benefits
- Guida introduttiva a Kinesis Agent per Windows

Informazioni su AWS

Amazon Web Services (AWS) è una raccolta di servizi infrastrutturali digitali utilizzabile dagli sviluppatori per elaborare le proprie applicazioni. I servizi includono elaborazione, archiviazione, database, analisi e sincronizzazione delle applicazioni (messaggistica e accodamento). AWS impiega un modello di servizi basato sul consumo. Ti vengono addebitati solo i servizi utilizzati da te o dalle tue applicazioni. Inoltre, al fine di rendere i propri servizi più accessibili per realizzare prototipi e sperimentare, AWS offre un piano di utilizzo gratuito. In questo piano, i servizi sono gratuiti al di sotto di un determinato livello di utilizzo. Per ulteriori informazioni sui costi AWS e suiPiano gratuitoConsultare.Centro risorse per le nozioni di base: . Per creare un account AWS, apri la Homepage AWS e accedi.

Cosa è possibile fare con Kinesis Agent per Windows?

Kinesis Agent per Windows offre le seguenti caratteristiche e funzionalità:

Raccogliere log, eventi e dati dei parametri

Kinesis Agent per Windows raccoglie, analizza, trasforma e invia log, eventi e parametri da parchi istanze di server e desktop a uno o più servizi AWS. Il payload ricevuto dai servizi può essere in un formato diverso dal supporto originale. Ad esempio, un log potrebbe essere archiviato in un determinato formato testuale (ad esempio formato syslog) su un server. Kinesis Agent per Windows è in grado di raccogliere e analizzare tale testo e, facoltativamente, trasformarlo in formato JSON, per esempio, prima dello streaming ad AWS. Ciò facilita un'elaborazione più semplice da parte di alcuni servizi AWS che utilizzano JSON. I dati in streaming verso Kinesis Data Streams possono essere continuamente elaborati da Kinesis Data Analytics per generare ulteriori parametri e parametri aggregati che a loro volta possono alimentare pannelli di controllo in tempo reale. È possibile archiviare i dati utilizzando un'ampia gamma di servizi AWS (ad esempio Amazon S3), in funzione del modo in cui i dati vengono utilizzati in una pipeline di dati in downstream.

Integrato con i servizi AWS

È possibile configurare Kinesis Agent per Windows per l'invio di file di log, eventi e parametri a diversi servizi AWS:

- Kinesis Data Firehose— Archivia facilmente i dati in streaming in Amazon S3, Amazon Redshift, Amazon ES oSplunkPer ulteriori analisi su.
- Kinesis Data Streams— Elabora i dati in streaming utilizzando applicazioni personalizzate in hosting in Kinesis Data Analytics o Apache Spark suAmazon EMR: . Oppure usa il codice personalizzato in esecuzione suAmazon EC2istanze o funzioni serverless personalizzate in esecuzione inAWS Lambda: .
- CloudWatch: visualizza i parametri in streaming sotto forma di grafico che possono essere abbinati in pannelli di controllo. Quindi imposta gli allarmi CloudWatch che vengono attivate da valori di parametri che violano le soglie predefinite.
- Log di CloudWatchArchivia log ed eventi in streaming, visualizzali e cercali nella Console di gestione AWS oppure elaborali più a valle in una pipeline di dati.

Istallazione e configurazione in tempi brevi

È possibile installare e configurare Kinesis Agent per Windows in pochi passaggi. Per ulteriori informazioni, consulta Installazione di Kinesis Agent per Windows e Configurazione di Amazon Kinesis Agent per Microsoft Windows. Un file di configurazione dichiarativa semplice specifica quanto segue:

- Le origini e i formati di log, eventi e parametri da raccogliere.
- Le trasformazioni da applicare ai dati raccolti. Possono essere inclusi dati aggiuntivi e i dati esistenti possono essere trasformati e filtrati.
- Le destinazioni in cui gli ultimi dati vengono trasmessi e buffering, sharding e formato per i payload di streaming.

Kinesis Agent per Windows viene fornito con parser integrati per i file di log generati da servizi aziendali Microsoft comuni, ad esempio:

- Microsoft Exchange
- SharePoint
- Controller di dominio Active Directory
- Server DHCP

Nessuna amministrazione manuale

Kinesis Agent per Windows si adatta automaticamente alle varie situazioni senza perdere dati. Questi includono la rotazione di log, il ripristino dopo il riavvio e la rete temporanea o le interruzioni del servizio. È possibile configurare Kinesis Agent per Windows per aggiornare automaticamente alle nuove versioni. Non è necessario l'intervento dell'operatore in nessuno di questi casi.

Ampliare utilizzando architetture aperte

Se le funzionalità dichiarative e i plug-in integrati sono sufficienti per il monitoraggio di sistemi desktop o server, è possibile estendere Kinesis Agent per Windows creando plug-in. I nuovi plug-in abilitano nuove origini e destinazioni per i log, gli eventi e i parametri. Il codice sorgente per Kinesis Agent per Windows è disponibile alla paginahttps://github.com/awslabs/kinesis-agent-windows: .

Benefits

Kinesis Agent per Windows esegue la raccolta dati iniziale, la trasformazione e lo streaming per log, eventi e parametri per le pipeline di dati. Creare queste pipeline di dati ha numerosi vantaggi:

Analisi e visualizzazione

L'integrazione di Kinesis Agent per Windows con Kinesis Data Firehose e le funzionalità di trasformazione semplificano l'integrazione con diversi servizi di visualizzazione e analisi:

- Amazon QuickSight— Un servizio di business intelligence basato sul cloud può acquisire diverse origini. Kinesis Agent per Windows può trasformare i dati e trasmetterli in streaming su Amazon S3 e Amazon Redshift tramite Kinesis Data Firehose. Questo processo consente il rilevamento di insight approfonditi derivati dai dati utilizzando le visualizzazioni Amazon QuickSight.
- Athena— Un servizio di query interattivo che consente l'interrogazione dei dati basata su SQL. Kinesis Agent per Windows può trasformare e trasmettere dati su Amazon S3 tramite Kinesis Data Firehose. Athena può quindi interattivamente eseguire query SQL sui dati per ispezionare e analizzare log ed eventi in tempi rapidi.
- Kibana— Uno strumento di visualizzazione dei dati open source. Kinesis Agent per Windows può trasformare e trasmettere i dati ad Amazon ES tramite Kinesis Data Firehose. È quindi possibile utilizzare Kibana per analizzare i dati. Creare e aprire visualizzazioni differenti, tra cui istogrammi, grafici a linee, grafici a torta, mappe termiche e grafici geospaziali.

Security

Una pipeline di analisi di dati relativi a eventi e log che include Kinesis Agent per Windows è in grado di rilevare e di inviare notifiche su violazioni della sicurezza nelle organizzazioni che possono bloccare o interrompere eventuali attacchi.

Prestazioni delle applicazioni

Kinesis Agent per Windows è in grado di raccogliere i log, gli eventi e i dati dei parametri relativi alle prestazioni di applicazioni o servizi. Una pipeline completa di dati può quindi analizzare queste informazioni. Questa analisi consente di migliorare le prestazioni e l'affidabilità di servizi e applicazioni rilevando e creando report su difetti che altrimenti potrebbero non essere evidenti. Ad esempio, è possibile rilevare notevoli modifiche nei tempi di esecuzione delle chiamate API del

servizio. Quando correlata a una distribuzione, questa funzionalità consente di individuare e risolvere nuovi problemi di prestazioni di servizi di proprietà dell'utente.

Operazioni di servizio

Una pipeline di dati è in grado di analizzare i dati raccolti per prevedere problematiche potenziali e fornire informazioni dettagliate su come evitare interruzioni del servizio. Ad esempio, è possibile analizzare i log, gli eventi e i parametri per stabilire l'utilizzo delle capacità attuale e previsto in modo da impiegare una capacità aggiuntiva prima che gli utenti subiscano un disservizio. Se si verifica un'interruzione del servizio, è possibile analizzare i dati per stabilire l'impatto sui clienti durante il periodo di interruzione.

Auditing

Una pipeline di dati è in grado di elaborare i log, gli eventi e i parametri che Kinesis Agent per Windows raccoglie e trasforma. È quindi possibile controllare questi dati elaborati utilizzando diversi servizi AWS. Ad esempio, Kinesis Data Firehose potrebbe ricevere un flusso di dati da Kinesis Agent per Windows che memorizza i dati in Amazon S3. È quindi possibile controllare questi dati eseguendo query SQL interattive utilizzando Athena.

Archiving

Spesso i dati operativi più importanti sono dati raccolti di recente. Tuttavia, l'analisi dei dati raccolti sulle applicazioni e i servizi su più anni può essere utile, ad esempio, per una pianificazione a lunga distanza. Conservare grandi quantità di dati può essere costoso. Kinesis Agent per Windows è in grado di raccogliere, trasformare e archiviare dati in Amazon S3 tramite Kinesis Data Firehose. Pertanto,Amazon S3 Glacierè disponibile per ridurre i costi di archiviazione dei dati meno recenti.

Alerting

Kinesis Agent per Windows trasmette le metriche a CloudWatch. A sua volta, è possibile creare allarmi CloudWatch per l'invio di una notifica tramiteSimple Notification Service (Amazon SNS)quando una metrica viola costantemente una soglia specifica. In questo modo gli ingegneri hanno una migliore consapevolezza dei problemi operativi di applicazioni e servizi.

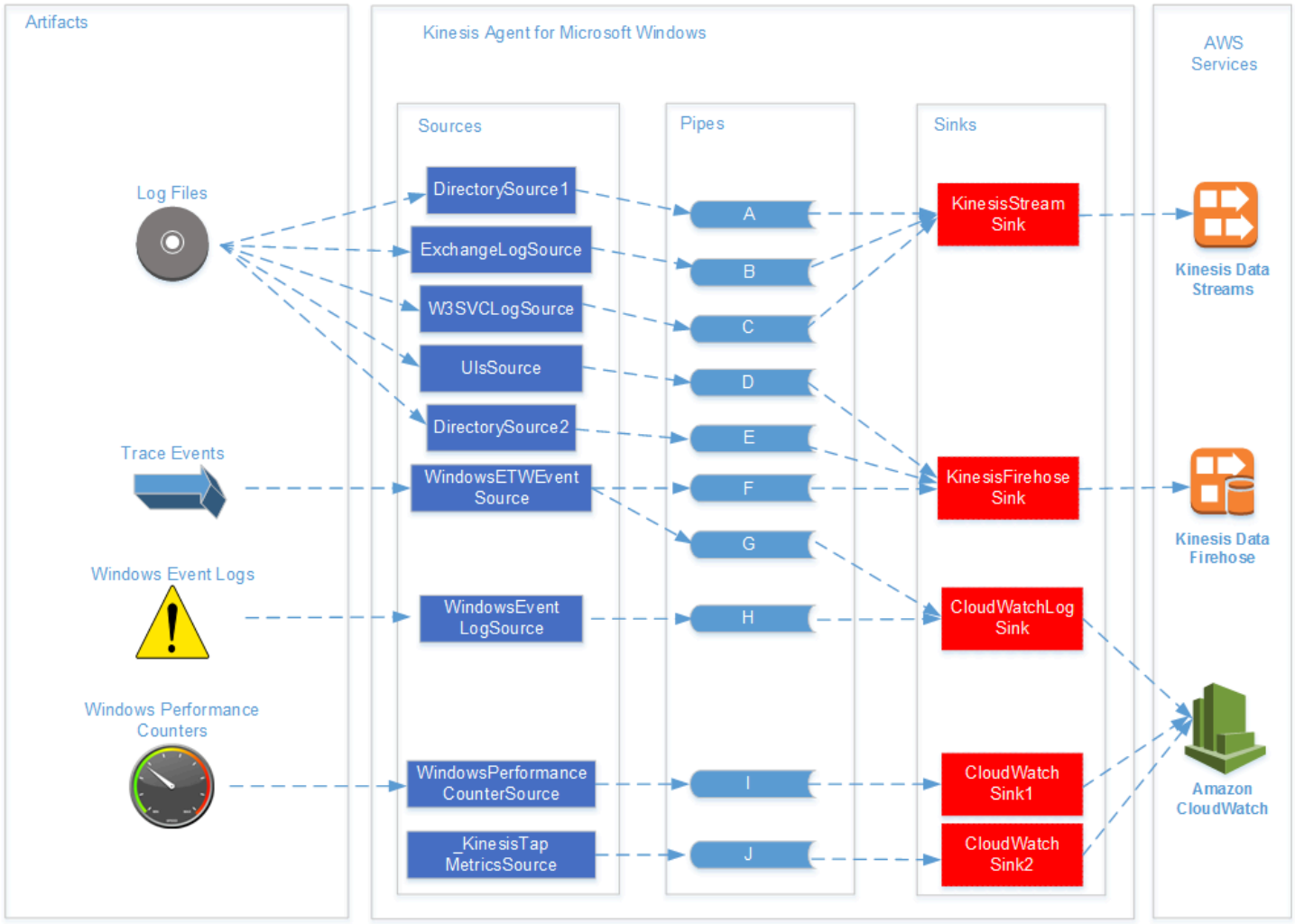
Guida introduttiva a Kinesis Agent per Windows

Per ulteriori informazioni su Kinesis Agent per Windows, ti consigliamo di iniziare con le seguenti sezioni:

- Amazon Kinesis Agent per Microsoft Windows Concetti
- Nozioni di base su Amazon Kinesis Agent per Microsoft Windows

Amazon Kinesis Agent per Microsoft Windows Concetti

Comprendere i concetti fondamentali di Amazon Kinesis Agent per Microsoft Windows (Kinesis Agent per Windows) può semplificare la trasmissione e la raccolta di dati su desktop e parchi di server al resto dei componenti della pipeline di dati per l'elaborazione.



Questo diagramma di una pipeline di dati illustra i seguenti componenti e processi:

Server e desktop dispongono di elementi quali file di registro, eventi e metriche raccolti da uno o più Kinesis Agent per Windowssources: . I dati possono essere facoltativamente trasformati da, per esempio, un formato di testo di un file flat in un oggetto.

I dati (in forma oggetto o testo) possono quindi fluire in uno o più Kinesis Agent per WindowsPipe: . Una pipe collega un'origine a un agente Kinesis per Windowssink: . La pipe può filtrare i dati non necessari.

Un sink può trasformare i dati analizzati in oggetti in formato JSON o XML. Il sink invia i dati a un servizio AWS specifico, ad esempio Kinesis Data Streams, Kinesis Data Firehose o Amazon CloudWatch.

Utilizzando più pipe, una singola sorgente può inviare gli stessi dati a più sink (per esempio, guarda le pipe F e G nel diagramma). Utilizzando più pipe, più sorgenti possono inviare i dati a un unico sink (per esempio, guarda le pipe A, B e C nel diagramma). È inoltre possibile utilizzare più pipe per lo streaming dei dati da più sink a più origini. Origini, sink e pipe hanno i tipi e non ci può essere più di una sorgente, un sink o una pipe dello stesso tipo.

Per esempi di file di configurazione che dichiarano origini, pipe e sink, consulta Esempi di configurazione di Kinesis Agent per Windows.

Argomenti

Pipeline di dati

AData Pipelineviene utilizzato per raccogliere, elaborare, visualizzare ed eventualmente generare allarmi per applicazioni e servizi. Kinesis Agent per Windows si inserisce all'inizio nelle pipeline di dati, in cui registri, eventi e metriche vengono raccolti da flotte di computer desktop o server. Kinesis Agent per Windows trasmette in streaming i dati raccolti ai vari servizi AWS che formano il resto pipeline di dati. Una pipeline di dati ha uno scopo, ad esempio visualizzare lo stato di un determinato servizio in tempo reale per aiutare gli ingegneri a utilizzare il servizio in modo più efficiente. Una pipeline di dati dello stato del servizio può eseguire una qualsiasi delle operazioni seguenti:

- Avvisare i tecnici di problemi prima che questi problemi influenzino l'esperienza per i clienti dei servizi.
- Aiutare i tecnici a gestire in modo efficiente i costi del servizio mostrando i trend di utilizzo delle risorse. Questi trend consentono di regolare i livelli di risorse in modo appropriato o anche di implementare scenari di scalabilità automatica.
- Approfondire la causa principale dei problemi rilevati dai clienti del servizio. Questa operazione consente di velocizzare la risoluzione dei problemi e di ridurre i costi di supporto.

Per un esempio dettagliato di come costruire una pipeline di dati utilizzando Kinesis Agent per Windows, consultaTutorial: Trasmetti file di registro JSON ad Amazon S3 utilizzando Kinesis Agent per Windows: .

Sources

Un agente Kinesis per Windowssource (origine)raccoglie registri, eventi o metriche. Una sorgente raccoglie un determinato tipo di dati da un determinato produttore di tali dati in base al tipo di origine. Ad esempio, il tipo `DirectorySource` raccoglie i file di log da directory specifiche nel file system. Se i dati non sono già strutturati (come con alcuni tipi di file di log), un'origine può essere utile per analizzare la rappresentazione testuale in alcuni moduli strutturati. Ogni sorgente corrisponde a un determinatoDichiarazione di originenell'agente Kinesis per Windows`appsettings.json`File di configurazione. La dichiarazione di origine fornisce i dettagli essenziali per la configurazione dell'origine e la sua personalizzazione in base ai requisiti di raccolta dei dati specifici. I tipi di dettagli che possono essere configurati variano a seconda del tipo di origine. Ad esempio, il tipo di origine `DirectorySource` richiede le specifiche della directory in cui si trovano i file di log.

Per ulteriori informazioni sui tipi e sulle dichiarazioni delle origini, consulta Dichiarazioni delle origini.

Sinks

Un agente Kinesis per WindowssinkComprende i dati raccolti da un'origine Kinesis Agent per Windows e trasmette in streaming tali dati a uno dei diversi servizi AWS che formano il resto pipeline di dati. Ogni sink corrisponde a un particolareDichiarazione dei sinknell'agente Kinesis per Windows`appsettings.json`File di configurazione. La dichiarazione del sink fornisce i dettagli essenziali per la configurazione del sink e la sua personalizzazione in base a requisiti di streaming dei dati specifici. I tipi di dettagli che possono essere configurati variano a seconda del tipo di sink. Ad esempio, alcuni tipi di sink consentono una dichiarazione di sink per specificare una determinata serializzazione `Format` per i dati forniti. Quando questa opzione viene specificata nella dichiarazione di sink, la serializzazione dei dati raccolti si verifica prima dello streaming dei dati al servizio AWS associato al sink.

Per ulteriori informazioni sui tipi e sulle dichiarazioni dei sink, consulta Dichiarazioni dei sink.

Pipes

Un agente Kinesis per WindowsPipecollega l'output di un'origine Kinesis Agent per Windows all'input di un sink Kinesis Agent per Windows. È possibile trasformare i dati mentre vengono trasmessi

attraverso la pipe. Ogni pipe corrisponde a una determinata dichiarazione di pipe nell'agente Kinesis per Windows`appsettings.json`File di configurazione. La dichiarazione di pipe fornisce i dettagli essenziali per la configurazione del sink, ad esempio l'origine e il sink per la pipe.

Per ulteriori informazioni sui tipi e sulle dichiarazioni delle pipe, consulta Dichiarazioni delle pipe.

Nozioni di base su Amazon Kinesis Agent per Microsoft Windows

È possibile usare Amazon Kinesis Agent per Microsoft Windows (Kinesis Agent per Windows) per raccogliere, analizzare, trasformare e trasmettere i log, gli eventi e i parametri dal tuo parco di Windows ai vari servizi AWS. Le informazioni seguenti contengono prerequisiti e istruzioni dettagliate per l'installazione e la configurazione di Kinesis Agent per Windows.

Argomenti

Prerequisites

Prima di installare Kinesis Agent per Windows, assicurati di disporre dei seguenti prerequisiti:

- Familiarità con i concetti Kinesis Agent per Windows. Per ulteriori informazioni, consulta Amazon Kinesis Agent per Microsoft Windows Concetti.
- Un account AWS per l'utilizzo di vari servizi AWS correlati alla pipeline di dati. Per ulteriori informazioni sulla creazione e la configurazione di un account AWS, consulta Configurazione di un account AWS.
- Microsoft.NET Framework 4.6 o versioni successive su ogni desktop o server che eseguirà Kinesis Agent per Windows. Per ulteriori informazioni, vedere Installare .NET Framework per gli sviluppatori nella documentazione Microsoft .NET.

  Per determinare la versione più recente di .NET Framework installata su un desktop o su un server, utilizzare il seguente script 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
```

- I flussi in cui si desidera inviare i dati da Kinesis Agent per Windows (se usi Amazon Kinesis Data Streams). Creare i flussi utilizzando ilConsole Kinesis Data Streams, ilAWS CLI, oppureStrumenti AWS per Windows PowerShell: . Per ulteriori informazioni, consultaCreazione e aggiornamento dei flussi di datinellaAmazon Kinesis Data Streams: .
- I flussi di distribuzione Firehose in cui si desidera inviare i dati da Kinesis Agent per Windows (se usi Amazon Kinesis Data Firehose). Creare flussi di consegna utilizzando ilKinesis Data Firehose, ilAWS CLI, oppureStrumenti AWS per Windows PowerShell: . Per ulteriori informazioni, consulta la sezione relativa alla creazione di un flusso di distribuzione Amazon Kinesis Data Firehose nella Guida per sviluppatori Amazon Kinesis Data Firehose.

Configurazione di un account AWS

Se non hai un account AWS, completa la procedura seguente per crearne uno.

Registrazione per creare un account AWS

1. Aprire la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
2. Seguire le istruzioni online.

   Come parte della procedura di registrazione riceverai una telefonata, durante la quale dovrai inserire un codice di verifica sulla tastiera del telefono.

Per creare un utente amministratore per se stessi e aggiungere l'utente a un gruppo di amministratori (console)

1. Accedere alla console IAM come proprietario dell'account scegliendo Root user (Utente root) e immettendo l'indirizzo email dell'account AWS. Nella pagina successiva, inserisci la password.

   > **Note**
   >
   > È fortemente consigliato rispettare la best practice sull'utilizzo dell'**`Administrator`**Utente IAM che segue e conservare in un luogo sicuro le credenziali dell'utente root. Accedere come utente root solo per eseguire alcune attività di gestione dell'account e del servizio.

2. Nel riquadro di navigazione selezionare Users (Utenti), quindi selezionare Add user (Aggiungi utente).
3. In User name (Nome utente), immettere **`Administrator`**.
4. Selezionare la casella di controllo accanto a AWS Management Console access (Accesso a Console di gestione AWS). Quindi, selezionare Custom password (Password personalizzata)e immettere la nuova password nella casella di testo.
5. (Facoltativo) Per impostazione predefinita, AWS richiede che il nuovo utente crei una nuova password al primo accesso. Puoi deselezionare la casella di controllo accanto a User must create a new password at next sign-in (L'utente deve creare una nuova password al prossimo accesso) per consentire al nuovo utente di reimpostare la propria password dopo aver effettuato l'accesso.
6. ScegliereSuccessivo: Autorizzazioni.
7. In Set permissions (Imposta autorizzazioni), selezionare Add user to group (Aggiungi l'utente al gruppo).
8. Seleziona Create group (Crea gruppo).
9. Nella finestra di dialogo Create group (Crea gruppo), per Group name (Nome gruppo) immettere **`Administrators`**.
10. SceglierePolicy di filtroe quindi selezionareAWS gestito - funzione di lavoroper filtrare il contenuto della tabella.
11. Nell'elenco delle policy, selezionare la casella di controllo accanto ad AdministratorAccess. Seleziona quindi Create group (Crea gruppo).

    > **Note**
    >
    > È necessario attivare l'accesso utente e ruolo IAM alla fatturazione prima di poter utilizzare le autorizzazioni `AdministratorAccess` per accedere alla console Fatturazione e gestione costi AWS. A questo scopo, seguire le istruzioni nella fase 1 del tutorial sulla delega dell'accesso alla console di fatturazione.

12. Nell'elenco dei gruppi seleziona la casella di controllo per il tuo nuovo gruppo. Se necessario, selezionare Refresh (Aggiorna) per visualizzare il gruppo nell'elenco.
13. ScegliereSuccessivo: Tags: .
14. (Facoltativo) Aggiungere metadati all'utente collegando i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo di tag in IAM, consultare Tagging di utenti e ruoli IAM nella Guida per l'utente di IAM.

15. ScegliereSuccessivo: Review (Revisione)Per visualizzare l'elenco delle appartenenze ai gruppi da aggiungere al nuovo utente. Quando sei pronto per continuare, seleziona Create user (Crea utente).

È possibile utilizzare questa stessa procedura per creare altri gruppi e utenti e concedere agli utenti l'accesso alle risorse dell'account AWS. Per ulteriori informazioni sull'utilizzo di policy per limitare le autorizzazioni degli utenti alle risorse AWS, consulta Gestione degli accessi e Esempi di policy.

Per effettuare la registrazione ad AWS e creare un account amministratore

1. Se non disponi di un account AWS, aprihttps://aws.amazon.com/: . Selezionare Create an AWS Account (Crea un account AWS), quindi seguire le istruzioni online.

   Come parte della procedura di registrazione si riceverà una telefonata, durante la quale si dovrà inserire un PIN usando la tastiera del telefono.
2. Accedere alla Console di gestione AWS e aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
3. Nel riquadro di navigazione, scegliere Groups (Gruppi), quindi Create New Group (Crea nuovo gruppo).
4. In Group Name (Nome gruppo), digitare un nome per il gruppo, come **`Administrators`** , quindi scegliere Next Step (Fase successiva).
5. Nell'elenco delle policy, seleziona la casella di controllo accanto alla policy AdministratorAccess. È possibile utilizzare il menu Filter (Filtro) e la casella Search (Cerca) per filtrare l'elenco di policy.
6. Selezionare Next Step (Fase successiva). Scegliere Create Group (Crea gruppo) e il tuo nuovo gruppo appare sotto Group Name (Nome gruppo).
7. Nel riquadro di navigazione, seleziona Users (Utenti), quindi scegli Create New Users (Crea nuovi utenti).
8. Nella casella 1, immettere un nome utente, deselezionare la casella di controllo accanto a Generate an access key for each user (Generare una chiave di accesso per ogni utente), quindi scegliere Create (Crea).
9. Nell'elenco di utenti, selezionare il nome (non la casella di controllo) dell'utente appena creato. È possibile utilizzare la casella Search (Cerca) per cercare il nome utente.
10. Selezionare la scheda Groups (Gruppi), quindi scegliere Add Users to Group (Aggiungi utenti al gruppo).

11. Selezionare la casella di controllo accanto al gruppo di amministratori, quindi selezionare Add to Groups (Aggiungi ai gruppi).
12. Seleziona la scheda Security Credentials (Credenziali di sicurezza). In Sign in Credentials (Credenziali di accesso), seleziona Manage Password (Gestisci password).
13. Selezionare Assign a custom password (Assegna una password personalizzata), immettere una password nelle caselle Password e Confirm Password (Conferma password), quindi selezionare Apply (Applica).

Installazione di Kinesis Agent per Windows

Sono disponibili tre modi per installare Kinesis Agent per Windows su Windows:

- Installare utilizzando MSI (un pacchetto di installazione di Windows).
- Installare daAWS Systems Manager, un set di servizi per l'amministrazione di server e desktop.
- Esecuzione di uno script PowerShell.

> **Note**
>
> Le seguenti istruzioni occasionalmente usano i termini KinesisTap e AWSKinesisTap. Queste parole significano la stessa cosa di Kinesis Agent per Windows, ma è necessario specificarle così come sono durante l'esecuzione di queste istruzioni.

Installare Kinesis Agent per Windows utilizzando MSI

È possibile scaricare il pacchetto Kinesis Agent per Windows MSI dallarepository kinesis-agent-windows su GitHub: . Dopo aver scaricato MSI, utilizzare Windows per avviarlo e seguire le istruzioni del programma di installazione. Dopo l'installazione, è possibile disinstallare come qualsiasi applicazione Windows.

In alternativa, è possibile utilizzare l'msiexecDal prompt dei comandi di Windows per l'installazione automatica, attivare la registrazione e disinstallare come illustrato negli esempi seguenti. Replace (Sostituisci)*AWSKinesisTap.1.1.216.4.msi* `with the appropriate version of Kinesis Agent for Windows for your application.`

Per installare Kinesis Agent per Windows in modo invisibile:

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

Per registrare i messaggi di installazione per la risoluzione dei problemi in un file denominato***logfile.log***:

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

Per disinstallare Kinesis Agent per Windows utilizzando il prompt dei comandi:

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

Installare Kinesis Agent per Windows utilizzando AWS Systems Manager

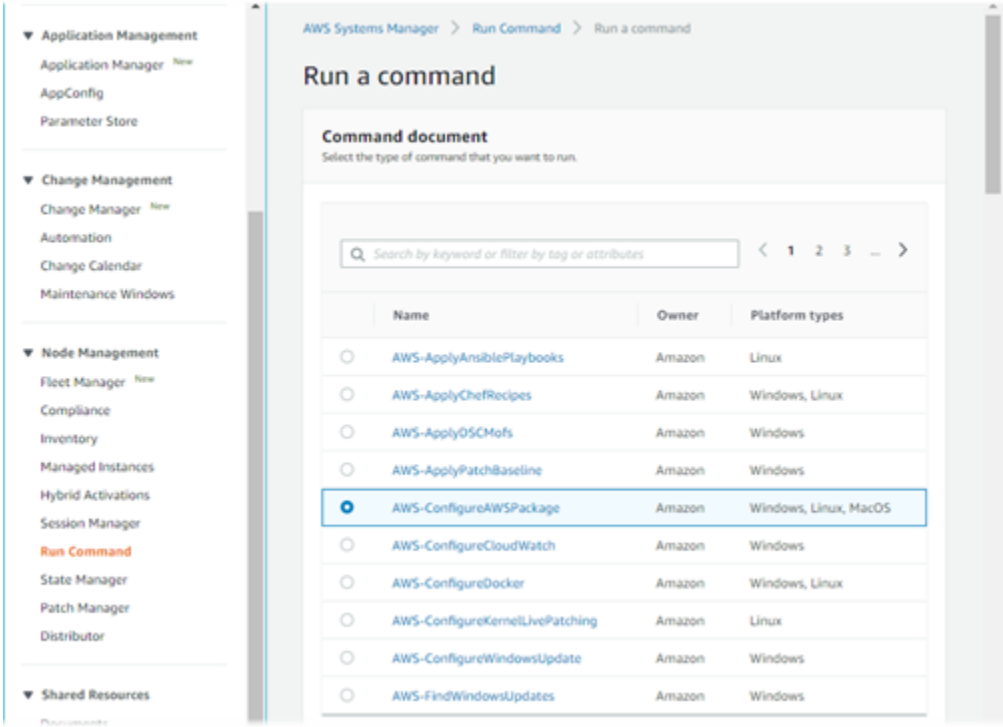
Segui questi passaggi per installare Kinesis Agent per Windows utilizzando il Run Command di Systems Manager. Per ulteriori informazioni su Run Command, consultaAWS Systems ManagernellaAWS Systems Manager: . Oltre a utilizzare Systems Manager Esegui Command, è anche possibile utilizzare Systems ManagerIntervallo di manutenzioneeState Managerper automatizzare la distribuzione di Kinesis Agent per Windows nel tempo.

> **Note**
>
> L'installazione di Systems Manager per Kinesis Agent per Windows è disponibile nelle regioni AWS elencate inAWS Systems ManagerAd eccezione di quanto segue:
>
> - cn-north-1
> - cn-northwest-1
> - Tutte le regioni AWS GovCloud.

Per installare Kinesis Agent per Windows tramite Systems Manager

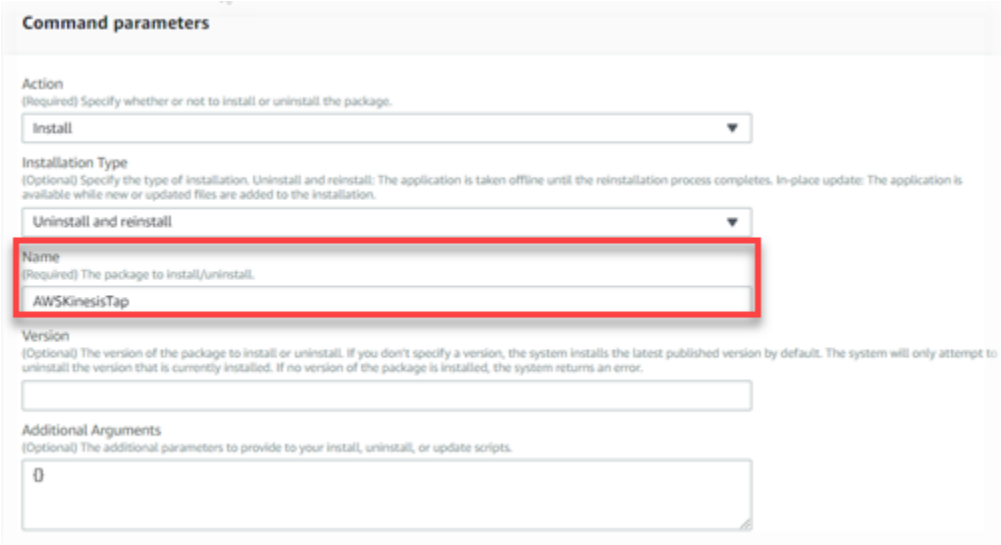
1. Verificare che la versione 2.58.0 o successiva dell'Agente SSM sia installata sulle istanze in cui si desidera installare Kinesis Agent per Windows. Per ulteriori informazioni, consultaInstallazione e configurazione dell'Agente SSM sulle istanze WindowsnellaAWS Systems Manager: .
2. Apri la console AWS Systems Manager all'indirizzohttps://console.aws.amazon.com/systems-manager/: .

3. Dal riquadro di navigazione, inGestione nodi, scegliereRun Command die quindi scegliereRun Command di: .
4. DalDocumento di comando, selezionare la casella di controlloAWS-ConfigureAWSPackagenon valido.



5. UNDERParametri di comando, perNome, immettereAwskineSap: . Lasciare altre impostazioni ai valori predefiniti.

> **Note**
>
> esciVersionvuoto per specificare la versione più recente del pacchetto AWSkinesISTap. Facoltativamente, è possibile inserire una versione specifica da installare.



6. UNDERTargetsSpecificare le istanze su cui eseguire il comando. È possibile scegliere di specificare le istanze in base ai tag associati alle istanze, scegliere le istanze manualmente oppure specificare un gruppo di risorse che includa le istanze.
7. Lasciare tutte le altre impostazioni ai valori predefiniti e selezionareEsegui: .

Installare Kinesis Agent per Windows utilizzando PowerShell

Utilizzare un editor di testo per copiare i seguenti comandi in un file e salvarlo come script PowerShell. Usiamo`InstallKinesisAgent.ps1`nell'esempio seguente.

```
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
}
```

Aprire una finestra di prompt di comandi elevati. Nella directory in cui è stato scaricato il file, utilizzare il seguente comando per eseguire lo script:

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

Per installare una versione specifica di Kinesis Agent per Windows, aggiungi la`-version`Opzione:

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

Replace (Sostituisci)*`version`*con un numero di versione valido di Kinesis Agent per Windows. Per informazioni sulla versione, consulta larepository kinesis-agent-windows su GitHub: .

Esistono molti strumenti di distribuzione che possono eseguire in remoto script di PowerShell. Possono essere utilizzati per automatizzare l'installazione di Kinesis Agent per Windows su parchi di server o desktop.

Configurazione e avvio di Kinesis Agent per Windows

Dopo aver installato Kinesis Agent per Windows, è necessario configurare e avviare l'agente. Dopodiché, non dovrebbero essere necessari ulteriori interventi.

Per configurare e avviare Kinesis Agent per Windows

1. Creare e distribuire un file di configurazione dell'Agente Kinesis per Windows. Questo file configura origini, sink e pipe, insieme ad altri elementi di configurazione globale.

   Per ulteriori informazioni sulla configurazione di Kinesis Agent per Windows, consultaConfigurazione di Amazon Kinesis Agent per Microsoft Windows: .

Per esempi completi di file di configurazione che è possibile personalizzare e installare, consulta Esempi di configurazione di Kinesis Agent per Windows.

2. Aprire una finestra del prompt dei comandi di PowerShell e avviare Kinesis Agent per Windows utilizzando il seguente comando PowerShell:

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

Configurazione di Amazon Kinesis Agent per Microsoft Windows

Prima di avviare Amazon Kinesis Agent per Microsoft Windows, è necessario creare un file di configurazione e distribuirlo. Il file di configurazione fornisce le informazioni necessarie per raccogliere, trasformare e trasmettere dati su server e computer desktop Windows ai vari servizi AWS. I file di configurazione definiscono set di origini, sink e pipe che collegano origini a sink, insieme a trasformazioni opzionali.

Il file di configurazione Kinesis Agent per Windows è denominato`appsettings.json`: . Distribuisci il file in `%PROGRAMFILES%\Amazon\AWSKinesisTap`.

Argomenti

Struttura configurazione di base

La struttura di base del file di configurazione di Amazon Kinesis Agent per Microsoft Windows è un documento in formato JSON con il seguente modello:

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

- Il valore di `Sources` è uno o più Dichiarazioni delle origini.
- Il valore di `Sinks` è uno o più Dichiarazioni dei sink.
- Il valore di `Pipes` è uno o più Dichiarazioni delle pipe.

Per ulteriori informazioni sui concetti di origine, pipe e sink di Kinesis Agent per Windows, consultaAmazon Kinesis Agent per Microsoft Windows Concetti: .

L'esempio seguente è un`appsettings.json`Configurare che configura l'agente Kinesis per Windows per lo streaming di eventi di log delle applicazioni Windows a 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"
    }
  ]
}
```

Per ulteriori informazioni su ciascun tipo di dichiarazione, consultare le sezioni seguenti:

- Dichiarazioni delle origini
- Dichiarazioni dei sink
- Dichiarazioni delle pipe

Distinzione tra lettere maiuscole e minuscole nella configurazione

I file in formato JSON effettuano in genere la distinzione tra lettere maiuscole e minuscole ed è necessario presumere che tutte le chiavi e i valori nei file di configurazione di Kinesis Agent

per Windows si comporta allo stesso modo. Alcune chiavi e valori nel file di configurazione `appsettings.json` non effettuano la distinzione tra lettere maiuscole e minuscole, ad esempio:

- Il valore della coppia chiave-valore `Format` per i sink. Per ulteriori informazioni, consulta Dichiarazioni dei sink.
- Il valore della coppia chiave-valore `SourceType` per le origini, la coppia chiave-valore `SinkType` per i sink e la coppia chiave-valore `Type` per pipe e plug-in.
- Il valore della coppia chiave-valore `RecordParser` per l'origine `DirectorySource`. Per ulteriori informazioni, consulta Configurazione DirectorySource.
- Il valore della coppia chiave-valore `InitialPosition` per le origini. Per ulteriori informazioni, consulta Configurazione del segnalibro.
- Prefissi per sostituzioni variabili. Per ulteriori informazioni, consulta Configurazione di sostituzione di variabili sink.

Dichiarazioni delle origini

In Amazon Kinesis Agent per Microsoft Windows,Dichiarazioni delle originidescrivono dove e quali dati di registro, evento e metrica devono essere raccolti. Inoltre, è possibile specificare le informazioni per l'analisi dei tali dati in modo che possano essere trasformati. Le seguenti sezioni descrivono le configurazioni per i tipi di origine integrati che sono disponibili in Kinesis Agent per Windows. Poiché Kinesis Agent per Windows è ampliabile, è possibile aggiungere tipi di origine personalizzati. Ogni tipo di origine di solito richiede coppie chiave-valore specifiche negli oggetti di configurazione che sono rilevanti per quel tipo di origine.

Tutte le dichiarazioni di origini devono contenere almeno le seguenti coppie chiave-valore:

`Id`

Una stringa univoca che identifica un determinato oggetto di origine all'interno del file di configurazione.

`SourceType`

Il nome del tipo di origine per questo oggetto origine. Il tipo di origine specifica l'origine dei dati di log, eventi o parametri raccolti da tale oggetto. Controlla anche quali altri aspetti dell'origine possono essere dichiarati.

Per esempi di file di configurazione completi che utilizzano diversi tipi di dichiarazioni di origini, vedi Streaming da diverse origini a Kinesis Data Streams.

Argomenti

Configurazione DirectorySource

Overview

Il tipo di origine `DirectorySource` raccoglie i log di file memorizzati nella directory specificata. Poiché i file di log sono disponibili in molti formati diversi, la dichiarazione `DirectorySource` consente di specificare il formato dei dati nel file di log. Quindi è possibile trasformare i contenuti dei log in un formato standard, ad esempio JSON o XML, prima di eseguire lo streaming a vari servizi AWS.

Di seguito è riportato un esempio della dichiarazione `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"
}
```

Tutte le dichiarazioni `DirectorySource` possono fornire le seguenti coppie chiave-valore:

`SourceType`

Deve essere una stringa letterale `"DirectorySource"` (obbligatoria).

`Directory`

Il percorso alla directory contenente i file di log (obbligatorio).

`FileNameFilter`

Eventualmente limita il set di file nella directory in cui i dati di log sono raccolti in base al modello di denominazione dei file jolly. Se si dispone di più modelli di nomi di file di registro, questa funzionalità consente di utilizzare un singolo`DirectorySource`, come mostrato nell'esempio seguente.

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

Gli amministratori di sistema a volte comprimere i file di registro prima di archiviarli. Se si specifica`"*.*"`in`FileNameFilter`, i file compressi noti sono ora esclusi. Questa funzione impedisce`.zip`,`.gz`, e`.bz2`di essere trasmessi accidentalmente. Se questa coppia chiave-valore non viene specificata, i dati provenienti da tutti i file nella directory sono raccolti per impostazione predefinita.

`IncludeSubdirectories`

Specifica di monitorare le sottodirectory a profondità arbitraria limitata dal sistema operativo. Questa funzione è utile per monitorare i server Web con più siti Web. È possibile utilizzare anche l'`IncludeDirectoryFilter`per monitorare solo alcune sottodirectory specificate nel filtro.

`RecordParser`

Specifica il modo in cui il tipo di origine `DirectorySource` deve analizzare i file di log disponibili nella directory specificata. Questa coppia chiave-valore è obbligatoria e i valori validi sono i seguenti:

- `SingleLine`— Ogni riga del file di log è un record di log.
- `SingleLineJson`— Ogni riga del file di log è un record di log in formato JSON. Questo parser è utile quando si desidera aggiungere ulteriori coppie chiave-valore al JSON tramite la decorazione degli oggetti. Per ulteriori informazioni, consulta Configurazione delle decorazioni sink. Per un esempio che utilizza il parser di record`SingleLineJson`, consulta Tutorial: Trasmetti file di registro JSON ad Amazon S3 utilizzando Kinesis Agent per Windows.
- `Timestamp`: Una o più righe possono includere un record di log. I record di log iniziano con un timestamp. Questa opzione richiede di specificare la coppia chiave-valore `TimestampFormat`.
- `Regex`— Ogni record inizia con il testo che corrisponde a una determinata espressione regolare. Questa opzione richiede di specificare la coppia chiave-valore `Pattern`.
- `SysLog`— Indica che il file di log viene scritto nellasyslogformato standard. Il file di log viene analizzato in record in base a tali specifiche.
- `Delimited`— Una versione più semplice del parser di record Regex in cui gli elementi dei dati nei record di log sono separati da un delimitatore coerente. Questa opzione è più facile da usare e più veloce nell'esecuzione rispetto al parser Regex ed è da preferire quando questa opzione è disponibile. Quando si utilizza questa opzione, è necessario specificare la coppia chiave-valore `Delimiter`.

`TimestampField`

Specifica quale campo JSON contiene il timestamp per il record. Questo viene utilizzato solo con `SingleLineJson` `RecordParser`. Questa coppia chiave-valore è facoltativa. Se non viene specificato, Kinesis Agent per Windows usa l'ora in cui il record è stato letto per il timestamp. Uno dei vantaggi di specificare questa coppia chiave-valore è che le statistiche di latenza generate da Kinesis Agent per Windows sono più accurate.

`TimestampFormat`

Specifica come analizzare la data e l'ora associati al record. Il valore è una stringa con il formato `epoch` o una stringa data/ora .NET. Se il valore è `epoch`, l'ora viene analizzata in base all'ora UNIX Epoch. Per ulteriori informazioni sull'ora UNIX Epoch, consulta Ora Unix. Per ulteriori informazioni sulle stringhe di formato data/ora .NET, consulta Stringhe personalizzate nel formato data e ora nella documentazione di Microsoft .NET). Questa coppia chiave-valore è obbligatoria solo se viene specificato il parser di record `Timestamp` oppure il parser di record `SingleLineJson` viene specificato con la coppia chiave-valore `TimestampField`.

`Pattern`

Specifica un'espressione regolare che deve corrispondere alla prima riga di un record potenzialmente multiriga. Questa coppia chiave-valore è necessaria solo per il parser di record `Regex`.

`ExtractionPattern`

Specifica un'espressione regolare che deve utilizzare gruppi denominati. Il record è analizzato con l'utilizzo di questa espressione regolare e i gruppi denominati formano i campi del record analizzato. Questi campi vengono quindi utilizzati come base per la costruzione di oggetti o documenti JSON o XML, che sono poi trasmessi dai sink ai vari servizi AWS. Questa coppia chiave-valore è facoltativa ed è disponibile con l'`Regex`e il parser Timestamp.

Il nome del gruppo `Timestamp` viene appositamente elaborato, poiché indica al parser `Regex` quale campo contiene la data e l'ora per ogni record di ciascun file di log.

`Delimiter`

Specifica il carattere o la stringa che separa ciascun elemento in ogni record di log. Questa coppia chiave-valore deve essere (e può solo essere) disponibile solo con il parser di record `Delimited`. Utilizzare la sequenza di due caratteri `\t` per rappresentare il carattere di tabulazione.

`HeaderPattern`

Specifica un'espressione regolare per la riga corrispondente nel file di log che contiene il set di intestazioni per il record. Se il file di log non contiene le informazioni dell'intestazione, utilizzare la coppia chiave-valore `Headers` per specificare le intestazioni implicite. La coppia chiave-valore `HeaderPattern` è facoltativa ed è solo valida per il parser di record `Delimited`.

> **Note**
>
> Una voce di intestazione vuota (lunghezza 0) per una colonna comporta che i dati per quella colonna vengono filtrati dall'output finale dell'output analizzato `DirectorySource`.

`Headers`

Specifica i nomi delle colonne di dati analizzati utilizzando il delimitatore specificato. Questa coppia chiave-valore è facoltativa ed è solo valida per il parser di record `Delimited`.

> **Note**
>
> Una voce di intestazione vuota (lunghezza 0) per una colonna comporta che i dati per quella colonna vengono filtrati dall'output finale dell'output analizzato `DirectorySource`.

`RecordPattern`

Specifica un'espressione regolare che identifica righe del file di log che contengono dati di record. A parte la riga di intestazione facoltativa identificata da `HeaderPattern`, le righe che non corrispondono al `RecordPattern` specificato vengono ignorate durante l'elaborazione dei record. Questa coppia chiave-valore è facoltativa ed è solo valida per il parser di record `Delimited`. Se non viene fornito, il valore di default è da considerare qualsiasi riga che non corrisponde all'opzionale `HeaderPattern` o l'opzionale `CommentPattern` come riga che contiene i dati dei record analizzabili.

`CommentPattern`

Specifica un'espressione regolare che identifica le righe del file di log che devono essere escluse prima di analizzare i dati nel file di log. Questa coppia chiave-valore è facoltativa ed è solo valida per il parser di record `Delimited`. Se non viene fornito, il valore di default è da considerare qualsiasi riga che non corrisponde all'opzionale `HeaderPattern` ed è una riga che contiene i dati dei record analizzabili, a meno che non venga specificato `RecordPattern`.

`TimeZoneKind`

Specifica se il timestamp nel file di log deve essere considerato nel fuso orario locale o nel fuso orario UTC. Si tratta di un'opzione facoltativa, impostata di default su UTC. Gli unici valori validi per questa coppia chiave-valore sono `Local` o `UTC`. Il timestamp non è mai alterato, se `TimeZoneKind` non è specificato o se il valore è UTC. Il timestamp viene convertito in UTC quando il`TimeZoneKind`Il valore è`Local`Se il sink che riceve il timestamp è CloudWatch Logs, oppure il record analizzato viene inviato ad altri sink. Date e ore incorporate nei messaggi non vengono convertite.

`SkipLines`

Quando specificato, controlla il numero di righe ignorati all'inizio di ogni file di log prima che avvenga l'analisi dei record. Si tratta di un'opzione facoltativa e il valore di default è 0.

Encoding

Per impostazione predefinita, Kinesis Agent per Windows è in grado di rilevare automaticamente la codifica dal bytemark. Tuttavia, la codifica automatica potrebbe non funzionare correttamente su alcuni formati unicode precedenti. Nell'esempio seguente viene specificata la codifica necessaria per eseguire lo streaming di un registro di Microsoft SQL Server.

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

Per un elenco dei nomi di codifica, consultaElenco delle codificazioninella documentazione di Microsoft.NET.

ExtractionRegexOptions

È possibile utilizzare`ExtractionRegexOptions`Per semplificare le espressioni regolari. Questa coppia chiave-valore è facoltativa. Il valore di default è `"None"`.

L'esempio seguente specifica che l'`"."`l'espressione corrisponde a qualsiasi carattere incluso`\r \n`: .

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

Per un elenco dei campi possibili per ExtractionRegexOptions, consulta leRegexOptions Enumnella documentazione di Microsoft.NET.

Parser di record **`Regex`**

È possibile analizzare i log di testo non strutturati usando il parser di record `Regex` insieme alle coppie chiave-valore `TimestampFormat`, `Pattern` e `ExtractionPattern`. Supponiamo, ad esempio, che il file di log appaia come il seguente:

```
[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'
```

È possibile specificare l'espressione regolare per la coppia chiave-valore `Pattern` per aiutare a dividere il file di log in singole voci di log:

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

Questa espressione regolare corrisponde alla sequenza seguente:

1. La posizione iniziale della stringa analizzata.
2. Uno o più caratteri di parole racchiusi in parentesi quadre.
3. Un timestamp circondato da parentesi quadre. Al timestamp corrisponde la sequenza indicata:
   a. Anno a quattro cifre
   b. Una barra
   c. Mese di due cifre
   d. Una barra
   e. Giorno di due cifre
   f. Carattere di spazio
   g. Ora di due cifre
   h. Due punti
   i. Minuto di due cifre
   j. Due punti
   k. Secondo di due cifre
   l. Un punto
   m. Millisecondo di tre cifre

È possibile specificare il formato seguente per la coppia chiave-valore `TimestampFormat` per convertire il timestamp testuale in una data e un'ora:

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

È possibile utilizzare la seguente espressione regolare per estrarre i campi dei record di log tramite la coppia chiave-valore `ExtractionPattern`.

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

Questa espressione regolare corrisponde ai seguenti gruppi in sequenza:

1. `Severity`— Uno o più caratteri di parole racchiusi in parentesi quadre.
2. `TimeStamp`— Consulta la descrizione precedente per il timestamp.
3. Tre sequenze tra parentesi quadre senza nome di zero o più caratteri vengono ignorate.
4. `SubSystem`— Uno o più caratteri di parole racchiusi in parentesi quadre.
5. `Module`— Uno o più caratteri di parole racchiusi in parentesi quadre.
6. Una sequenza tra parentesi quadre senza nome di zero o più caratteri viene ignorata.
7. Uno spazio senza nome viene ignorato.
8. `Message`— Zero o più caratteri racchiusi in virgolette singole.

La seguente dichiarazione di origine abbina queste espressioni regolari e il formato data ora per fornire le istruzioni complete all'agente Kinesis per Windows per l'analisi di questo tipo di file di 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**
>
> Le barre oblique in file in formato JSON devono essere precedute da un'ulteriore barra obliqua.

Per ulteriori informazioni sulle espressioni regolari, vedere Regular Expression Language - Guida di riferimento rapida nella documentazione di Microsoft.NET.

Parser di record **`Delimited`**

È possibile usare il parser di record `Delimited` per analizzare file di dati e di log semistrutturati in cui vi è una sequenza di caratteri coerente che separa ogni colonna di dati in ogni riga di dati. Ad esempio, i file CSV utilizzano la virgola per separare ogni colonna di dati, mentre i file TSV utilizzano una tabulazione.

Supponiamo che si desidera analizzare un file di log Microsoft NPS Database Format creato da un server dei criteri di rete. Di seguito è riportato un possibile esempio di tale file:

```
"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,,,,
```

Il seguente file di configurazione di esempio `DirectorySource` include una dichiarazione `appsettings.json` che utilizza il parser di record `Delimited` per analizzare questo testo nella rappresentazione di un oggetto. Quindi, effettua lo streaming di dati in formato JSON a 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"
        }
    ]
}
```

I dati in formato JSON trasmessi in streaming a Kinesis Data Firehose hanno l'aspetto seguente:

```
{
    "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": ""
}
```

Parser di record **SysLog**

Per il parser di record SysLog, l'output analizzato dalla sorgente include le informazioni riportate di seguito:

Attributo	Tipo	Descrizione
SysLogTimeStamp	Stringa	La data e l'ora originali dal file di log in formato syslog.
Hostname	Stringa	Il nome del computer in cui risiede il file di log in formato syslog.
Program	Stringa	Il nome dell'applicazione o del servizio che ha generato il file di log.
Message	Stringa	Il messaggio di log generato dall'appl icazione o dal servizio.
TimeStamp	Stringa	La data e l'ora analizzati nel formato ISO 8601.

Di seguito è riportato un esempio di dati SysLog trasformati nel formato 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

Di seguito sono illustrate le coppie chiave-valore disponibili per l'origine `DirectorySource` e i `RecordParser` relativi alle coppie chiave-valore.

Nome chiave	RecordParser	Note
`SourceType`	Obbligatorio per tutti	Deve avere il valore `DirectorySource`
`Directory`	Obbligatorio per tutti	
`FileNameFilter`	Facoltativo per tutti	
`RecordParser`	Obbligatorio per tutti	
`TimestampField`	Facoltativo per `SingleLineJson`	
`TimestampFormat`	Richiesto per `Timestamp` e per `SingleLineJson` se viene specificato il `TimestampField`	
`Pattern`	Obbligatorio per `Regex`	
`ExtractionPattern`	Facoltativo per `Regex`	Richiesto per `Regex` se il sink specifica il formato `json` o `xml`
`Delimiter`	Obbligatorio per `Delimited`	
`HeaderPattern`	Facoltativo per `Delimited`	

Nome chiave	RecordParser	Note
`Headers`	Facoltativo per `Delimited`	
`RecordPattern`	Facoltativo per `Delimited`	
`CommentPattern`	Facoltativo per `Delimited`	
`TimeZoneKind`	Facoltativo per `Regex`, `Timestamp` , `SysLog` e `SingleLineJson` quando viene identificato un campo data e ora	
`SkipLines`	Facoltativo per tutti	

Configurazione ExchangeLogSource

Il tipo `ExchangeLogSource` viene utilizzato per raccogliere i log da Microsoft Exchange. Exchange produce log in diversi tipi di formati di log. Questo tipo di origine li analizza tutti. Sebbene sia possibile analizzarli utilizzando il tipo `DirectorySource` con il record parser `Regex`, è molto più semplice utilizzare `ExchangeLogSource`. Questo perché non è più necessario progettare e fornire espressioni regolari per i formati di file di log. Di seguito è riportato un esempio della dichiarazione `ExchangeLogSource`:

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

Tutte le dichiarazioni di scambio possono offrire le seguenti coppie chiave-valore:

`SourceType`

Deve essere una stringa letterale `"ExchangeLogSource"` (obbligatoria).

`Directory`

Il percorso alla directory contenente i file di log (obbligatorio).

`FileNameFilter`

Eventualmente limita il set di file nella directory in cui i dati di log sono raccolti in base al modello di denominazione dei file jolly. Se questa coppia chiave-valore non viene specificata, allora, per impostazione predefinita, vengono raccolti i dati di log da parte di tutti i file nella directory.

`TimestampField`

Il nome della colonna contenente la data e l'ora del record. Questa coppia chiave-valore è facoltativa e non deve essere specificata se il nome del campo è `date-time` o `DateTime`. Altrimenti, è obbligatoria.

Configurazione W3SVCLogSource

Il tipo `W3SVCLogSource` viene utilizzato per raccogliere i log da Internet Information Services (IIS) per Windows.

Di seguito è riportato un esempio della dichiarazione `W3SVCLogSource`:

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

Tutte le dichiarazioni `W3SVCLogSource` possono fornire le seguenti coppie chiave-valore:

`SourceType`

Deve essere una stringa letterale `"W3SVCLogSource"` (obbligatoria).

`Directory`

Il percorso alla directory contenente i file di log (obbligatorio).

`FileNameFilter`

Eventualmente limita il set di file nella directory in cui i dati di log sono raccolti in base al modello di denominazione dei file jolly. Se questa coppia chiave-valore non viene specificata, allora, per impostazione predefinita, vengono raccolti i dati di log da parte di tutti i file nella directory.

Configurazione UlsSource

Il tipo `UlsSource` viene utilizzato per raccogliere i log da Microsoft SharePoint. Di seguito è riportato un esempio della dichiarazione `UlsSource`:

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

Tutte le dichiarazioni `UlsSource` possono fornire le seguenti coppie chiave-valore:

`SourceType`

Deve essere una stringa letterale `"UlsSource"` (obbligatoria).

`Directory`

Il percorso alla directory contenente i file di log (obbligatorio).

`FileNameFilter`

Eventualmente limita il set di file nella directory in cui i dati di log sono raccolti in base al modello di denominazione dei file jolly. Se questa coppia chiave-valore non viene specificata, allora, per impostazione predefinita, vengono raccolti i dati di log da parte di tutti i file nella directory.

Configurazione WindowsEventLogSource

Il tipo `WindowsEventLogSource` viene utilizzato per raccogliere gli eventi dal servizio Log eventi di Windows. Di seguito è riportato un esempio della dichiarazione `WindowsEventLogSource`:

```
{
```

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

Tutte le dichiarazioni `WindowsEventLogSource` possono fornire le seguenti coppie chiave-valore:

`SourceType`

Deve essere una stringa letterale `"WindowsEventLogSource"` (obbligatoria).

`LogName`

Gli eventi vengono raccolti dal log specificato. I valori comuni includono `Application`, `Security` e `System`, ma è possibile specificare qualsiasi nome di log di eventi di Windows valido. Questa coppia chiave-valore è obbligatoria.

`Query`

Eventualmente limita gli eventi output di `WindowsEventLogSource`. Se questa coppia chiave-valore non viene specificata, allora, per impostazione predefinita, tutti gli eventi sono output. Per ulteriori informazioni sulla sintassi di questo valore, consulta Query ed XML di eventi nella documentazione di Windows. Per ulteriori informazioni sulle definizioni del livello di log, consulta Tipi di eventi nella documentazione di Windows.

`IncludeEventData`

Eventualmente consente la raccolta e lo streaming di dati di eventi specifici del provider associati a eventi dal log di eventi di Windows specificato quando il valore di questa coppia chiave-valore è `"true"`. Vengono inclusi solo i dati di eventi che possono essere correttamente serializzati. Questa coppia chiave-valore è facoltativa e, se non è specificata, i dati di eventi specifici del provider non vengono raccolti.

> **Note**
>
> L'inclusione di dati relativi a eventi potrebbe aumentare in modo significativo la quantità di dati provenienti da questa origine. La dimensione massima di un evento può essere 262.143 byte con dati relativi a eventi inclusi.

L'output analizzato da `WindowsEventLogSource` contiene le informazioni riportate di seguito:

Attributo	Tipo	Descrizione
`EventId`	Int	L'identificatore del tipo di evento.
`Description`	Stringa	Testo che descrive i dettagli dell'evento.
`LevelDisplayName`	Stringa	La categoria dell'evento (uno tra Error, Warning, Information, Success Audit, Failure Audit).
`LogName`	Stringa	Dove l'evento è stato registrato (i valori tipici sono `Application` , `Security` e `System`, ma ci sono molte possibilità).
`MachineName`	Stringa	Quale computer ha registrato l'evento.
`ProviderName`	Stringa	Quale applicazione o servizio ha registrato l'evento.
`TimeCreated`	Stringa	Quando si è verificato l'evento nel formato ISO 8601.
`Index`	Int	Dove si trova la voce all'interno del log.
`UserName`	Stringa	Chi ha creato la voce, se noto.
`Keywords`	Stringa	Tipo di evento. I valori standard includono `AuditFailure` (eventi di audit di sicurezza non riusciti), `AuditSuccess` (eventi di audit di sicurezza riusciti), `Classic` (eventi generati con la funzione `RaiseEvent`), `Correlation Hint` (eventi di trasferimento), `SQM` (eventi Service Quality Mechanism), `WDI Context` (eventi di contesto infrastrutturale della

Attributo	Tipo	Descrizione
		diagnostica Windows) e `WDI Diag` (eventi di diagnostica infrastrutturale della diagnostica Windows).
`EventData`	Elenco di oggetti	Dati aggiuntivi facoltativi specifici del provider sull'evento di log. Questo è incluso solo se il valore per la coppia chiave-valore `IncludeEventData` è `"true"`.

Di seguito è illustrato un esempio di un evento trasformato in 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=="
]}
```

Configurazione WindowsEventLogPollingSource

`WindowsEventLogPollingSource`utilizza un meccanismo basato sul polling per raccogliere tutti i nuovi eventi dal registro eventi che corrispondono ai parametri configurati. L'intervallo di polling viene aggiornato dinamicamente tra 100 ms e 5000 ms a seconda del numero di eventi raccolti durante l'ultimo sondaggio. Di seguito è riportato un esempio della dichiarazione `WindowsEventLogPollingSource`:

```
{
```

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

Tutte le dichiarazioni `WindowsEventLogPollingSource` possono fornire le seguenti coppie chiave-valore:

`SourceType`

Deve essere una stringa letterale `"WindowsEventLogPollingSource"` (obbligatoria).

`LogName`

Specifica il registro. Opzioni valide sono`Application`,`Security`,`System`o altri registri validi.

`IncludeEventData`

Facoltativo. Quando`true`, specifica che EventData extra quando viene eseguito lo streaming come JSON e XML è incluso. Il valore predefinito è `false`.

`Query`

Facoltativo. I registri eventi di Windows supportano l'esecuzione di query di eventi utilizzando espressioni XPath, che è possibile specificare utilizzando`Query`: . Per ulteriori informazioni, consultaQuery di eventi e XML di eventinella documentazione di Microsoft.

`CustomFilters`

Facoltativo. Un elenco di filtri separati da un punto e virgola (`;`). È possibile specificare i seguenti filtri.

`ExcludeOwnSecurityEvents`

Esclude gli eventi di protezione generati dall'agente Kinesis per Windows stesso.

Configurazione WindowsETWEventSource

Il tipo `WindowsETWEventSource` viene utilizzato per raccogliere tracce di eventi di servizi e di applicazioni utilizzando una funzionalità denominata Event Tracing for Windows (ETW). Per ulteriori informazioni, consulta Tracciamento degli eventi nella documentazione di Windows.

Di seguito è riportato un esempio della dichiarazione `WindowsETWEventSource`:

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

Tutte le dichiarazioni `WindowsETWEventSource` possono fornire le seguenti coppie chiave-valore:

`SourceType`

Deve essere una stringa letterale `"WindowsETWEventSource"` (obbligatoria).

`ProviderName`

Specifica quale provider di eventi utilizzare per raccogliere gli eventi di traccia. Questo deve essere un nome di provider ETW valido per un fornitore installato. Per stabilire quali fornitori sono installati, eseguire il seguente comando in una finestra del prompt dei comandi di Windows:

```
logman query providers
```

`TraceLevel`

Specifica quali categorie di eventi di traccia devono essere raccolte. I valori consentiti includono `Critical`, `Error`, `Warning`, `Informational` e `Verbose`. Il significato esatto dipende dal provider ETW selezionato.

`MatchAnyKeyword`

Questo valore è un numero a 64 bit, in cui ogni bit rappresenta una singola parola chiave. Ogni parola chiave descrive una categoria di eventi da raccogliere. Per le parole chiave supportate e i relativi valori e come sono correlati a `TraceLevel`, consulta la documentazione del provider relativo. Ad esempio, per informazioni sul provider CLR ETW, consulta Parole chiave e livelli CLR ETW nella documentazione di Microsoft.NET Framework.

Nell'esempio precedente, 32768 (0x00008000) rappresenta la `ExceptionKeyword` per il provider CLR ETW che indica al provider di raccogliere informazioni sulle eccezioni generate. Sebbene JSON non supporti nativamente costanti hex, è possibile specificarle

per `MatchAnyKeyword` inserendole in una stringa. È anche possibile specificare diverse costanti separate da virgole. Ad esempio, è possibile usare quanto segue per specificare `ExceptionKeyword` e `SecurityKeyword` (0x00000400):

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

Per assicurarsi che tutte le parole chiave specificate per un provider vengano attivate, vengono abbinati più valori di parole chiave utilizzando O che vengono trasferiti poi a tale provider.

L'output da `WindowsETWEventSource` contiene le informazioni riportate di seguito per ogni evento:

Attributo	Tipo	Descrizione
`EventName`	Stringa	Che tipo di evento si è verificato.
`ProviderName`	Stringa	Quale provider ha rilevato l'evento.
`FormattedMessage`	Stringa	Un riepilogo testuale dell'evento.
`ProcessID`	Int	Quale processo ha segnalato l'evento.
`ExecutingThreadID`	Int	Quale thread all'interno del processo ha segnalato l'evento.
`MachineName`	Stringa	Il nome del desktop o del server che effettua la segnalazione dell'evento.
`Payload`	Tabella hash	Una tabella con una chiave stringa e qualsiasi tipo di oggetto come valore. La chiave è il nome

Attributo	Tipo	Descrizione
		della voce di payload e il valore è il valore della voce di payload. Il payload dipende dal provider.

Di seguito è illustrato un esempio di un evento trasformato in 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
      }
}
```

Configurazione WindowsPerformanceCounterSource

Il tipo `WindowsPerformanceCounterSource` raccoglie parametri contatori di prestazioni da Windows. Di seguito è riportato un esempio della dichiarazione `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"]
  }
 ]
}
```

Tutte le dichiarazioni `WindowsPerformanceCounterSource` possono fornire le seguenti coppie chiave-valore:

`SourceType`

Deve essere una stringa letterale `"WindowsPerformanceCounterSource"` (obbligatoria).

`Categories`

Specifica un set di gruppi di parametri contatori delle prestazioni da raccogliere da Windows. Ogni gruppo di parametri contiene le seguenti coppie chiave-valore:

`Category`

Specifica il set di contatori di parametri da raccogliere (obbligatorio).

`Instances`

Specifica il set di oggetti di interesse quando c'è un set unico di contatori delle prestazioni per oggetto. Ad esempio, quando la categoria è `LogicalDisk`, c'è un set di contatori delle

prestazioni per unità disco. Questa coppia chiave-valore è facoltativa. È possibile utilizzare i caratteri jolly `*` e `?` per soddisfare più istanze. Per aggregare i valori su tutte le istanze, specificare `_Total`.

È possibile utilizzare anche`InstanceRegex`, che accetta espressioni regolari che contengono`*`carattere jolly come parte del nome dell'istanza.

`Counters`

Specifica quali parametri raccogliere per la categoria specificata. Questa coppia chiave-valore è obbligatoria. È possibile utilizzare i caratteri jolly `*` e `?` per soddisfare più contatori. È possibile specificare `Counters` utilizzando solo il nome oppure utilizzando il nome e l'unità. Se le unità del contatore non sono specificate, Kinesis Agent per Windows tenta di dedurre le unità dal nome. Se tali estrapolazioni non sono corrette, l'unità può essere specificata in modo esplicito. È possibile modificare i nomi dei `Counter`, se necessario. La rappresentazione più complessa di un contatore è un oggetto con le seguenti coppie chiave-valore:

`Counter`

Il nome del contatore. Questa coppia chiave-valore è obbligatoria.

`Rename`

Il nome del contatore da presentare al sink. Questa coppia chiave-valore è facoltativa.

`Unit`

Il significato del valore associato al contatore. Per un elenco completo dei nomi di unità validi, consulta la documentazione sulle unità in[MetricDatum](#)nellaGuida di riferimento all'API di Amazon CloudWatch: .

Di seguito è riportato un esempio di una specifica complessa di un contatore:

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

`WindowsPerformanceCounterSource`Può essere utilizzato solo con una pipe che specifica un sink Amazon CloudWatch. Utilizzare un sink separato se i parametri integrati

di Kinesis Agent per Windows vengono trasmessi in streaming anche a CloudWatch. Esaminare il log di Kinesis Agent per Windows dopo l'avvio del servizio per stabilire quali unità sono state desunte per i contatori quando queste non sono state specificate nella finestra di`WindowsPerformanceCounterSource`Dichiarazioni. Utilizzare PowerShell per stabilire i nomi validi per categorie, istanze e contatori.

Per visualizzare informazioni su tutte le categorie, inclusi i contatori associati a set di contatori, eseguire questo comando in una finestra di PowerShell:

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

Per stabilire quali istanze sono disponibili per ciascuno dei contatori nel set di contatori, eseguire un comando simile a quello dell'esempio seguente in una finestra di PowerShell:

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

Il valore del parametro `Counter` deve essere uno dei percorsi da un membro `PathsWithInstances` elencato dalla precedente invocazione del comando `Get-Counter -ListSet`.

Origine dei parametri integrati di Kinesis Agent per Windows

Oltre alle fonti di metriche ordinarie, come il`WindowsPerformanceCounterSource`tipo (vedereConfigurazione WindowsPerformanceCounterSource), il tipo di sink CloudWatch è in grado di ricevere i parametri da un'origine speciale che raccoglie i parametri sull'agente Kinesis per Windows. I parametri di Kinesis Agent per Windows sono disponibili anche nella`KinesisTap`dei contatori delle prestazioni di Windows.

La`MetricsFilter`La coppia chiave-valore per le dichiarazioni sink CloudWatch specifica quali parametri vengono trasmessi a CloudWatch dall'origine integrata di parametri Kinesis Agent per Windows. Il valore è una stringa che contiene una o più espressioni di filtro separate da punti e virgola; ad esempio:

`"MetricsFilter": "FilterExpression1;FilterExpression2"`

Una metrica che corrisponde a una o più espressioni di filtro viene inviata in streaming a CloudWatch.

I parametri di istanze singole sono di natura globale e non sono associati a origini o sink particolari. I parametri di più istanze sono dimensionali in base alla sorgente o alla dichiarazione del sink `Id`. Ogni tipo di origine o sink può avere una serie diversa di parametri.

Per un elenco dei nomi dei parametri integrati di Kinesis Agent per Windows, consultaElenco delle metriche di Kinesis Agent per Windows: .

Per i parametri di una singola istanza, l'espressione di filtro è il nome della metrica, ad esempio:

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

Per più parametri di istanza, l'espressione del filtro è il nome del parametro, un punto (`.`) e quindi l'`Id` dell'origine o la dichiarazione del sink che ha generato tale parametro. Ad esempio, supponiamo che ci sia una dichiarazione sink con un `Id` di `MyFirehose`:

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

È possibile utilizzare i modelli di carattere jolly speciali progettati per distinguere i parametri di istanze singole e multiple.

- L'asterisco (`*`) corrisponde a zero o più caratteri tranne il punto (`.`).
- Il punto interrogativo (`?`) corrisponde a un carattere tranne il punto.
- Qualsiasi altro carattere corrisponde solo a se stesso.
- `_Total` è un token speciale che provoca l'aggregazione di tutti i valori che corrispondono a più istanze in tutta la dimensione.

L'esempio seguente corrisponde a tutti i parametri di istanze singole:

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

Poiché un asterisco non corrisponde al carattere punto, sono inclusi solo i parametri di istanze singole.

L'esempio seguente corrisponde a tutti i parametri di istanze multiple:

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

L'esempio seguente corrisponde a tutti i parametri (singoli e multipli):

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

L'esempio seguente aggrega tutti i parametri di istanze multiple in tutte le origini e i sink:

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

L'esempio seguente aggrega tutti i parametri di Kinesis Data Firehose per tutti i sink di Kinesis Data:

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

L'esempio seguente corrisponde a tutti i parametri di errore delle istanze singole e multiple:

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

L'esempio seguente corrisponde a tutti i parametri di errori non recuperabili in tutte le origini e i sink:

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

Per informazioni su come specificare una pipe che utilizza l'origine dei parametri integrati di Kinesis Agent per Windows, vediConfigurazione dell'agente Kinesis per le pipe metriche di Windows: .

Elenco delle metriche di Kinesis Agent per Windows

Di seguito è riportato un elenco di parametri di istanze singole e multiple disponibili per Kinesis Agent per Windows.

Parametri di istanze singole

Sono disponibili i seguenti parametri di istanze singole:

`KinesisTapBuildNumber`

Il numero di versione di Kinesis Agent per Windows.

`PipesConnected`

Quante pipe hanno collegato l'origine al sink in modo corretto.

`PipesFailedToConnect`

Quante pipe hanno collegato l'origine al sink in modo errato.

`SinkFactoriesFailedToLoad`

Quanti tipi di sink non è stato possibile caricare in Kinesis Agent per Windows correttamente.

`SinkFactoriesLoaded`

Quanti tipi di sink sono stati caricati in Kinesis Agent per Windows correttamente.

`SinksFailedToStart`

Quanti sink non sono stati iniziati correttamente, di solito a causa di dichiarazioni sink errate.

`SinksStarted`

Quanti sink sono stati avviati correttamente.

`SourcesFailedToStart`

Quante origini non sono state iniziate correttamente, di solito a causa di dichiarazioni errate.

`SourcesStarted`

Quante origini sono state avviate correttamente.

`SourceFactoriesFailedToLoad`

Quanti tipi di origine non sono stati caricati in Kinesis Agent per Windows correttamente.

`SourceFactoriesLoaded`

Quanti tipi di origine sono stati caricati in Kinesis Agent per Windows.

Parametri di istanze multiple

Per le istanze multiple sono disponibili i seguenti parametri:

Parametri DirectorySource

`DirectorySourceBytesRead`

Quanti byte sono stati letti durante l'intervallo per questo `DirectorySource`.

`DirectorySourceBytesToRead`

Quanti numeri noti di byte sono disponibili da leggere che non sono stati ancora letti da Kinesis Agent per Windows.

`DirectorySourceFilesToProcess`

Quanti file noti da esaminare che non sono ancora stati esaminati da Kinesis Agent per Windows.

`DirectorySourceRecordsRead`

Quanti record sono stati letti durante l'intervallo per questo `DirectorySource`.

Parametri WindowsEventLogSource

`EventLogSourceEventsError`

Quanti eventi di log di eventi di Windows non sono stati letti correttamente.

`EventLogSourceEventsRead`

Quanti eventi di log di eventi di Windows sono stati letti correttamente.

Parametri sink KinesisFirehose

`KinesisFirehoseBytesAccepted`

Quanti byte sono stati accettati durante l'intervallo.

`KinesisFirehoseClientLatency`

Quanto tempo è trascorso tra la generazione e lo streaming dei record al servizio Kinesis Data Firehose.

`KinesisFirehoseLatency`

Quanto tempo è trascorso tra l'inizio e la fine dello streaming dei record al servizio Kinesis Data Firehose.

`KinesisFirehoseNonrecoverableServiceErrors`

Quante volte i record non potevano essere inviati senza errori al servizio Kinesis Data Firehose nonostante i tentativi.

`KinesisFirehoseRecordsAttempted`

Quanti record hanno tentato di eseguire lo streaming al servizio Kinesis Data Firehose.

`KinesisFirehoseRecordsFailedNonrecoverable`

Quanti record non sono stati trasmessi correttamente al servizio Kinesis Data Firehose nonostante i tentativi.

`KinesisFirehoseRecordsFailedRecoverable`

Quanti record sono stati trasmessi correttamente al servizio Kinesis Data Firehose ma solo dopo nuovi tentativi.

`KinesisFirehoseRecordsSuccess`

Quanti record sono stati trasmessi correttamente al servizio Kinesis Data Firehose senza nuovi tentativi.

`KinesisFirehoseRecoverableServiceErrors`

Quante volte i record sono stati inviati correttamente al servizio Kinesis Data Firehose ma solo dopo nuovi tentativi.

Parametri KinesisStream

`KinesisStreamBytesAccepted`

Quanti byte sono stati accettati durante l'intervallo.

`KinesisStreamClientLatency`

Quanto tempo è trascorso tra la generazione e lo streaming dei record al servizio Kinesis Data Streams.

`KinesisStreamLatency`

Quanto tempo è trascorso tra l'inizio e la fine dello streaming dei record al servizio Kinesis Data Streams.

`KinesisStreamNonrecoverableServiceErrors`

Quante volte i record non potevano essere inviati senza errori al servizio Kinesis Data Streams nonostante i tentativi.

`KinesisStreamRecordsAttempted`

Quanti record hanno tentato di eseguire lo streaming al servizio Kinesis Data Streams.

`KinesisStreamRecordsFailedNonrecoverable`

Quanti record non sono stati trasmessi correttamente al servizio Kinesis Data Streams nonostante i tentativi.

`KinesisStreamRecordsFailedRecoverable`

Quanti record sono stati trasmessi correttamente al servizio Kinesis Data Streams ma solo dopo nuovi tentativi.

`KinesisStreamRecordsSuccess`

Quanti record sono stati trasmessi correttamente al servizio Kinesis Data Streams senza nuovi tentativi.

`KinesisStreamRecoverableServiceErrors`

Quante volte i record sono stati inviati correttamente al servizio Kinesis Data Streams ma solo dopo nuovi tentativi.

Parametri CloudWatchLog

`CloudWatchLogBytesAccepted`

Quanti byte sono stati accettati durante l'intervallo.

`CloudWatchLogClientLatency`

Quanto tempo è trascorso tra la generazione e lo streaming dei record al servizio CloudWatch Logs.

`CloudWatchLogLatency`

Quanto tempo è trascorso tra l'inizio e la fine dello streaming dei record al servizio CloudWatch Logs.

`CloudWatchLogNonrecoverableServiceErrors`

Quante volte i record non potevano essere inviati senza errori al servizio CloudWatch Logs nonostante i tentativi.

`CloudWatchLogRecordsAttempted`

Quanti record hanno tentato di eseguire lo streaming al servizio CloudWatch Logs.

`CloudWatchLogRecordsFailedNonrecoverable`

Quanti record non sono stati trasmessi correttamente al servizio CloudWatch Logs nonostante i tentativi.

`CloudWatchLogRecordsFailedRecoverable`

Quanti record sono stati trasmessi correttamente al servizio CloudWatch Logs ma solo dopo nuovi tentativi.

`CloudWatchLogRecordsSuccess`

Quanti record sono stati trasmessi correttamente al servizio CloudWatch Logs senza nuovi tentativi.

`CloudWatchLogRecoverableServiceErrors`

Quante volte i record sono stati inviati correttamente al servizio CloudWatch Logs ma solo dopo nuovi tentativi.

Parametri di CloudWatch

`CloudWatchLatency`

Quanto tempo in media è trascorso tra l'inizio e la fine dello streaming dei parametri al servizio CloudWatch.

`CloudWatchNonrecoverableServiceErrors`

Quante volte i parametri non potevano essere inviati senza errori al servizio CloudWatch nonostante i tentativi.

`CloudWatchRecoverableServiceErrors`

Quante volte i parametri sono stati inviati senza errori al servizio CloudWatch ma solo dopo nuovi tentativi.

`CloudWatchServiceSuccess`

Quante volte i parametri sono stati inviati senza errori al servizio CloudWatch senza nuovi tentativi.

Configurazione del segnalibro

Per impostazione predefinita, Kinesis Agent per Windows invia i record di log ai sink creati dopo l'avvio dell'agente. Talvolta è utile inviare i record di log precedenti, ad esempio, i record di log creati quando Kinesis Agent per Windows si blocca durante un aggiornamento automatico. La funzione di segnalibro monitora quali record sono stati inviati ai sink. Quando Kinesis Agent per Windows è in modalità segnalibro e si avvia, invia tutti i record di log creati dopo l'arresto di Kinesis Agent per Windows, insieme a qualsiasi record di log creato successivamente. Per controllare questo comportamento, le dichiarazioni di origine basate su file possono eventualmente includere le seguenti coppie chiave-valore:

`InitialPosition`

Specifica la situazione iniziale del segnalibro. I valori possibili sono i seguenti:

`EOS`

Specifica la fine del flusso (EOS). Solo i record di log creati durante l'esecuzione dell'agente vengono inviati ai sink.

`0`

Tutti gli eventi e i record di log vengono inizialmente inviati. Quindi viene creato un segnalibro che assicura che ogni nuovo record di log ed evento creato dopo la creazione del segnalibro venga infine inviato, se Kinesis Agent for Windows è in esecuzione oppure no.

`Bookmark`

Il segnalibro viene inizializzato solo dopo l'ultimo record di log o evento. Quindi viene creato un segnalibro che assicura che ogni nuovo record di log ed evento creato dopo la creazione del segnalibro venga infine inviato, se Kinesis Agent for Windows è in esecuzione oppure no.

I segnalibri sono abilitati come impostazione predefinita. I file vengono memorizzati nella`%ProgramData%\Amazon\KinesisTap`directory.

`Timestamp`

Vengono inviati i record di log e gli eventi creati dopo il valore `InitialPositionTimestamp` (segue la definizione). Quindi viene creato un segnalibro che assicura che ogni nuovo record di log ed evento creato dopo la creazione del segnalibro venga infine inviato, se Kinesis Agent per Windows è in esecuzione oppure no.

`InitialPositionTimestamp`

Specifica il primo timestamp desiderato di record di log o evento. Specificare questa coppia chiave-valore solo quando `InitialPosition` ha un valore di `Timestamp`.

`BookmarkOnBufferFlush`

Questa impostazione può essere aggiunta a qualsiasi fonte segnalibro. Se impostato su`true`, assicura che gli aggiornamenti dei segnalibri si verifichino solo quando un sink invia correttamente un evento a AWS. È possibile sottoscrivere un solo sink a una fonte. Se spedisci i registri a più destinazioni, duplicare le origini per evitare potenziali problemi con la perdita di dati.

Quando Kinesis Agent per Windows viene arrestato per un periodo di tempo prolungato, potrebbe essere necessario eliminare questi segnalibri, perché i record di log ed eventi contrassegnati dal segnalibro non esistono più. I file del segnalibro per un dato id origine sono situati in `%PROGRAMDATA%\Amazon\AWSKinesisTap\`*`source id`*`.bm`.

I segnalibri non funzionano su file rinominati o troncati. Per propria natura, gli eventi ETW e i contatori delle prestazioni non possono essere contrassegnati da segnalibri.

Dichiarazioni dei sink

Le Dichiarazioni dei sink specificano dove e in quale forma log, eventi e parametri devono essere inviati ai diversi servizi AWS. Le seguenti sezioni descrivono le configurazioni per i tipi di sink integrati disponibili in Amazon Kinesis Agent per Microsoft Windows. Poiché Kinesis Agent per Windows è ampliabile, è possibile aggiungere tipi di sink personalizzati. Ogni tipo di sink richiede in genere coppie chiave-valore univoche nella dichiarazioni di configurazione pertinenti per quel tipo di sink.

Tutte le dichiarazioni di sink possono offrire le seguenti coppie chiave-valore:

`Id`

Una stringa univoca che identifica un determinato sink all'interno del file di configurazione (obbligatorio).

`SinkType`

Il nome del tipo di sink per questo sink (obbligatorio). Il tipo di sink specifica la destinazione dei dati di log, degli eventi o dei parametri inviati in streaming da questo sink.

`AccessKey`

Specifica la chiave di accesso AWS da utilizzare per autorizzare l'accesso al servizio AWS associato al tipo di sink. Questa coppia chiave-valore è facoltativa. Per ulteriori informazioni, consulta Configurazione di sicurezza del sink.

`SecretKey`

Specifica la chiave segreta AWS da utilizzare per autorizzare l'accesso al servizio AWS associato al tipo di sink. Questa coppia chiave-valore è facoltativa. Per ulteriori informazioni, consulta Configurazione di sicurezza del sink.

`Region`

Specifica la regione AWS che contiene le risorse di destinazione per lo streaming. Questa coppia chiave-valore è facoltativa.

`ProfileName`

Specifica il profilo AWS da utilizzare per l'autenticazione. Questa coppia chiave-valore è facoltativa, ma se specificata, sostituisce qualsiasi chiave di accesso e chiave segreta specificata. Per ulteriori informazioni, consulta Configurazione di sicurezza del sink.

`RoleARN`

Specifica il ruolo IAM da utilizzare per accedere al servizio AWS associato al tipo di sink. Questa opzione è utile quando Kinesis Agent per Windows è in esecuzione su un'istanza di EC2, ma un altro ruolo sarà più appropriato rispetto al ruolo referenziato dal profilo dell'istanza. Ad esempio, il ruolo su più account può essere utilizzato per risorse di destinazione che non sono nello stesso account AWS come l'istanza EC2. Questa coppia chiave-valore è facoltativa.

`Format`

Specifica il tipo di serializzazione applicato ai dati di eventi e di log prima dello streaming. I valori validi sono `json` e `xml`. Questa opzione è utile quando i dati di analisi downstream nella pipeline dei dati richiedono o preferiscono i dati in una determinata forma. Questa coppia chiave-valore è facoltativa e, se non è specificato, il testo normale dall'origine viene inviato in streaming dal sink al servizio AWS associato al tipo di sink.

`TextDecoration`

Quando non viene specificato un `Format`, `TextDecoration` specifica quale testo aggiuntivo includere durante lo streaming di record di eventi o di log. Per ulteriori informazioni, consulta Configurazione delle decorazioni sink. Questa coppia chiave-valore è facoltativa.

`ObjectDecoration`

Quando viene specificato `Format`, `ObjectDecoration` specifica i dati aggiuntivi inclusi nei record di eventi o di log prima della serializzazione e dello streaming. Per ulteriori informazioni, consulta Configurazione delle decorazioni sink. Questa coppia chiave-valore è facoltativa.

`BufferInterval`

Per ridurre al minimo le chiamate API al servizio AWS associato al tipo di sink Kinesis il buffering di più record di log, eventi o parametri prima dello streaming. In questo modo è possibile risparmiare denaro per servizi che comportano un addebito per chiamata API. `BufferInterval` specifica il tempo massimo (in secondi) in cui deve essere eseguito il buffering dei record prima dello streaming al servizio AWS. Questa coppia chiave-valore è facoltativa e, se specificata, utilizza una stringa per rappresentare il valore.

`BufferSize`

Per ridurre al minimo le chiamate API al servizio AWS associato al tipo di sink Kinesis il buffering di più record di log, eventi o parametri prima dello streaming. In questo modo è possibile risparmiare denaro per servizi che comportano un addebito per chiamata API. `BufferSize` specifica il numero massimo di record per cui deve essere eseguito il buffering prima dello streaming al servizio AWS. Questa coppia chiave-valore è facoltativa e, se specificata, utilizza una stringa per rappresentare il valore.

`MaxAttempts`

Specifica il numero massimo di volte in cui Kinesis Agent per Windows tenta di trasmettere un set di record di log, eventi e parametri a un servizio AWS se lo streaming in maniera regolare ha esito negativo. Questa coppia chiave-valore è facoltativa. Se è specificato, utilizza una stringa per rappresentare il valore. Il valore predefinito è "3".

Per esempi di file di configurazione completi che utilizzano vari tipi di sink, consulta Streaming da log di eventi di applicazioni Windows ai sink.

Argomenti

- Configurazione sink KinesisStream
- Configurazione sink KinesisFirehose
- Configurazione sink CloudWatch
- Configurazione sink CloudWatchLogs

- LocaleFileSystemConfigurazione sink
- Configurazione di sicurezza del sink
- Configurazione diProfileRefreshingAWSCredentialProviderPer aggiornare le credenziali AWS
- Configurazione delle decorazioni sink
- Configurazione di sostituzione di variabili sink
- Configurazione dell'accodamento dei sink
- Configurazione di un Proxy per i sink
- Configurazione delle variabili di risoluzione in più attributi sink
- Configurazione degli endpoint regionali di AWS STS quando si utilizza la proprietà RoleARN nei sink AWS
- Configurazione di VPC Endpoint per i sink AWS
- Configurazione di un mezzo alternativo di proxy

Configurazione sink **`KinesisStream`**

La`KinesisStream`Il tipo di sink invia in streaming record di log e di eventi al servizio di Kinesis Data Streams. Di solito, i dati inviati in streaming a Kinesis Data Streams vengono elaborati da una o più applicazioni personalizzate eseguite utilizzando diversi servizi AWS. I dati vengono inviati in streaming a un flusso denominato configurato utilizzando Kinesis Data Streams. Per ulteriori informazioni, consulta la .Guida per gli sviluppatori di Amazon Kinesis Data Streams: .

Di seguito è riportato un esempio Kinesis Data Streams di:

```
{
    "Id": "TestKinesisStreamSink",
    "SinkType": "KinesisStream",
    "StreamName": "MyTestStream",
    "Region": "us-west-2"
}
```

Tutte le dichiarazioni sink `KinesisStream` possono offrire le seguenti coppie chiave-valore aggiuntive:

`SinkType`

Deve essere specificato e il valore deve essere la stringa letterale `KinesisStream`.

`StreamName`

Specifica il nome del flusso di dati Kinesis che riceve i dati provenienti dalla`KinesisStream`tipo di sink (obbligatorio). Prima di inviare i dati in streaming, configurare il flusso nella console di gestione AWS, nell'interfaccia della riga di comando AWS o tramite un'applicazione utilizzando l'API Kinesis Data Streams.

`RecordsPerSecond`

Specifica il numero massimo di record in Kinesis Data Streams a al secondo. Questa coppia chiave-valore è facoltativa. Se è specificato, utilizza un numero intero per rappresentare il valore. Il valore predefinito è 1000 record.

`BytesPerSecond`

Specifica il numero massimo di byte in Kinesis Data Streams a al secondo. Questa coppia chiave-valore è facoltativa. Se è specificato, utilizza un numero intero per rappresentare il valore. Il valore predefinito è 1 MB.

L'impostazione predefinita di `BufferInterval` per questo tipo di sink è di 1 secondo, e l'impostazione predefinita di `BufferSize` è di 500 record.

Configurazione sink **`KinesisFirehose`**

La`KinesisFirehose`Il tipo di sink invia in streaming record di log e di eventi al servizio Kinesis Data Firehose. Kinesis Data Firehose fornisce i dati provenienti da altri servizi per lo storage. In genere i dati archiviati vengono poi analizzati in fasi successive della pipeline di dati. I dati vengono inviati in streaming a un flusso di distribuzione denominato configurato utilizzando Kinesis Data Firehose. Per ulteriori informazioni, consulta la .Guida per sviluppatori di Amazon Kinesis Data Firehose: .

Di seguito è riportato un esempio della dichiarazione Kinesis Data Firehose:

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

Tutte le dichiarazioni sink `KinesisFirehose` possono offrire le seguenti coppie chiave-valore aggiuntive:

`SinkType`

Deve essere specificato e il valore deve essere la stringa letterale `KinesisFirehose`.

`StreamName`

Specifica il nome del flusso di distribuzione Kinesis Data Firehose che riceve i dati provenienti dalla`KinesisStream`tipo di sink (obbligatorio). Prima di inviare i dati in streaming, configurare il flusso di distribuzione utilizzando la console di gestione AWS, l'interfaccia della riga di comando AWS o tramite un'applicazione utilizzando l'API Kinesis Data Firehose.

`CombineRecords`

Se impostato su`true`, specifica di combinare più record di piccole dimensioni in un record di grandi dimensioni con una dimensione massima di 5 KB. Questa coppia chiave-valore è facoltativa. I record combinati utilizzando questa funzione sono separati da`\n`: . Se si utilizza AWS Lambda per trasformare un record Kinesis Data Firehose, la funzione Lambda deve tenere conto del carattere separatore.

`RecordsPerSecond`

Specifica il numero massimo di record in Kinesis Data Streams a al secondo. Questa coppia chiave-valore è facoltativa. Se è specificato, utilizza un numero intero per rappresentare il valore. Il valore predefinito è 5000 record.

`BytesPerSecond`

Specifica il numero massimo di byte in Kinesis Data Streams a al secondo. Questa coppia chiave-valore è facoltativa. Se è specificato, utilizza un numero intero per rappresentare il valore. Il valore predefinito è 5 MB.

L'impostazione predefinita di `BufferInterval` per questo tipo di sink è di 1 secondo, e l'impostazione predefinita di `BufferSize` è di 500 record.

Configurazione sink CloudWatch

La`CloudWatch`Il tipo di sink effettua lo streaming dei parametri al servizio CloudWatch. Puoi visualizzare i parametri nella console di gestione AWS. Per ulteriori informazioni, consulta la Guida per l'utente di Amazon CloudWatch.

Di seguito è riportato un esempio della dichiarazione sink `CloudWatch`:

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

Tutte le dichiarazioni sink `CloudWatch` possono offrire le seguenti coppie chiave-valore aggiuntive:

`SinkType`

Deve essere specificato e il valore deve essere la stringa letterale `CloudWatch`.

`Interval`

Specifica la frequenza (in secondi) con cui Kinesis Agent per Windows comunica i parametri al servizio CloudWatch. Questa coppia chiave-valore è facoltativa. Se è specificato, utilizza un numero intero per rappresentare il valore. Il valore predefinito è 60 secondi. Specificare 1 secondo se si desiderano parametri CloudWatch ad alta risoluzione.

`Namespace`

Specifica lo spazio dei nomi CloudWatch in cui vengono riportati i dati dei parametri. Gli spazi dei nomi CloudWatch raggruppano un set di parametri. Questa coppia chiave-valore è facoltativa. Il valore predefinito è `KinesisTap`.

`Dimensions`

Specifica le dimensioni CloudWatch utilizzate per isolare i set di parametri all'interno di uno spazio dei nomi. Questo può essere utile per fornire set separati di dati di parametri per ogni desktop o server, ad esempio. Questa coppia chiave-valore è facoltativa e, se specificata, il valore deve rispettare il seguente formato: `"chiave1=valore1;chiave2=valore2..."`. Il valore predefinito è `"ComputerName={computername};InstanceId={instance_id}"`. Questo valore supporta la sostituzione delle variabili sink. Per ulteriori informazioni, consulta Configurazione di sostituzione di variabili sink.

`MetricsFilter`

Specifica quali parametri vengono trasmessi in streaming a CloudWatch dall'origine integrata di Kinesis Agent per Windows. Per ulteriori informazioni sull'origine di parametri integrata di Kinesis Agent per Windows, inclusi i dettagli della sintassi del valore di questa coppia chiave-valore, vedereOrigine dei parametri integrati di Kinesis Agent per Windows: .

Configurazione sink **CloudWatchLogs**

La`CloudWatchLogs`Il tipo di sink invia in streaming record di log e di eventi a Amazon CloudWatch Logs. Puoi visualizzare i log nella console di gestione AWS o elaborarli tramite altre fasi di una pipeline di dati. I dati vengono inviati in streaming a un flusso di log denominato configurato nei CloudWatch Logs. I flussi di log sono organizzati in gruppi di log denominati. Per ulteriori informazioni, consulta la .Amazon CloudWatch Logs: .

Di seguito è riportato un esempio della dichiarazione sink CloudWatch Logs:

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

Tutte le dichiarazioni sink `CloudWatchLogs` devono offrire le seguenti coppie chiave-valore aggiuntive:

`SinkType`

Deve essere la stringa letterale `CloudWatchLogs`.

`LogGroup`

Specifica il nome del gruppo di log di CloudWatch Logs contenente il flusso di log che riceve i record di eventi e di log in streaming dalla proprietà di`CloudWatchLogs`tipo di lavello. Se il gruppo di log specificato non esiste, Kinesis Agent per Windows tenta di crearlo.

`LogStream`

Specifica il nome del flusso di log CloudWatch Logs che riceve il flusso di record di eventi e di log dal`CloudWatchLogs`tipo di lavello. Questo valore supporta la sostituzione delle variabili sink. Per ulteriori informazioni, consulta Configurazione di sostituzione di variabili sink. Se il flusso di log specificato non esiste, Kinesis Agent per Windows tenta di crearlo.

L'impostazione predefinita di `BufferInterval` per questo tipo di sink è di 1 secondo, e l'impostazione predefinita di `BufferSize` è di 500 record. La dimensione massima del buffer è di 10.000 record.

Locale**`FileSystem`**Configurazione sink

Il tipo di lavello`FileSystem`salva i record di log ed eventi in un file nel file system locale invece di inviarli in streaming ai servizi AWS.`FileSystem`sono utili per test e diagnostica. Ad esempio, è possibile utilizzare questo tipo di sink per esaminare i record prima di inviarli a AWS.

con`FileSystem`, è anche possibile utilizzare i parametri di configurazione per simulare batch, limitazione e retry-on-error per simulare il comportamento dei sink AWS effettivi.

Tutti i record di tutte le origini connesse a un`FileSystem`vengono salvati nel singolo file specificato come`FilePath`: . Se`FilePath`non viene specificato, i record vengono salvati in un file denominato*`SinkId`*`.txt`nella`%TEMP%`, che di solito è`C:\Users\`*`UserName`*`\AppData\Local\Temp`, dove*`SinkId`*è l'identificatore univoco del sink e*`UserName`*è il nome utente di Windows dell'utente attivo.

Questo tipo di sink supporta gli attributi di decorazione del testo. Per ulteriori informazioni, consulta Configurazione delle decorazioni sink.

Un esempio`FileSystem`Nell'esempio seguente viene visualizzata la seguente configurazione.

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

La`FileSystem`è costituito dalle seguenti coppie chiave-valore.

`SinkType`

Deve essere la stringa letterale `FileSystem`.

`FilePath`

Specifica il percorso e il file in cui vengono salvati i record. Questa coppia chiave-valore è facoltativa. Se il valore non viene specificato, viene usato*TempPath*`\\`*SinkId*`.txt`, dove*TempPath*è la cartella memorizzata nella`%TEMP%`Variabile e*SinkId*è l'identificatore univoco del sink.

`Format`

Specifica il formato dell'evento da`json`o`xml`: . Questa coppia di valori è facoltativa e senza distinzione tra maiuscole e minuscole Se omesso, gli eventi vengono scritti nel file in testo normale.

`TextDecoration`

Si applica solo agli eventi scritti in testo normale. Questa coppia chiave-valore è facoltativa.

`ObjectDecoration`

Si applica solo agli eventi in cui`Format`è impostato su `.json`: . Questa coppia chiave-valore è facoltativa.

Utilizzo avanzato: simulazione di limitazione dei record e guasti

`FileSystem`può imitare il comportamento dei sink AWS simulando la limitazione dei record. È possibile utilizzare le seguenti coppie chiave-valore per specificare gli attributi di limitazione dei record e di simulazione di errori.

Acquisendo un blocco sul file di destinazione e impedendo le scritture su di esso, è possibile utilizzare`FileSystem`per simulare ed esaminare il comportamento dei sink AWS quando la rete fallisce.

L'esempio seguente mostra un`FileSystem`con attributi di simulazione.

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

```
}
```

`RequestsPerSecond`

Facoltativo e specificato come tipo di stringa. Se omesso, viene usato il valore predefinito`"5"`: . Controlla la frequenza delle richieste che il sink elabora, ovvero scrive sul file, non il numero di record. Kinesis Agent per Windows effettua richieste batch agli endpoint AWS, pertanto una richiesta può contenere più record.

`BufferSize`

Facoltativo e specificato come tipo di stringa. Specifica il numero massimo di record di eventi che il sink batch prima di salvare nel file.

`MaxBatchSize`

Facoltativo e specificato come tipo di stringa. Specifica la quantità massima di dati del record di eventi in byte che il sink batch prima di salvare nel file.

Il limite massimo di velocità di registrazione è una funzione di`BufferSize`, che determina il numero massimo di record per richiesta e`RequestsPerSecond`: . È possibile calcolare il limite di velocità record al secondo utilizzando la formula seguente.

RecordRate=`BufferSize`*`RequestsPerSecond`

Dati i valori di configurazione nell'esempio precedente, esiste una velocità massima di record di 1000 record al secondo.

Configurazione di sicurezza del sink

Configurazione dell'autenticazione

Per consentire a Kinesis Agent per Windows di trasmettere in streaming log, eventi e parametri ai servizi AWS, l'accesso deve essere autenticato. Vi sono diversi modi per fornire l'autenticazione per Kinesis Agent per Windows. Il metodo scelto dipende dalla situazione in cui è in esecuzione Kinesis Agent per Windows e i requisiti di sicurezza specifici di una determinata organizzazione.

- Se Kinesis Agent per Windows è in esecuzione su un host Amazon EC2, il modo più semplice e sicuro per fornire l'autenticazione è creare un ruolo IAM con accesso sufficiente alle operazioni richieste per i servizi AWS necessari e un profilo dell'istanza EC2 che fa riferimento a tale

ruolo. Per ulteriori informazioni su come creare i profili dell'istanza, consulta Uso dei profili dell'istanza. Per ulteriori informazioni sulle policy da collegare al ruolo IAM, consultaConfigurazione dell'autorizzazione: .

Dopo aver creato il profilo dell'istanza, è possibile associarlo a qualsiasi istanza EC2 che utilizza Kinesis Agent per Windows. Se le istanze hanno già un profilo di istanza associato, è possibile allegare policy appropriate al ruolo associato a quel profilo dell'istanza.

- Se Kinesis Agent per Windows viene eseguito su un host EC2 in un account, ma le risorse che sono la destinazione del sink risiedono in un altro account, è possibile creare un ruolo IAM per l'accesso a più account. Per ulteriori informazioni, consultaTutorial: Delegare l'accesso agli account AWS tramite ruoli IAM: . Dopo aver creato il ruolo tra più account, specificare l'ARN (Amazon Resource Name) per il ruolo tra più account come valore dell'opzione`RoleARN`Coppia chiave-valore nella dichiarazione sink. Kinesis Agent per Windows tenta quindi di assumersi il ruolo specificato tra più account per l'accesso alle risorse AWS associate al tipo di sink per quel sink.
- Se Kinesis Agent per Windows è in esecuzione al di fuori di Amazon EC2 (ad esempio, in locale), esistono diverse opzioni:
  - Se è accettabile registrare il server locale o il computer desktop come istanza gestita da Amazon EC2 Systems Manager, utilizzare la seguente procedura per configurare l'autenticazione:
    1. Utilizzare il processo descritto in Impostazione di AWS Systems Manager in ambienti ibridi per creare un ruolo di servizio, creare un'attivazione per un'istanza gestita e installare l'agente SSM.
    2. Collegare le policy appropriate per il ruolo del servizio per consentire a Kinesis Agent per Windows di accedere alle risorse necessarie per lo streaming dei dati dai sink configurati. Per ulteriori informazioni sulle policy da collegare al ruolo IAM, consultaConfigurazione dell'autorizzazione: .
    3. Utilizzare il processo descritto inConfigurazione di`ProfileRefreshingAWSCredentialProvider`Per aggiornare le credenziali AWSPer aggiornare le credenziali AWS.

    Questo è l'approccio consigliato per le istanze non EC2, perché le credenziali sono gestite in modo sicuro da SSM e AWS.
  - Se è accettabile eseguire il servizio AWSKinesisTap per Kinesis Agent per Windows con un determinato utente anziché con l'account di sistema predefinito, utilizzare la seguente procedura:
    1. Creare un utente IAM nell'account AWS in cui i servizi AWS saranno utilizzati. Acquisire la chiave di accesso e la chiave segreta di questo utente durante il processo di creazione. Queste informazioni saranno necessarie più avanti in questa procedura.

2. Collegare i criteri all'utente IAM che autorizza l'accesso alle operazioni necessarie per i servizi richiesti. Per ulteriori informazioni sulle policy da collegare all'utente IAM, consultaConfigurazione dell'autorizzazione: .
3. Cambiare il servizio AWSKinesisTap su ogni desktop o server in modo che venga eseguito con un determinato utente piuttosto che con l'account predefinito del sistema.
4. Creare un profilo nell'archivio SDK utilizzando la chiave di accesso e la chiave segreta registrate in precedenza. Per ulteriori informazioni, vedi Configurazione delle credenziali AWS.
5. Aggiornare il file `AWSKinesisTap.exe.config` nella directory `%PROGRAMFILES%\Amazon\AWSKinesisTap` per specificare il nome del profilo creato nel passo precedente. Per ulteriori informazioni, vedi Configurazione delle credenziali AWS.

   Questo approccio è consigliato per host non EC2 che non possono essere istanze gestite poiché le credenziali sono crittografate per l'host specifico e l'utente specifico.

- Se è necessario eseguire il servizio AWSKinesisTap per Kinesis Agent per Windows con l'account di sistema predefinito, è necessario utilizzare un file di credenziali condiviso. Questo perché l'account di sistema non ha un profilo utente Windows per l'abilitazione dell'archivio SDK. I file di credenziali condivise non sono crittografati, perciò non consigliamo di adottare questo approccio. Per informazioni su come utilizzare i file di configurazione condivisi, consultaConfigurazione delle credenziali AWSnellaSDK AWS per .NET: . Se si utilizza questo approccio, è consigliabile utilizzare la crittografia NTFS e l'accesso ai file limitato per il file di configurazione condiviso. Le chiavi devono essere ruotate da una piattaforma di gestione e il file di configurazione condiviso deve essere aggiornato quando si verifica la rotazione della chiave.

Anche se è possibile fornire direttamente le chiavi di accesso e le chiavi segrete nelle dichiarazioni dei sink, questo approccio è sconsigliato perché le dichiarazioni non sono crittografate.

Configurazione dell'autorizzazione

Collegare le policy appropriate per l'utente o il ruolo IAM che verrà utilizzato da Kinesis Agent per Windows per trasmettere i dati ai servizi AWS:

Kinesis Data Streams

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
```

```
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord",
                "kinesis:PutRecords"
            ],
            "Resource": "arn:aws:kinesis:*:*:stream/*"
        }
    ]
}
```

Per limitare l'autorizzazione a una regione, un account o un nome di flusso specifici, sostituire gli asterischi appropriati nell'ARN con valori specifici. Per ulteriori informazioni, vedere "Amazon Resource Name (ARN) per flussi di dati di Kinesis" in Controllo degli accessi alle risorse dei flussi di dati di Amazon Kinesis tramite IAM.

Kinesis Data Firehose

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Resource": "arn:aws:firehose:*:*:deliverystream/*"
        }
    ]
}
```

Per limitare l'autorizzazione a una regione, un account o un flusso di distribuzione specifici, sostituire gli asterischi appropriati nell'ARN con valori specifici. Per ulteriori informazioni, consultaControllare gli accessi con Amazon Kinesis Data FirehosenellaGuida per sviluppatori di Amazon Kinesis Data Firehose: .

CloudWatch

```
{
```

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

Per ulteriori informazioni, consultaPanoramica sulla gestione delle autorizzazioni di accesso alle risorse CloudWatchnellaAmazon CloudWatch Logs: .

CloudWatch Logs con un gruppo di log e di flussi di log

```
{
    "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:*:*:*"
        }
    ]
}
```

Per limitare l'accesso a una regione, un account, un gruppo o un flusso di log specifico, sostituire gli asterischi opportuni negli ARN con i valori appropriati. Per ulteriori informazioni, consultaPanoramica

sulla gestione delle autorizzazioni di accesso alle risorse CloudWatch LogsnellaAmazon CloudWatch Logs: .

CloudWatch Logs con autorizzazioni aggiuntive per Kinesis Agent per Windows per creare gruppi di log e flussi di log

```
{
    "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": "*"
        }
    ]
}
```

Per limitare l'accesso a una regione, un account, un gruppo o un flusso di log specifico, sostituire gli asterischi opportuni negli ARN con i valori appropriati. Per ulteriori informazioni, consultaPanoramica sulla gestione delle autorizzazioni di accesso alle risorse CloudWatch LogsnellaAmazon CloudWatch Logs: .

Le autorizzazioni necessarie per l'espansione delle variabili dei tag EC2

L'utilizzo dell'espansione delle variabili con il prefisso della variabile `ec2tag` richiede l'autorizzazione `ec2:Describe*`.

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

> **Note**
>
> È possibile abbinare più dichiarazioni in una singola policy finché il `Sid` per ogni istruzione è univoco all'interno di tale policy. Per informazioni sulla creazione di policy, consultaCreazione di policy IAMnellaGuida per l'utente di IAM: .

Configurazione di**`ProfileRefreshingAWSCredentialProvider`**Per aggiornare le credenziali AWS

Se si utilizza AWS Systems Manager per ambienti ibridi per gestire le credenziali AWS, Systems Manager ruota le credenziali di sessione in`c:\Windows\System32\config\systemprofile\.aws\credentials`: . Per ulteriori informazioni su Systems Manager per gli ambienti ibridi, consultaConfigurazione di AWS Systems Manager per ambienti ibridinellaAWS Systems Manager: .

Poiché AWS .net SDK non rileva automaticamente nuove credenziali, forniamo il`ProfileRefreshingAWSCredentialProvider`per aggiornare le credenziali.

Puoi utilizzare`CredentialRef`di qualsiasi configurazione di sincronizzazione AWS per fare riferimento a`Credentials`in cui viene`CredentialType`L'attributo è impostato su`ProfileRefreshingAWSCredentialProvider`Come mostrato nell'esempio seguente.

```
{
    "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
    }]
}
```

Una definizione di credenziali è costituita dai seguenti attributi come coppie chiave-valore.

`Id`

Definisce la stringa che le definizioni di sink possono specificare utilizzando`CredentialRef`per fare riferimento a questa configurazione delle credenziali.

`CredentialType`

Impostare sulla stringa letterale`ProfileRefreshingAWSCredentialProvider`: .

`Profile`

Facoltativo. Il valore di default è `default`.

`FilePath`

Facoltativo. Specifica il percorso per il file delle credenziali AWS. Se omesso, il valore predefinito è `%USERPROFILE%/.aws/credentials`.

`RefreshingInterval`

Facoltativo. Frequenza con cui le credenziali vengono aggiornate, in secondi. Se omesso, il valore predefinito è `300`.

Configurazione delle decorazioni sink

Le dichiarazioni sink possono facoltativamente includere coppie chiave-valore che specificano dati aggiuntivi per lo streaming ai vari servizi AWS per aumentare i record raccolti dall'origine.

`TextDecoration`

Utilizzare questa coppia chiave-valore quando nessun `Format` viene specificato nella dichiarazione sink. Il valore è una stringa con formato speciale in cui si verifica la sostituzione delle variabili. Ad esempio, supponiamo che `TextDecoration` di `"{ComputerName}:::{timestamp:yyyy-MM-dd HH:mm:ss}:::{_record}"` venga fornito per un sink. Quando un'origine emette un record di log che contiene il testo `The system has resumed from sleep.` e tale origine è connessa al sink tramite una pipe, allora il testo `MyComputer1:::2017-10-26 06:14:22:::The system has resumed from sleep.` viene inviato in streaming al servizio AWS associato al tipo di sink. La variabile `{_record}` si riferisce al record del testo originale distribuito dall'origine.

`ObjectDecoration`

Utilizzare questa coppia chiave-valore quando `Format` viene specificato nella dichiarazione del sink per aggiungere ulteriori dati prima della serializzazione dei record. Ad esempio, supponiamo che una `ObjectDecoration` di `"ComputerName={ComputerName};DT={timestamp:yyyy-MM-dd HH:mm:ss}"` venga fornita per un sink che specifica JSON `Format`. Il risultante JSON in streaming al servizio AWS associato al tipo di sink include le seguenti coppie chiave-valore in aggiunta ai dati originali dell'origine:

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

Per un esempio di utilizzo di `ObjectDecoration`, vedi Tutorial: Trasmetti file di registro JSON ad Amazon S3 utilizzando Kinesis Agent per Windows.

`ObjectDecorationEx`

Specifica un'espressione, che consente l'estrazione e la formattazione dei dati più flessibili rispetto a`ObjectDecoration`: . Questo campo può essere utilizzato quando il formato del sink è`json`: . La sintassi dell'espressione è illustrata di seguito.

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

Ad esempio,`ObjectDecorationEx`Attributo

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

Trasforma il record letterale:

```
System log message
```

In un oggetto JSON come segue, con i valori restituiti dalle espressioni:

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

Per ulteriori informazioni sulla formulazione di espressioni, consultaSuggerimenti per scrivere espressioni: . La maggior parte dei`ObjectDecoration`dovrebbe funzionare usando la nuova sintassi con l'eccezione delle variabili timestamp. A`{timestamp:yyyyMMdd}`Campo in`ObjectDecoration`è espresso come`{format(_timestamp,'yyyyMMdd')}`in`ObjectDecorationEx`: .

`TextDecorationEx`

Specifica un'espressione, che consente l'estrazione e la formattazione dei dati più flessibili rispetto a`TextDecoration`come mostrato nell'esempio seguente.

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

Puoi utilizzare`TextDecorationEx`per comporre oggetti JSON. Utilizzare '@ {'per sfuggire alla parentesi graffa aperta, come mostrato nell'esempio seguente.

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

Se il tipo di origine dell'origine connessa al sink è `DirectorySource`, allora il sink può utilizzare tre variabili aggiuntive:

`_FilePath`

Il percorso completo al file di log.

`_FileName`

Il nome file e l'estensione del nome file del file.

`_Position`

Un numero intero che rappresenta dove si trova il record nel file di log.

Queste variabili sono utili quando si utilizza un'origine che raccoglie record di log da più file connessi a un sink che invia tutti i record a un solo flusso. Inserire i valori di queste variabili nel record di streaming consente di effettuare analisi downstream nella pipeline di dati per ordinare i record per file e per ubicazione all'interno di ciascun file.

Suggerimenti per scrivere espressioni

Un'espressione può essere qualsiasi tra i seguenti:

- Espressione variabile.
- Un'espressione costante, ad esempio`'hello'`,`1`,`1.21`,`null`,`true`,`false`: .
- Espressione di chiamata che chiama una funzione, come mostrato nell'esempio seguente.

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

Caratteri speciali

Per sfuggire a caratteri speciali sono necessari due barre rovesciate.

Nesting

Le chiamate di funzione possono essere nidificate, come mostrato nell'esempio seguente.

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

Variables

Ci sono tre tipi di variabili: locale, meta e globale.

- Variabili localiInizia con un'`$come$message`: . Vengono utilizzati per risolvere la proprietà dell'oggetto evento, una voce se l'evento è un dizionario o un attributo se l'evento è un oggetto JSON. Se la variabile locale contiene spazio o caratteri speciali, utilizzare una variabile locale tra virgolette, ad esempio`$'date created'`: .
- Variabili di metaInizia con un trattino basso (_) e vengono utilizzati per risolvere i metadati dell'evento. Tutti i tipi di eventi supportano le seguenti meta-variabili.

  `_timestamp`

  Il timestamp dell'evento.

  `_record`

  La rappresentazione in formato non elaborato dell'evento.

  Gli eventi di registro supportano le seguenti meta-variabili aggiuntive.

  `_filepath`

  `_filename`

  `_position`

  `_linenumber`

- Variabili globalirisolvere le variabili di ambiente, i metadati dell'istanza EC2 o EC2tag. Per prestazioni migliori, ti consigliamo di utilizzare il prefisso per limitare l'ambito di ricerca, ad esempio`{env:ComputerName}`,`{ec2:InstanceId}`, e`{ec2tag:Name}`: .

Funzioni integrate

Kinesis Agent per Windows supporta le seguenti funzioni integrate. Se uno qualsiasi degli argomenti è`NULL`e la funzione non è progettata per gestire`NULL`, un`NULL`viene restituito.

```
//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)
```

Configurazione di sostituzione di variabili sink

Le dichiarazioni sink `KinesisStream`, `KinesisFirehose` e `CloudWatchLogs` richiedono una coppia chiave-valore `LogStream` o `StreamName`. Il valore di queste coppie chiave-valore può contenere riferimenti a variabili che vengono automaticamente risolti da Kinesis Agent per Windows. Per`CloudWatchLogs`, il`LogGroup`È richiesta anche una coppia chiave-valore e può contenere riferimenti a variabili automaticamente risolti da Kinesis Agent per Windows. Le variabili sono specificate utilizzando il modello {*`prefix`*:*`variablename`*} in cui *`prefix`*: è facoltativo. I prefissi supportati sono i seguenti:

- `env`— Il riferimento alla variabile viene risolto dal valore della variabile di ambiente con lo stesso nome.

- `ec2`— Il riferimento alla variabile viene risolto dai metadati dell'istanza EC2 con lo stesso nome.
- `ec2tag`— Il riferimento alla variabile viene risolto dal valore del tag dell'istanza EC2 con lo stesso nome. L'autorizzazione `ec2:Describe*` è obbligatoria per accedere ai tag di istanza. Per ulteriori informazioni, consulta Le autorizzazioni necessarie per l'espansione delle variabili dei tag EC2.

Se il prefisso non è specificato, se c'è una variabile di ambiente con lo stesso nome di `variablename`, il riferimento alla variabile viene risolto dal valore della variabile di ambiente. Altrimenti, se `variablename` è `instance_id` o `hostname`, il riferimento alla variabile viene risolto dal valore dei metadati EC2 con lo stesso nome. In caso contrario, il riferimento alla variabile non viene risolto.

Di seguito sono elencati alcuni esempi di coppie chiave-valore valide che utilizzano i riferimenti alle variabili:

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

Le dichiarazioni sink `CloudWatchLogs` supportano una variabile timestamp con un formato speciale che consente al timestamp del record dell'evento o del log originale dell'origine di modificare il nome del flusso di log. Il formato è `{timestamp:timeformat}`. Fai riferimento al file di esempio seguente:

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

Se il record di eventi o di log è stato generato il 5 giugno 2017, il valore della coppia chiave-valore `LogStream` dell'esempio precedente si risolverebbe in `"LogStream_20170605"`.

Se autorizzato, il tipo di sink `CloudWatchLogs` è in grado di creare automaticamente nuovi flussi di log generati quando richiesto in base ai nomi. Non è possibile eseguire questa operazione per altri tipi di sink perché richiedono un'ulteriore configurazione oltre il nome del flusso.

Non ci sono sostituzioni specifiche di variabili che si verificano nel testo e nella decorazione dell'oggetto. Per ulteriori informazioni, consulta Configurazione delle decorazioni sink.

Configurazione dell'accodamento dei sink

Le dichiarazioni dei sink `KinesisStream`, `KinesisFirehose` e `CloudWatchLogs` possono facoltativamente abilitare l'accodamento di record che non hanno lo streaming per il servizio AWS associato a questi tipi di sink a causa di problemi di connessione transitori. Per abilitare nuovi tentativi di streaming automatico e di accodamento quando la connettività viene ripristinata, utilizzare le seguenti coppie chiave-valore nelle dichiarazioni sink:

`QueueType`

Specifica il tipo di meccanismo di accodamento da utilizzare. L'unico valore supportato è `file`, che indica che i record devono essere messi in coda in un file. Questa coppia chiave-valore è necessaria per abilitare la funzione di accodamento di Kinesis Agent per Windows. Se non è specificato, il comportamento predefinito consiste nell'accodare solo in memoria e lo streaming dà esito negativo quando vengono raggiunti i limiti di accodamento in memoria.

`QueuePath`

Specificare il percorso alla cartella che contiene i file di record in coda. Questa coppia chiave-valore è facoltativa. Il valore predefinito è `%PROGRAMDATA%\KinesisTap\Queue\SinkId` dove SinkId è l'identificatore assegnato come valore di `Id` per la dichiarazione del sink.

`QueueMaxBatches`

Limita la quantità totale di spazio che Kinesis Agent per Windows può consumare quando si verifica l'accodamento dei record per lo streaming. La quantità di spazio è limitato al valore di questa coppia chiave-valore moltiplicato per il numero massimo di byte per batch. Il numero massimo di byte per batch per i tipi di sink `KinesisStream`, `KinesisFirehose` e `CloudWatchLogs` sono, rispettivamente, 5 MB, 4 MB e 1 MB. Quando viene raggiunto il limite, qualsiasi tentativo non riuscito di streaming non viene messo in coda e viene segnalato come tentativo non recuperabile. Questa coppia chiave-valore è facoltativa. Il valore predefinito è 10.000 batch.

Configurazione di un Proxy per i sink

Per configurare un proxy per tutti i tipi di sink di Kinesis Agent per Windows che accedono ai servizi AWS, modificare il file di configurazione dell'Agente Kinesis per Windows che si trova in`%Program Files%\Amazon\KinesisTap\AWSKinesisTap.exe.config`: . Per istruzioni, consulta`proxy`SezioneDocumentazione di riferimento per AWS SDK for .NETnellaGuida per gli sviluppatori dell'SDK AWS per .NET: .

Configurazione delle variabili di risoluzione in più attributi sink

L'esempio seguente mostra una configurazione sink che utilizza ilRegionVariabile di ambiente per il valore delRegionCoppia chiave-valore dell'attributo. PerRoleARN, specifica la chiave del tag EC2MyRoleARN, che restituisce il valore associato a quella chiave.

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

Configurazione degli endpoint regionali di AWS STS quando si utilizza la proprietà RoleARN nei sink AWS

Questa funzione si applica solo se utilizzi KinesiStap su Amazon EC2 e utilizzi ilRoleARNdei sink AWS per assumere un ruolo IAM esterno per l'autenticazione con i servizi AWS di destinazione.

ImpostandoUseSTSRegionalEndpointsDa atrue, è possibile specificare che un agente utilizzi l'endpoint regionale (ad esempiohttps://sts.us-east-1.amazonaws.com) invece dell'endpoint globale (ad esempio,https://sts.amazonaws.com). L'utilizzo di un endpoint STS regionale riduce la latenza di andata e ritorno per l'operazione e limita l'impatto degli errori nel servizio endpoint globale.

Configurazione di VPC Endpoint per i sink AWS

È possibile specificare un endpoint VPC nella configurazione sink perCloudWatchLogs,CloudWatch,KinesisStreams, eKinesisFirehosetipi di lavello. Un endpoint VPC consente di connettere privatamente il VPC a servizi AWS supportati e servizi endpoint VPC powered by AWS PrivateLink senza richiedere un Internet gateway, un dispositivo NAT, una connessione VPN o una connessione AWS Direct Connect. Le istanze nel VPC non richiedono indirizzi IP pubblici per comunicare con risorse nel servizio. Il traffico tra il VPC e gli altri servizi non lascia la rete Amazon. Per ulteriori informazioni, consultaEndpoint VPCnellaGuida per l'utente di Amazon VPC: .

È possibile specificare l'endpoint VPC utilizzando il comandoServiceURLcome mostrato nell'esempio seguente di un'CloudWatchLogsconfigurazione del sink. Impostare il

valore`ServiceURL`al valore mostrato nella schedaDettagli dell'endpoint VPCcon la console Amazon VPC.

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

Configurazione di un mezzo alternativo di proxy

Questa funzionalità consente di configurare un server proxy in una configurazione sink utilizzando il supporto proxy incorporato in AWS SDK anziché .NET. In precedenza, l'unico modo per configurare l'agente per utilizzare un proxy era utilizzare una funzionalità nativa di .NET, che instradava automaticamente tutte le richieste HTTP/S tramite il proxy definito nel file proxy.

Se si sta attualmente utilizzando l'agente con un server proxy, non è necessario modificare per utilizzare questo metodo.

Puoi utilizzare`ProxyHost`e`ProxyPort`Per configurare un proxy alternativo, come mostrato nell'esempio seguente.

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

Dichiarazioni delle pipe

UtilizzaDichiarazioni delle pipeper connettere una sorgente (vedereDichiarazioni delle origini) a un lavandino (vediDichiarazioni dei sink) nell'agente Amazon Kinesis per Microsoft Windows. Una dichiarazione della pipe viene espressa come un oggetto JSON. Una volta avviato l'agente Kinesis

per Windows, i log, gli eventi o i parametri vengono raccolti dall'origine per una determinata pipe. Vengono quindi trasmessi a diversi servizi AWS utilizzando il sink associato a tale pipe.

Di seguito è riportato un esempio della dichiarazione di una pipe :

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

Argomenti

Configurazione di pipe

Tutte le dichiarazioni di pipe possono contenere le seguenti coppie chiave-valore:

`Id`

Specifica il nome della pipe (obbligatorio). Deve essere univoco all'interno del file di configurazione.

`Type`

Specifica il tipo di trasformazione (se presente) che viene applicata dalla pipe quando i dati dei log vengono trasferiti dall'origine al sink. L'unico valore supportato è `RegexFilterPipe`. Questo valore consente il filtraggio regolare di espressioni della sottostante rappresentazione testuale del record di log. L'utilizzo del filtraggio può ridurre i costi di trasmissione e di storage inviando solo alla pipeline di dati il downstream di record di log pertinenti. Questa coppia chiave-valore è facoltativa. Il valore di default è di non fornire alcuna trasformazione.

`FilterPattern`

Specifica l'espressione regolare per le pipeline `RegexFilterPipe` che consentono di filtrare record di log raccolti dall'origine prima di essere trasferite al sink. I record dei log vengono trasferiti dalle pipe di tipo `RegexFilterPipe` quando l'espressione regolare corrisponde alla rappresentazione testuale sottostante del record. Record di log strutturati, generati, ad esempio, quando si utilizza la coppia chiave-valore `ExtractionPattern` in una dichiarazione

`DirectorySource` possono comunque essere filtrati tramite il meccanismo `RegexFilterPipe`. Questo perché questo meccanismo agisce sulla rappresentazione testuale originale prima di eseguire l'analisi. Questa coppia chiave-valore è facoltativa, ma deve essere fornita se la pipe specifica il tipo `RegexFilterPipe`.

Di seguito è riportato un esempio della dichiarazione di una pipe `RegexFilterPipe`:

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

`SourceRef`

Specifica il nome (il valore della coppia chiave-valore `Id`) della dichiarazione di definizione dell'origine che raccoglie dati di log, eventi e parametri per la pipe (obbligatorio).

`SinkRef`

Specifica il nome (il valore della coppia chiave-valore `Id`) della dichiarazione di definizione del sink che raccoglie dati di log, eventi e parametri per la pipe (obbligatorio).

`IgnoreCase`

Facoltativo. Accetta valori di`true`o`false`: . Se impostato su`true`, Regex corrisponderà ai record in modo senza distinzione tra maiuscole e minuscole.

`Negate`

Facoltativo. Accetta valori di`true`o`false`: . Se impostato su`true`, la pipe inoltrerà i record cheNON FARECorrispondono all'espressione regolare.

Per un esempio di un file di configurazione completo che usa il tipo di pipe `RegexFilterPipe`, consulta Utilizzo di pipe.

Configurazione dell'agente Kinesis per le pipe metriche di Windows

Vi è una sorgente di metrica integrata denominata`_KinesisTapMetricsSource`che produce metriche sull'agente Kinesis per Windows. Se è presente un`CloudWatch`dichiarazione sink con un`Id`di`MyCloudWatchSink`La seguente dichiarazione di pipeline di Kinesis isce i parametri generati da da Windows a tale sink:

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

Per ulteriori informazioni sull'origine dei parametri integrati di Kinesis Agent per Windows, consultaOrigine dei parametri integrati di Kinesis Agent per Windows: .

Se il file di configurazione invia anche i parametri contatori delle prestazioni, si consiglia di utilizzare una pipe e un sink separati anziché utilizzare lo stesso sink sia Kinesis parametri sia per i parametri contatori delle prestazioni di Windows.

Configurazione di aggiornamenti automatici

Utilizzo dell'`appsettings.json`File di configurazione per abilitare gli aggiornamenti automatici di Amazon Kinesis Agent per Microsoft Windows e il file di configurazione per Kinesis Agent per Windows. Per controllare il comportamento di aggiornamento, specificare la coppia chiave-valore `Plugins` allo stesso livello nel file di configurazione di `Sources`, `Sinks` e `Pipes`.

La coppia chiave-valore `Plugins` specifica le funzionalità generali aggiuntive da utilizzare che non rientrano specificamente nelle categorie di origini, sink e pipe. Ad esempio, c'è un plug-in per l'aggiornamento di Kinesis Agent per Windows e c'è un plug-in per l'aggiornamento del file`appsettings.json`File di configurazione. I plug-in sono rappresentati come oggetti JSON e hanno sempre a disposizione una coppia chiave-valore `Type`. `Type` definisce le altre coppie chiave-valore che possono essere specificate per il plug-in. Sono supportati i seguenti tipi di plug-in:

`PackageUpdate`

Specifica che Kinesis Agent per Windows deve controllare periodicamente un file di configurazione della versione del pacchetto. Se il file della versione del pacchetto indica che deve essere installata un'altra versione di Kinesis Agent per Windows, allora Kinesis Agent per

Windows scarica tale versione e la installa. Le coppie chiave-valore del plug-in `PackageUpdate` includono:

`Type`

Il valore deve essere la stringa `PackageUpdate` ed è obbligatorio.

`Interval`

Specifica la frequenza con cui verificare il file della versione del pacchetto per eventuali modifiche in pochi minuti rappresentate come stringa. Questa coppia chiave-valore è facoltativa. Se non specificato, il valore predefinito è 60 minuti. Se il valore è inferiore a 1, non è possibile verificare gli aggiornamenti.

`PackageVersion`

Specifica la posizione del file JSON della versione del pacchetto. Il file può risiedere su una condivisione di file (`file://`), un sito web (`http://`), o Amazon S3 (`s3://`). Ad esempio, un valore compreso tra e rappresenta`s3://mycompany/config/agent-package-version.json`indica che Kinesis Agent per Windows deve controllare il contenuto della`config/agent-package-version.json`file nel file`mycompany`Bucket Amazon S3. È necessario eseguire gli aggiornamenti in base al contenuto del file.

> **Note**
>
> Il valore della proprietà`PackageVersion`coppia chiave-valore effettua la distinzione tra maiuscole Amazon S3 uscole

Di seguito è riportato un esempio dei contenuti di un file della versione di un pacchetto:

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

La`Version`Specifica la versione di Kinesis Agent per Windows da installare se non è già installata. La variabile di riferimento `{Version}` in `PackageUrl` risolve il valore specificato per la coppia chiave-valore `Version`. In questo esempio, la variabile restituisce la stringa `1.0.0.106`. Questa risoluzione di variabile viene fornita in modo che ci sia un'unica

posizione della versione del pacchetto di file laddove la versione desiderata specificata viene memorizzata. È possibile utilizzare più file della versione del pacchetto per controllare il ritmo di implementazione di nuove versioni di Kinesis Agent per Windows e convalidare una nuova versione prima di un'implementazione di dimensioni maggiori. Per eseguire il rollback di una distribuzione di Kinesis Agent per Windows, modificare uno o più file della versione del pacchetto per specificare una versione precedente di Kinesis Agent per Windows che funziona correttamente nell'ambiente.

Il valore della coppia chiave-valore `PackageVersion` subisce la sostituzione delle variabili per facilitare la selezione automatica dei file della versione di pacchetti differenti. Per ulteriori informazioni sulla sostituzione delle variabili, consulta Configurazione di sostituzione di variabili sink.

`AccessKey`

Specifica la chiave di accesso da utilizzare per autenticare l'accesso al file della versione del pacchetto in Amazon S3. Questa coppia chiave-valore è facoltativa. Non è consigliabile utilizzare questa coppia chiave-valore. Per gli approcci di autenticazione alternativi raccomandati, consulta Configurazione dell'autenticazione.

`SecretKey`

Specifica la chiave segreta da utilizzare per autenticare l'accesso al file della versione del pacchetto in Amazon S3. Questa coppia chiave-valore è facoltativa. Non è consigliabile utilizzare questa coppia chiave-valore. Per gli approcci di autenticazione alternativi raccomandati, consulta Configurazione dell'autenticazione.

`Region`

Specifica l'endpoint della regione da utilizzare per accedere al file della versione del pacchetto da Amazon S3. Questa coppia chiave-valore è facoltativa.

`ProfileName`

Specifica il profilo di sicurezza da utilizzare per autenticare l'accesso al file della versione del pacchetto in Amazon S3. Per ulteriori informazioni, consulta Configurazione dell'autenticazione. Questa coppia chiave-valore è facoltativa.

`RoleARN`

Specifica quale ruolo assumere durante l'autenticazione dell'accesso al file della versione del pacchetto in Amazon S3 in uno scenario tra più account. Per ulteriori informazioni, consulta Configurazione dell'autenticazione. Questa coppia chiave-valore è facoltativa.

Se nessun plug-in `PackageUpdate` viene specificato, allora nessun file della versione del pacchetto viene controllato per stabilire se un aggiornamento è obbligatorio.

`ConfigUpdate`

Specifica che Kinesis Agent per Windows deve controllare periodicamente la presenza di`appsettings.json`File di configurazione archiviato in una condivisione di file, sito Web o Amazon S3. Se un file di configurazione aggiornato esiste, viene scaricato e installato da Kinesis Agent per Windows.`ConfigUpdate`Le coppie chiave-valore includono quanto segue:

`Type`

Il valore deve essere la stringa `ConfigUpdate` ed è obbligatorio.

`Interval`

Specifica la frequenza con cui verificare un nuovo file di configurazione in pochi minuti rappresentato come stringa. Questa coppia chiave-valore è facoltativa e, se non è specificato, il valore predefinito sarà 5 minuti. Se il valore è inferiore a 1, l'aggiornamento del file di configurazione non viene selezionato.

`Source`

Specifica dove cercare un file di configurazione aggiornato. Il file può risiedere su una condivisione di file (`file://`), un sito web (`http://`), o Amazon S3 (`s3://`). Ad esempio, un valore compreso tra e rappresenta`s3://mycompany/config/appsettings.json`indica che Kinesis Agent per Windows deve verificare la disponibilità di aggiornamenti per`config/appsettings.json`file nel file`mycompany`Bucket Amazon S3.

> **Note**
>
> Il valore della proprietà`Source`Specifica la distinzione tra maiuscole e minuscole per Amazon S3.

Il valore della coppia chiave-valore `Source` subisce la sostituzione delle variabili per facilitare la selezione automatica di file di configurazione differenti. Per ulteriori informazioni sulla sostituzione delle variabili, consulta Configurazione di sostituzione di variabili sink.

`Destination`

Specifica dove archiviare il file di configurazione nel computer locale. Questo può essere un percorso relativo, un percorso assoluto o un percorso contenente riferimenti a

variabili di ambiente, ad esempio `%PROGRAMDATA%`. Se il percorso è relativo, è relativo al percorso in cui è installato Kinesis Agent per Windows. In genere, il valore deve essere `.\appsettings.json`. Questa coppia chiave-valore è obbligatoria.

`AccessKey`

Specifica la chiave di accesso da utilizzare per autenticare l'accesso al file di configurazione in Amazon S3. Questa coppia chiave-valore è facoltativa. Non è consigliabile utilizzare questa coppia chiave-valore. Per gli approcci di autenticazione alternativi raccomandati, consulta Configurazione dell'autenticazione.

`SecretKey`

Specifica la chiave segreta da utilizzare per autenticare l'accesso al file di configurazione in Amazon S3. Questa coppia chiave-valore è facoltativa. Non è consigliabile utilizzare questa coppia chiave-valore. Per gli approcci di autenticazione alternativi raccomandati, consulta Configurazione dell'autenticazione.

`Region`

Specifica l'endpoint della regione da utilizzare per accedere al file di configurazione da Amazon S3. Questa coppia chiave-valore è facoltativa.

`ProfileName`

Specifica il profilo di sicurezza da utilizzare per autenticare l'accesso al file di configurazione in Amazon S3. Per ulteriori informazioni, consulta Configurazione dell'autenticazione. Questa coppia chiave-valore è facoltativa.

`RoleARN`

Specifica quale ruolo assumere durante l'autenticazione dell'accesso al file di configurazione in Amazon S3 in uno scenario tra più account. Per ulteriori informazioni, consulta Configurazione dell'autenticazione. Questa coppia chiave-valore è facoltativa.

Se nessun plug-in `ConfigUpdate` viene specificato, allora nessun file di configurazione viene controllato per stabilire se un aggiornamento del file di configurazione è obbligatorio.

Il seguente file di configurazione di esempio `appsettings.json` dimostra l'utilizzo dei plug-in `PackageUpdate` e `ConfigUpdate`. In questo esempio, c'è un file della versione del pacchetto che si trova nel file`mycompany`Bucket Amazon S3 denominato`config/agent-package-version.json`: . Questo file è selezionato per eventuali modifiche circa ogni 2 ore. Se un'altra

versione di Kinesis Agent per Windows è specificata nel file della versione del pacchetto, la versione dell'agente specificata viene installata dal percorso specificato nel file della versione del pacchetto.

Inoltre, c'è un file`appsettings.json`file di configurazione archiviato nel file`mycompany`Bucket Amazon S3 denominato`config/appsettings.json`: . Circa ogni 30 minuti, il file viene confrontato con l'attuale file di configurazione. Se sono diversi, il file di configurazione aggiornato viene scaricato da Amazon S3 e installato nel percorso locale tipico per il file`appsettings.json`File di configurazione.

```
{
  "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"
    }
  ]
}
```

Esempi di configurazione di Kinesis Agent per Windows

La`appsettings.json`Il file di configurazione è un documento JSON che controlla il modo in cui Amazon Kinesis Agent per Microsoft Windows raccoglie i log, gli eventi e i parametri. Controlla anche il modo in cui Kinesis Agent per Windows trasforma i dati e li trasmette ai vari servizi AWS. Per ulteriori informazioni sulle dichiarazioni di pipe, sink e origine nel file di configurazione, consulta Dichiarazioni delle origini, Dichiarazioni dei sink e Dichiarazioni delle pipe.

Le seguenti sezioni contengono esempi di file di configurazione per diversi tipi di scenari.

Argomenti

- Streaming da diverse origini a Kinesis Data Streams
- Streaming da log di eventi di applicazioni Windows ai sink
- Utilizzo di pipe
- Utilizzo di più origini e pipe

Streaming da diverse origini a Kinesis Data Streams

L'esempio seguente`appsettings.json`Il file di configurazione mostrano i log di streaming e gli eventi da origini diverse a Kinesis Data Streams e da contatori di performance Windows a parametri Amazon CloudWatch.

Parser di record **`DirectorySource`**, **`SysLog`**

Il seguente file effettua lo streaming di record di log in formato syslog da tutti i file con un`.log`Estensione file nella directory`C:\LogSource\`directory alla directory`SyslogKinesisDataStream`Kinesis Data Streams nella regione us-east-1. Viene definito un segnalibro per garantire che vengano inviati tutti i dati dai file di log anche se l'agente viene arrestato e riavviato più tardi. Un'applicazione personalizzata può leggere ed elaborare i record dal flusso `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"
    }
  ]
}
```

Parser di record **`DirectorySource`**, **`SingleLineJson`**

Il seguente file effettua lo streaming di record di log in formato JSON da tutti i file con un `.log`Estensione file nella directory`C:\LogSource\`directory alla directory`JsonKinesisDataStream`Kinesis Data Streams nella regione us-east-1. Prima dello streaming, le coppie chiave-valore per le chiavi `ComputerName` e `DT` vengono aggiunte a ogni oggetto JSON, con i valori per il nome del computer e la data e l'ora in cui il record viene elaborato. Un'applicazione personalizzata può leggere ed elaborare i record dal flusso `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

Il seguente file effettua lo streaming di record di log generati da Microsoft Exchange e archiviati in file con il `.log`Estensione nella directory`C:\temp\ExchangeLog\`directory alla directory`ExchangeKinesisDataStream`Streaming di dati di kinesis nella regione us-east-1 in formato JSON. Anche se i log di Exchange non sono in formato JSON, Kinesis Agent per Windows è in grado di analizzare i log e di trasformarli in formato JSON. Prima dello streaming, le coppie chiave-valore per le chiavi `ComputerName` e `DT` vengono aggiunte a ogni oggetto JSON contenente i valori per il nome del computer e la data e l'ora in cui il record viene elaborato. Un'applicazione personalizzata può leggere ed elaborare i record dal flusso `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

Il seguente file effettua lo streaming di record di log di Internet Information Services (IIS) per Windows archiviati nel percorso standard di tali file al file di`IISKinesisDataStream`Kinesis Data Streams nella regione us-east-1. Un'applicazione personalizzata può leggere ed elaborare i record dal flusso `IISKinesisDataStream`. IIS è un server Web per 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` con Query

Il file seguente esegue flussi di eventi di registro dal registro eventi di sistema di Windows con un livello di`CriticaloError`(minore di o uguale a 2) al`SystemKinesisDataStream`Streaming di dati di kinesis nella regione us-east-1 in formato JSON. Un'applicazione personalizzata può leggere ed elaborare i record dal flusso `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`

Il seguente file effettua lo streaming dell'eccezione Microsoft Common Language Runtime (CLR) e degli eventi di sicurezza al`ClrKinesisDataStream`Streaming di dati di kinesis nella regione us-east-1 in formato JSON. Un'applicazione personalizzata può leggere ed elaborare i record dal flusso `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

Il seguente file effettua lo streaming di contatori di prestazioni per i file totali aperti, i tentativi di accesso totali dal riavvio, il numero di letture del disco al secondo e la percentuale di spazio di disco libero ai parametri CloudWatch nella regione us-east-1. Puoi tracciare un grafico di queste metriche in CloudWatch, creare dashboard dai grafici e impostare allarmi che inviano notifiche quando vengono superate soglie.

```
{
  "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 da log di eventi di applicazioni Windows ai sink

L'esempio seguente`appsettings.json`l file di configurazione mostrano lo streaming di log di eventi di applicazioni Windows a diversi sink in Amazon Kinesis Agent per Microsoft Windows. Per esempi di utilizzo dei tipi di sink `KinesisStream` e `CloudWatch`, consulta Streaming da diverse origini a Kinesis Data Streams.

`KinesisFirehose`

I seguenti flussi di file`CriticaloError`Eventi del registro applicazioni di Windows nel`WindowsLogFirehoseDeliveryStream`Streaming di distribuzione di Kinesis Data Firehose nella regione us-east-1. Se la connettività a Kinesis Data Firehose viene interrotta, gli eventi vengono prima messi in coda in memoria. Quindi, se necessario, sono in coda a un file su disco fino al ripristino della connettività. Quindi gli eventi non sono in coda e vengono inviati seguiti da eventuali nuovi eventi.

È possibile configurare Kinesis Data Firehose per memorizzare i dati in streaming in diversi tipi di storage e servizi di analisi in base ai requisiti della pipeline di dati.

```
{
  "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`

I seguenti flussi di file`CriticaloError`Registrazione degli eventi di log di applicazioni Windows ai flussi di CloudWatch Logs nella directory`MyServiceApplicationLog-Group`gruppo di log. Il nome di ogni flusso inizia con `Stream-`. Termina con l'anno di quattro cifre, il mese di due cifre e il giorno di due cifre in cui il flusso è stato creato, tutti concatenati (per esempio, `Stream-20180501` è il flusso creato il 1° maggio 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"
    }
  ]
}
```

Utilizzo di pipe

Il seguente file di configurazione di esempio `appsettings.json` dimostra l'utilizzo di funzionalità relative alle pipe.

In questo esempio viene eseguito lo streaming di voci di log da`c:\LogSource \al`a`ApplicationLogFirehoseDeliveryStream`Streaming di distribuzione di Kinesis Data Firehose. Include solo le linee che soddisfano l'espressione regolare specificata dalla coppia chiave-valore `FilterPattern`. In particolare, solo le righe del file di log che iniziano con`10`o`11`vengono trasmesse a 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),.*"
    }
  ]
}
```

Utilizzo di più origini e pipe

Il seguente file di configurazione di esempio `appsettings.json` dimostra l'utilizzo di più origini e pipe.

In questo esempio viene eseguito lo streaming di log di sistema Windows Event, dell'applicazione e di sicurezza al`EventLogStream`Il flusso di distribuzione di Kinesis Data Firehose utilizza tre origini, tre tubi e un unico lavandino.

```
{
    "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"
  }
    ]
}
```

Configurazione della Telemetria

Per abilitare un supporto migliore, per impostazione predefinita, Amazon Kinesis Agent per Microsoft Windows raccoglie le statistiche relative al funzionamento dell'agente e le invia ad AWS. Queste informazioni non contengono informazioni personali. Non include i dati raccolti o inviati ai servizi AWS. Raccogliamo circa 1-2 KB di questi dati dei parametri ogni 60 minuti.

È possibile annullare la raccolta e la trasmissione di queste statistiche. Per eseguire questa operazione, aggiungere la seguente coppia chiave-valore al file di configurazione `appsettings.json` allo stesso livello di origini, sink e pipe:

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

Ad esempio, il seguente file di configurazione configura un'origine, un sink e una pipe e disabilita anche la 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"
    }
}
```

Quando la telemetria è abilitata raccogliamo i seguenti parametri:

`ClientId`

L'ID univoco assegnato automaticamente quando il software è installato.

`ClientTimestamp`

La data e l'ora in cui la telemetria viene raccolta.

`OSDescription`

Una descrizione del sistema operativo.

`DotnetFramework`

L'attuale versione del framework dotnet.

`MemoryUsage`

La quantità di memoria che consuma Kinesis Agent per Windows (MB).

`CPUUsage`

La percentuale Kinesis della CPU di in decimali. Ad esempio, 0,01 equivale a 1%.

`InstanceId`

ID istanza Amazon EC2 se Kinesis Agent per Windows è in esecuzione su un'istanza Amazon EC2.

`InstanceType (string)`

Tipo di istanza Amazon EC2 se Kinesis Agent per Windows è in esecuzione su un'istanza Amazon EC2.

Inoltre, raccogliamo i parametri elencati in Elenco delle metriche di Kinesis Agent per Windows.

Tutorial: Trasmetti file di registro JSON ad Amazon S3 utilizzando Kinesis Agent per Windows

In questo tutorial vengono illustrati i passaggi dettagliati per la configurazione di una pipeline di dati utilizzando Amazon Kinesis Agent per Microsoft Windows (Kinesis Agent per Windows).

Il tutorial include i seguenti passaggi:

- Utilizzo di Kinesis Agent per Windows per eseguire lo streaming di file di log in formato JSON suAmazon Simple Storage Service (Amazon S3)TramiteAmazon Kinesis Data Firehose: . Per informazioni su Kinesis Agent per Windows, vedereCos'è Amazon Kinesis Agent per Microsoft Windows?: .
- Migliorare i dati di log prima che lo streaming utilizzi la decorazione degli oggetti. Per ulteriori informazioni, consulta Configurazione delle decorazioni sink.
- Utilizzo diAmazon AthenaPer cercare determinati tipi di record di log.

Prerequisites

Se non disponi già di un account AWS, creane le istruzioni inConfigurazione di un account AWSper ottenerne uno.

Argomenti

- Fase 1: Configurare Servizi AWS
- Fase 2: Installare, configurare ed eseguire Kinesis Agent per Windows
- Fase 3: Eseguire una query sui dati di log in Amazon S3
- Fasi successive

Fase 1: Configurare Servizi AWS

Per preparare l'ambiente allo streaming dei dati di log ad Amazon Simple Storage Service (Amazon S3) utilizzando Amazon Kinesis Agent per Microsoft Windows, segui questa procedura. Per ulteriori informazioni e prerequisiti, consulta Tutorial: Streaming di file di log JSON su Amazon S3.

Utilizza AWS Management Console per configurare AWS Identity and Access Management (IAM), Amazon S3, Kinesis Data Firehose e Amazon Elastic Compute Cloud (Amazon EC2) per preparare lo streaming dei dati di log da un'istanza EC2 ad Amazon S3.

Argomenti

Configurare policy e ruoli IAM

Creare la seguente policy, che autorizza Kinesis Agent per Windows a eseguire lo streaming dei record a un determinato flusso di distribuzione 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"
        }
    ]
}
```

Replace (Sostituisci)*region*Con il nome della regione AWS in cui verrà creato il flusso di distribuzione Kinesis Data Firehose (`us-east-1`, per esempio). Sostituire *account-id* con l'ID dell'account di 12 cifre per l'account AWS in cui verrà creato il flusso di distribuzione.

Nella barra di navigazione, scegliereSupporto, e quindiCentro di supporto: . Il tuo numero di conto (ID) a 12 cifre attualmente connesso viene visualizzato nellaCentro di supportoRiquadro di navigazione.

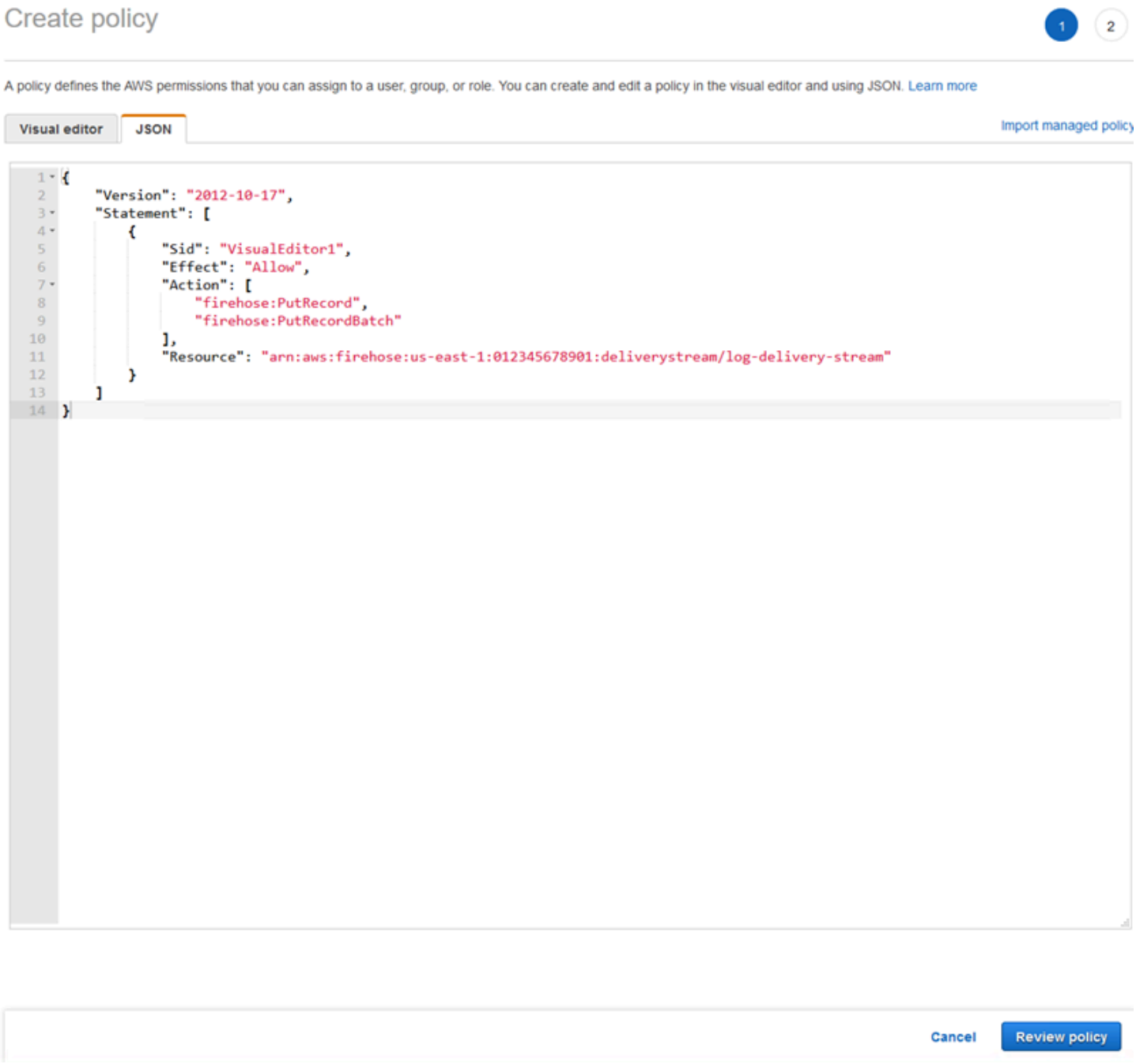
Usare la procedura seguente per creare una policy. Assegnare un nome alla policy `log-delivery-stream-access-policy`.

Utilizzare l'editor della policy JSON per creare una policy

1. Accedere alla Console di gestione AWS e aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, selezionare Policies (Policy).

   Se è la prima volta che si seleziona Policies (Policy), verrà visualizzata la pagina Welcome to Managed Policies (Benvenuto nelle policy gestite). Selezionare Get Started (Inizia).
3. Nella parte superiore della pagina scegliere Create policy (Crea policy).
4. Seleziona la scheda JSON.
5. Specificare un documento della policy JSON. Per informazioni sul linguaggio delle policy IAM, consultaRiferimento alla policy JSON IAM di riferimentonellaGuida per l'utente di IAM: .
6. Al termine, selezionare Review policy (Rivedi policy). In Policy Validator (Validatore di policy) vengono segnalati eventuali errori di sintassi.

   > **Note**
   >
   > È possibile passare tra le schede Visual editor (Editor visivo) e JSON in qualsiasi momento. Tuttavia, se si apportano modifiche o si sceglieEsamina policynellaVisual editor (Editor visivo), IAM potrebbe modificare la policy per ottimizzarla per l'editor visivo. Per ulteriori informazioni, consultaModifica della struttura delle policynellaGuida per l'utente di IAM: .
7. Nella pagina Review policy (Rivedi policy) digitare i valori per Name (Nome) e Description (Descrizione) (facoltativa) per la policy che si sta creando. Consultare il Summary (Riepilogo) della policy per visualizzare le autorizzazioni concesse dalla policy. Selezionare Create policy (Crea policy) per salvare il proprio lavoro.



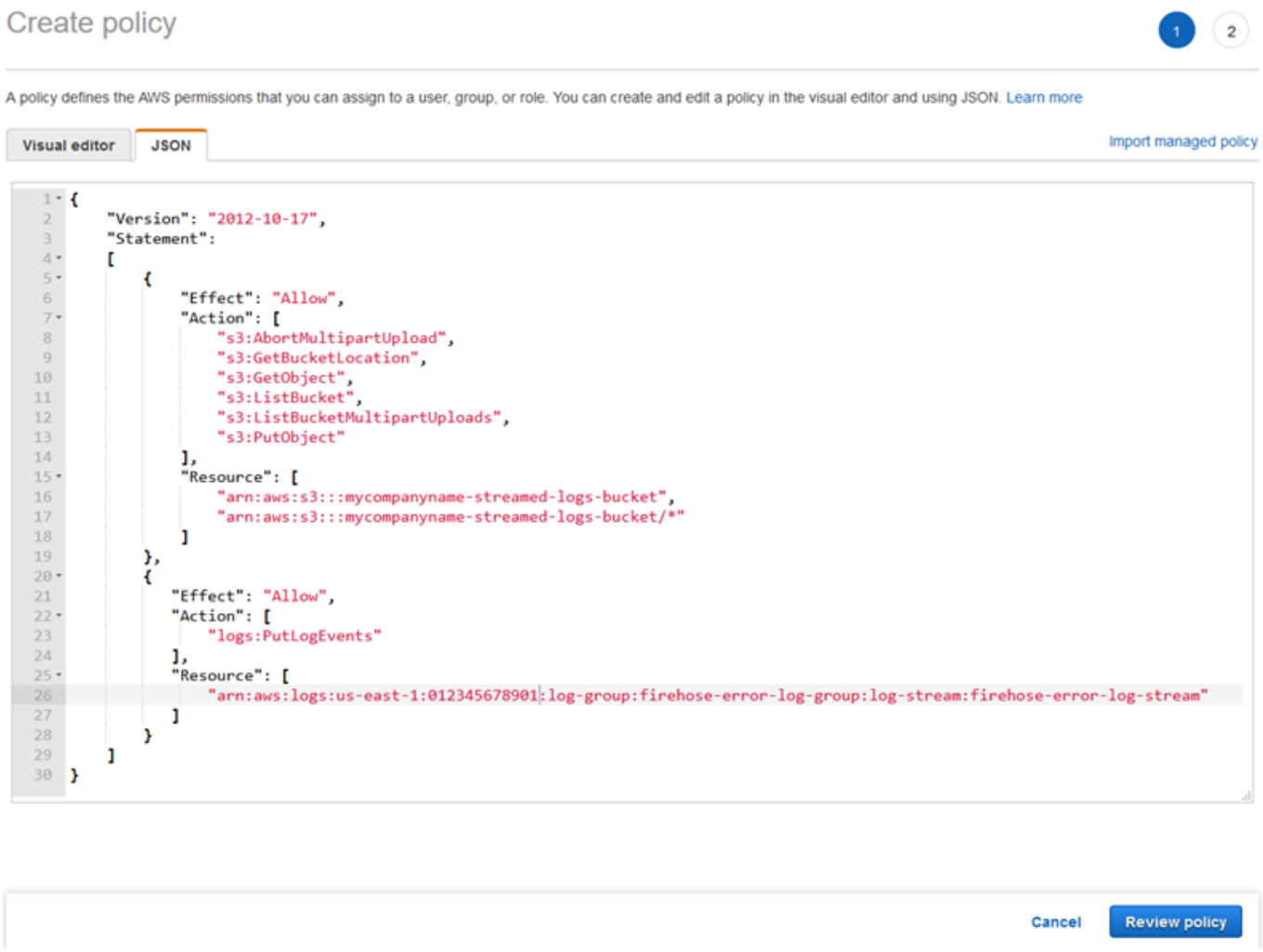
Per creare il ruolo che offre a Kinesis Data Firehose l'accesso a un bucket S3

1. Utilizzando la procedura precedente, creare una policy denominata `firehose-s3-access-policy` definita utilizzando il seguente 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"
           ]
        }
    ]
}
```

Sostituire *bucket-name* con un nome di bucket univoco in cui saranno archiviati i log. Replace (Sostituisci)*region*Con la Regione AWS in cui verranno creati il gruppo di log e il flusso di log CloudWatch Logs (Log) Questi log sono per gli eventuali errori che si verificano durante lo streaming dei dati ad Amazon S3 tramite Kinesis Data Firehose. Sostituire *account-id* con l'ID dell'account di 12 cifre per l'account in cui verranno creati il flusso di log e il gruppo di log.



2. Nel riquadro di navigazione della console IAM, scegliere Roles (Ruoli) e quindi Create role (Crea ruolo).
3. SelezionaServizio AWSTipo di ruolo, quindi scegliere laKinesisServizio.
4. ScegliereKinesis Data FirehosePer il caso d'uso, quindi scegliereSuccessivo: Autorizzazioni.
5. Nella casella di ricerca immetti**`firehose-s3-access-policy`**, scegliere tale policy e quindi sceglieresuccessivo: Review (Revisione): .
6. Nella casella Role name (Nome ruolo), immettere **`firehose-s3-access-role`**.
7. Seleziona Create role (Crea ruolo).

Per creare il ruolo da associare al profilo dell'istanza per l'istanza EC2 che eseguirà Kinesis Agent per Windows

1. Nel riquadro di navigazione della console IAM, scegliere Roles (Ruoli) e quindi Create role (Crea ruolo).

2. SelezionaServizio AWSTipo di ruolo, quindi scegliereEC2: .
3. ScegliereSuccessivo: Autorizzazioni.
4. Nella casella di ricerca immetti **`log-delivery-stream-access-policy`**.
5. Selezionare la policy (Policy), quindi sceglieSuccessivo: Review (Revisione): .
6. Nella casella Role name (Nome ruolo), immettere **`kinesis-agent-instance-role`**.
7. Seleziona Create role (Crea ruolo).

Crea il bucket Amazon S3

Creare il bucket S3 in cui Kinesis Data Firehose esegue lo streaming dei log.

Per creare il bucket S3 per lo storage dei log

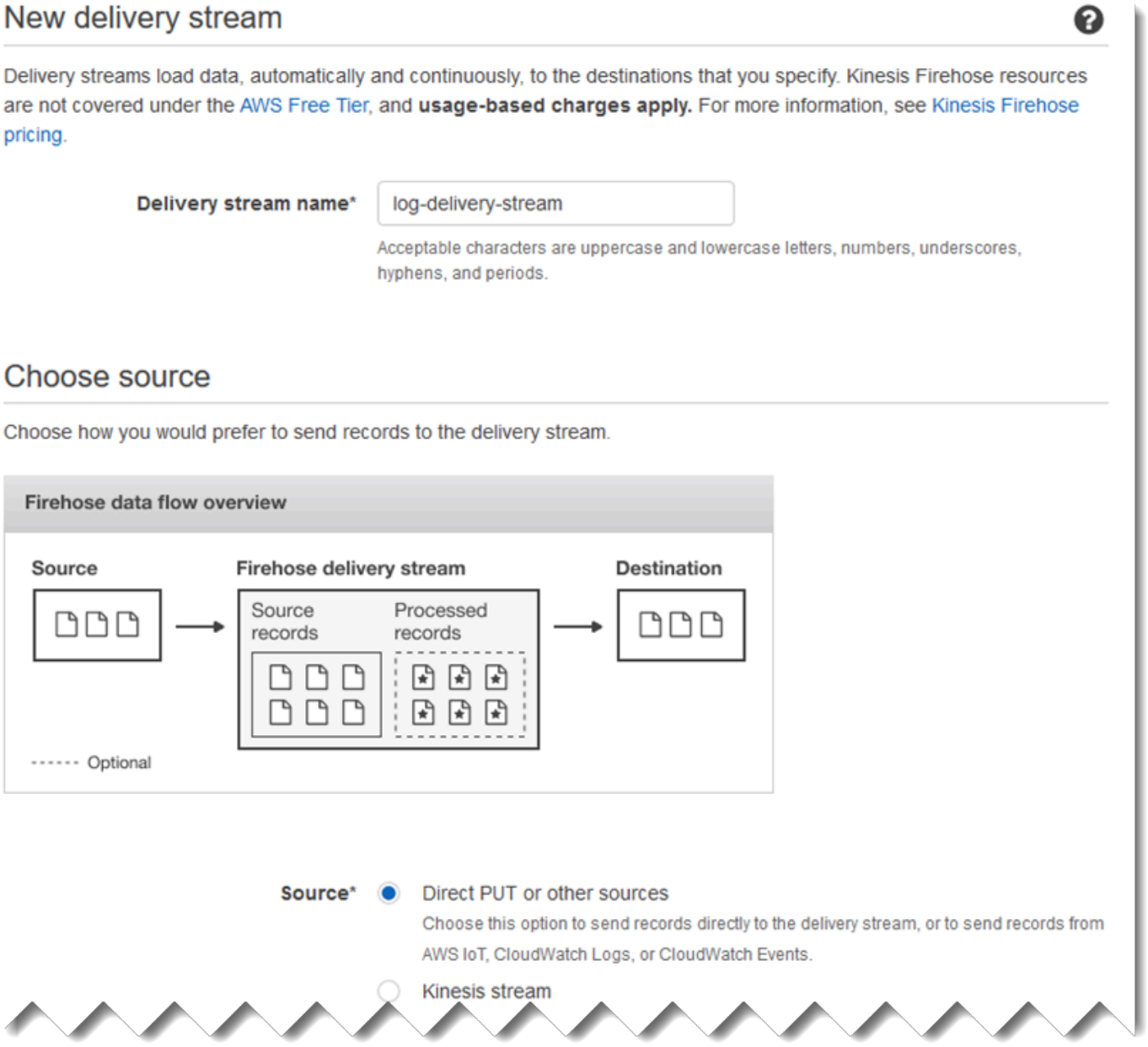
1. Apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Scegliere Create bucket (Crea bucket).
3. In Bucket name (Nome bucket) immettere il nome del bucket S3 univoco scelto in Configurare policy e ruoli IAM.
4. Scegliere la regione in cui il bucket deve essere creato. In genere si tratta della stessa regione in cui si desidera creare il flusso di distribuzione Kinesis Data Firehose e l'istanza Amazon EC2.
5. Scegliere Create (Crea).

Crea il flusso di distribuzione Kinesis Data Firehose

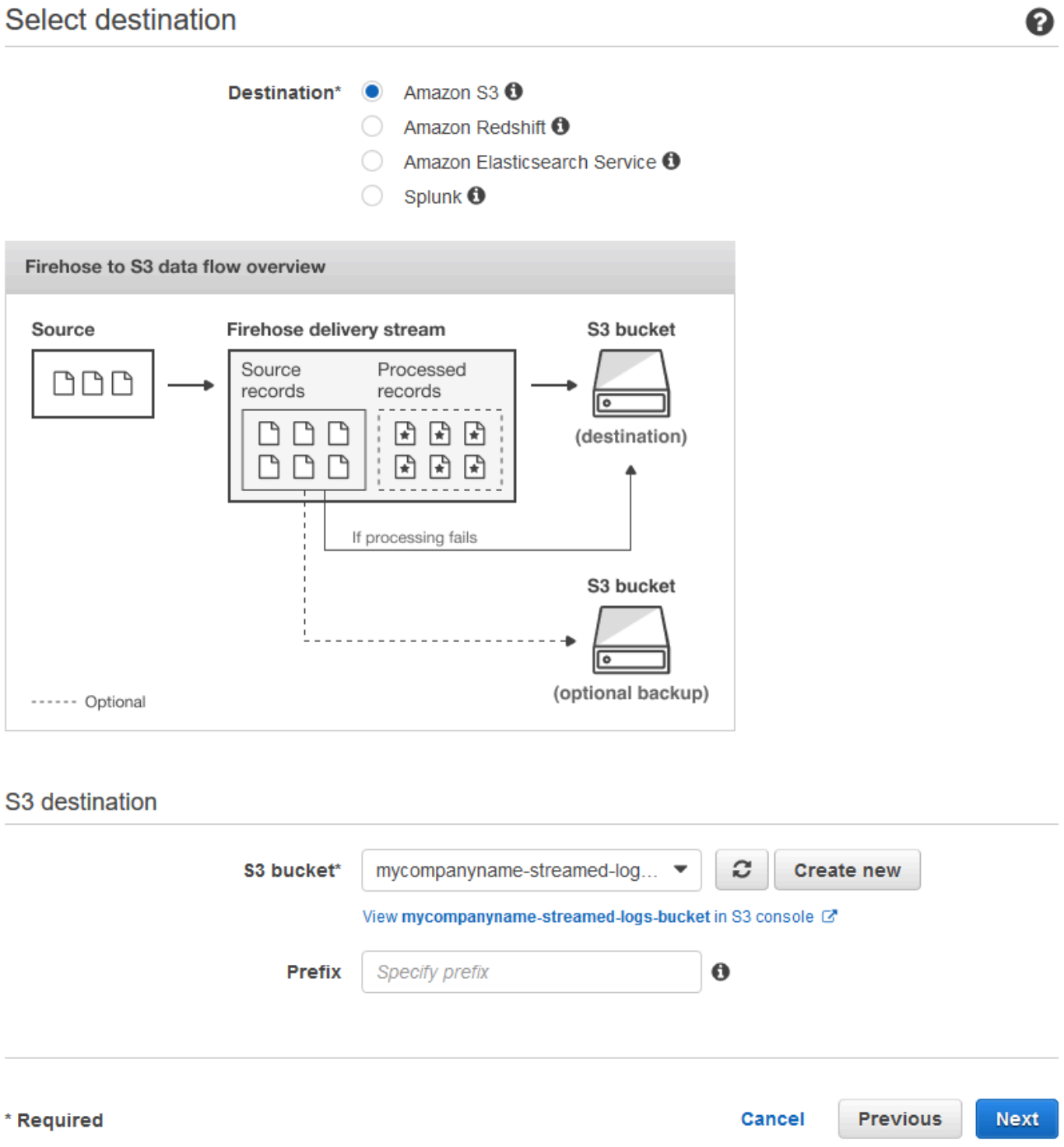
Crea il flusso di distribuzione Kinesis Data Firehose che può archiviare i record dello streaming in Amazon S3.

Per creare il flusso di distribuzione Kinesis Data Firehose

1. Aprire la console Kinesis Data Firehose all'indirizzohttps://console.aws.amazon.com/firehose/: .
2. Scegli Create Delivery Stream (Crea flusso di consegna).
3. Nella casella Delivery stream name (Nome del flusso di distribuzione), immettere **`log-delivery-stream`**.
4. In Source (Sorgente), scegliere Direct PUT o other sources (Direct PUT o altre sorgenti).



5. Seleziona Successivo.
6. Scegliere Next (Successivo) di nuovo.
7. Per la destinazione scegliereAmazon S3: .
8. Per il S3 bucket (Bucket S3), scegliere il nome del bucket creato in Crea il bucket Amazon S3.



9. Seleziona Successivo.
10. Nella casella Buffer interval (Intervallo buffer) immettere **`60`**.
11. In IAM role (Ruolo IAM) selezionare Create new or choose (Crea nuovo o scegli).
12. In IAM role (Ruolo IAM) scegliere `firehose-s3-access-role`.

13. Scegli Allow (Permetti).

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. Seleziona Successivo.
15. Selezionare Create delivery stream (Crea flusso di distribuzione).

Creare l'istanza Amazon EC2 per eseguire Kinesis Agent per Windows

Creare l'istanza EC2 che utilizza Kinesis Agent per Windows per eseguire lo streaming dei record di log tramite Kinesis Data Firehose.

Per creare un'istanza EC2

1. Aprire la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Seguire le istruzioni in Nozioni di base sulle istanze di Amazon EC2 Windows, utilizzando i seguenti passaggi aggiuntivi:
   - Per il IAM role (Ruolo IAM) per l'istanza, scegliere `kinesis-agent-instance-role`.
   - Se non si dispone già di un VPC pubblico connesso a Internet, seguire le istruzioni contenute inConfigurare con Amazon EC2nellaGuida per l'utente di Amazon EC2 User Guide per le: .
   - Creare o utilizzare un gruppo di sicurezza che limita l'accesso all'istanza solo dal computer dell'utente oppure solo dai computer dell'organizzazione. Per ulteriori informazioni, consultaConfigurare con Amazon EC2nellaGuida per l'utente di Amazon EC2 User Guide per le: .
   - Se si specifica una coppia di chiavi esistente, verificare di avere l'accesso alla chiave privata per la coppia di chiavi. In alternativa, creare una nuova coppia di chiavi e salvare la chiave privata in un luogo sicuro.
   - Prima di continuare, attendere finché l'istanza è in esecuzione e ha completato tutti e due i controlli dello stato.
   - L'istanza richiede un indirizzo IP pubblico. Se non è stato allocato un indirizzo, seguire le istruzioni inIndirizzi IP elasticinellaGuida per l'utente di Amazon EC2 User Guide per le: .

Fasi successive

Fase 2: Installare, configurare ed eseguire Kinesis Agent per Windows

Fase 2: Installare, configurare ed eseguire Kinesis Agent per Windows

In questo passaggio si utilizza la Console di gestione AWS per connettersi in remoto all'istanza che viene avviata inCreare l'istanza Amazon EC2 per eseguire Kinesis Agent per Windows: . È quindi possibile installare Amazon Kinesis Agent per Microsoft Windows sull'istanza, creare e distribuire il file di configurazione per Kinesis Agent per Windows e avviare laAwskineSapServizio.

1. Connettersi all'istanza da remoto tramite Remore Desktop Protocol (RDP) seguendo le istruzioni contenute inFase 2: Connessione all'istanzanellaGuida per l'utente di Amazon EC2 User Guide per le: .
2. Nell'istanza, utilizzare Windows Server Manager per disabilitare la configurazione di sicurezza avanzata di Microsoft Internet Explorer per gli utenti e gli amministratori. Per ulteriori informazioni, vedere Procedura per disabilitare la configurazione di sicurezza avanzata di Internet Explorer nel sito Web Microsoft TechNet.
3. Nell'istanza, installare e configurare Kinesis Agent per Windows. Per ulteriori informazioni, consulta Installazione di Kinesis Agent per Windows.
4. Nell'istanza, utilizzare Notepad per creare un file di configurazione Kinesis Agent per Windows. Salvare il file in `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`. Aggiungere i seguenti contenuti al file di configurazione:

```
{
  "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"
    }
  ]
}
```

Questo file configura Kinesis Agent per Windows per l'invio dei record di log in formato JSON dai file nella`c:\logsource\`(la directorysource (origine)) a un flusso di distribuzione Kinesis Data Firehose denominato`log-delivery-stream`(ilsink). Prima che ogni record di log venga trasmesso in streaming a Kinesis Data Firehose, viene migliorato con due coppie chiave-valore aggiuntive che contengono il nome del computer e un timestamp.

5. Creare la directory `c:\LogSource\` e usare Notepad per creare un file `test.log` nella directory con il seguente contenuto:

```
{ "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. In una sessione PowerShell elevata, utilizzare il comando seguente per avviare il servizio AWSKinesisTap:

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

7. Utilizzando Esplora file, andare alla directory `%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`. Aprire il file di log più recente. Il file di log si presenta in maniera simile a quanto riportato di seguito:

```
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.
```

Questo file di log indica che il servizio è stato avviato e i record di log vengono ora raccolti dalla directory `c:\LogSource\`. Ogni linea viene analizzata come un singolo oggetto JSON. Le coppie chiave-valore per il nome del computer e il timestamp vengono aggiunte per ogni oggetto. Quindi viene trasmesso in streaming a Kinesis Data Firehose.

8. In uno o due minuti, passare al bucket Amazon S3 creato inCrea il bucket Amazon S3Utilizzando la Console di gestione AWS. Assicurarsi di aver scelto la Regione corretta nella console.

   In quel bucket, c'è una cartella per l'anno corrente. Aprire tale cartella per visualizzare una cartella per il mese corrente. Aprire tale cartella per visualizzare una cartella per il giorno corrente. Aprire tale cartella per visualizzare una cartella per l'ora corrente (in UTC). Aprire la cartella per visualizzare uno o più elementi che iniziano con il nome `log-delivery-stream`.

9. Aprire il contenuto dell'elemento più recente per confermare che i record di log sono stati archiviati in Amazon S3 con i miglioramenti desiderati. Se tutto è configurato correttamente, i contenuti appaiono simili a quanto segue:

```
{"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. Per informazioni sulla risoluzione di una delle problematiche seguenti, consulta Risoluzione dei problemi di Amazon Kinesis Agent per Microsoft Windows:

    - Il file di log Kinesis Agent per Windows contiene errori.
    - Le cartelle o gli elementi attesi in Amazon S3 non esistono.
    - I contenuti di un articolo Amazon S3 non sono corretti.

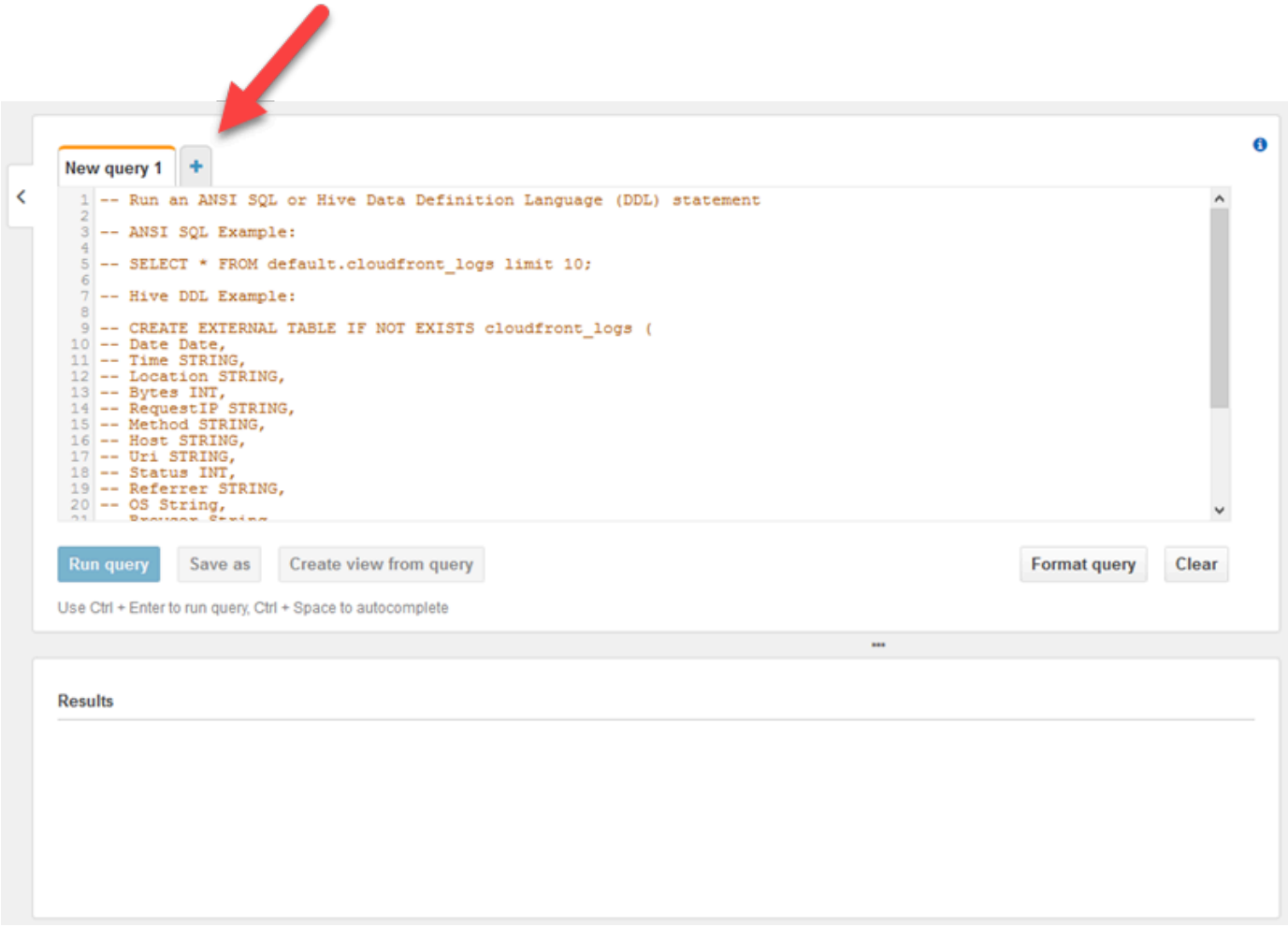
Fasi successive

Fase 3: Eseguire una query sui dati di log in Amazon S3

Fase 3: Eseguire una query sui dati di log in Amazon S3

Nella fase finale di questo Amazon Kinesis Agent per Microsoft WindowsTutorial suÈ possibile utilizzare Amazon Athena per eseguire una query sui dati di log archiviati in Amazon Simple Storage Service (Amazon S3).

1. Aprire la console Athena all'indirizzo https://console.aws.amazon.com/athena/.
2. Seleziona il segno più (+) nella finestra delle query Athena (Query Athena) per creare una nuova finestra di query.



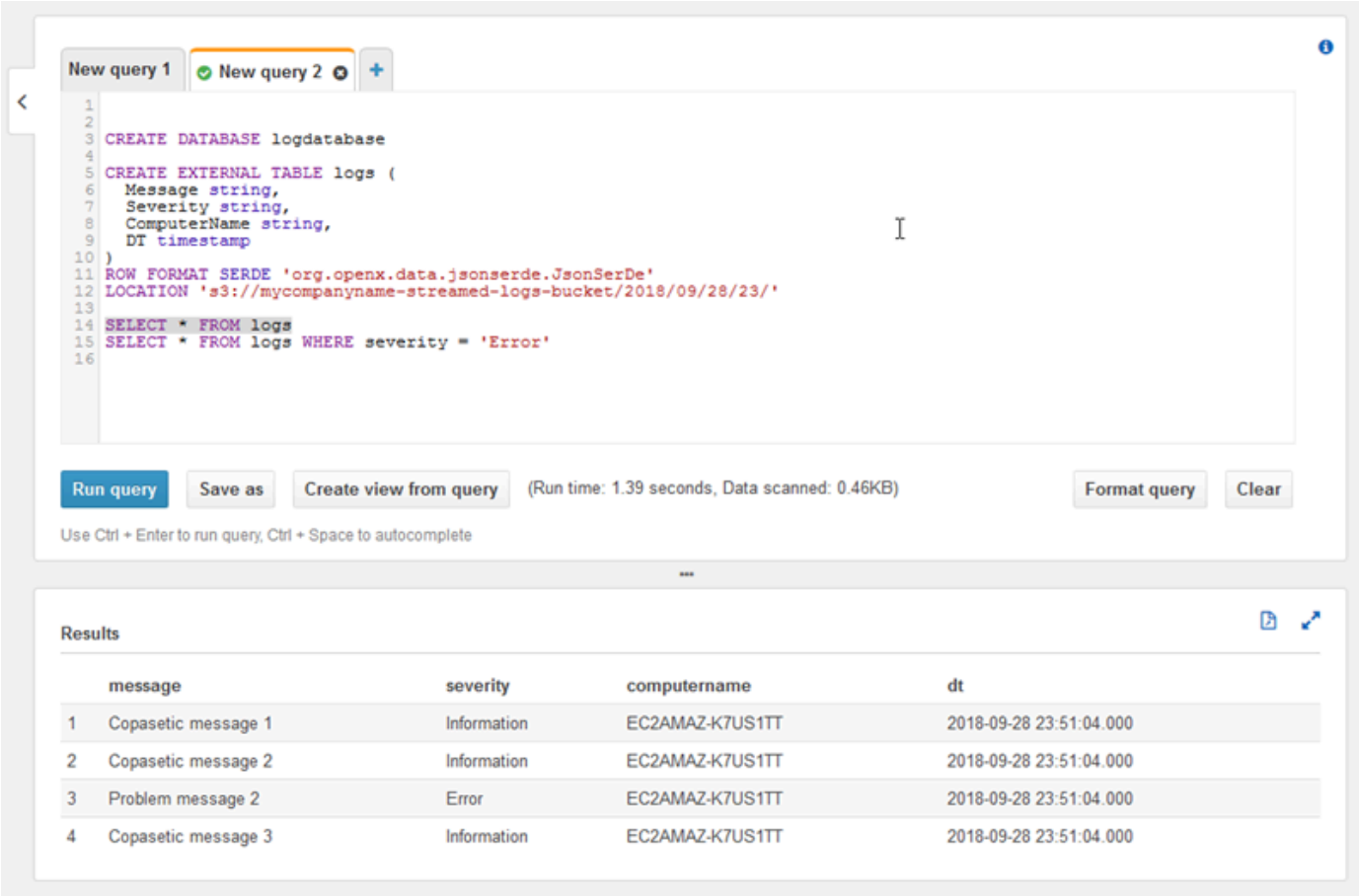
3. Immettere il testo seguente nella finestra di query:

```
CREATE DATABASE logdatabase

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

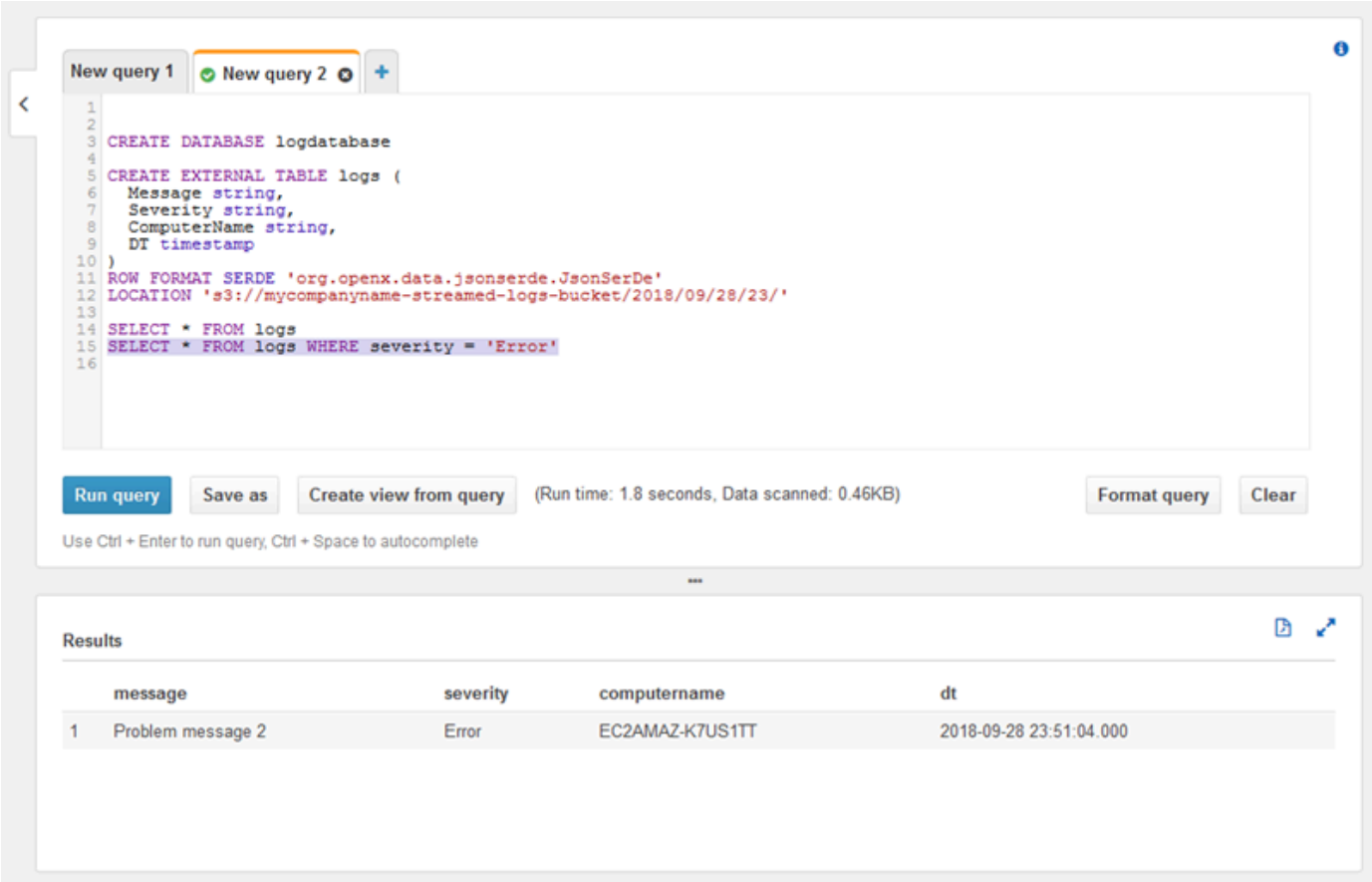
```
  ComputerName string,
  DT timestamp
)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://bucket/year/month/day/hour/'

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

Sostituire *bucket* con il nome del bucket creato in Crea il bucket Amazon S3. Replace (Sostituisci)*year*,*month*,*day*e*hour*Con l'anno, il mese, il giorno e l'ora in cui il file di log Amazon S3 è stato creato in UTC.

4. Selezionare il testo per l'istruzione `CREATE DATABASE`, quindi scegliere Run query (Esegui query). Viene creato il database dei log in Athena.
5. Selezionare il testo per l'istruzione `CREATE EXTERNAL TABLE`, quindi scegliere Run query (Esegui query). In questo modo viene creata una tabella Athena che fa riferimento al bucket S3 con i dati di log e la mappatura dello schema per JSON allo schema per la tabella Athena.
6. Selezionare il testo per la prima istruzione `SELECT`, quindi scegliere Run query (Esegui query). Questo mostra tutte le righe nella tabella.



7. Selezionare il testo per la seconda istruzione `SELECT`, quindi scegliere Run query (Esegui query). In questo modo vengono visualizzate le righe della tabella che rappresentano i record di log con una gravità di livello `Error`. Questo tipo di query trova record di log interessanti da un set di record di log potenzialmente di grandi dimensioni.



Fasi successive

Utilizzare la Console di gestione AWS per eliminare le risorse create durante l'esercitazione:

1. Terminare l'istanza EC2 (vedi il passaggio 3 in Nozioni di base sulle istanze Amazon EC2 di Windows).

   > **Important**
   >
   > Se è stata avviata un'istanza che non era all'interno delPiano gratuito di AWS, verrà addebitato il costo dell'istanza finché non verrà terminata.

2. Eliminare il flusso di distribuzione Kinesis Data Firehose.

   a. Aprire la console Kinesis Data Firehose all'indirizzohttps://console.aws.amazon.com/firehose/: .

   b. Scegliere il flusso di distribuzione creato.

c. Scegli Elimina.

d. Selezionare Delete delivery stream (Elimina flusso di distribuzione).

3. Eliminare il bucket S3. Per istruzioni, consultaCome eliminare un bucket S3?nellaGuida per l'utente di Amazon Simple Storage Service: .

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Configurazione di Amazon Kinesis Agent per Microsoft Windows
- Cos'è Amazon Kinesis Data Firehose?
- Che cos'è Amazon S3?
- Che cos'è Amazon Athena?

Risoluzione dei problemi di Amazon Kinesis Agent per Microsoft Windows

Utilizza le seguenti istruzioni per diagnosticare e risolvere i problemi quando si usa Amazon Kinesis Agent per Microsoft Windows.

Argomenti

Nessun dato viene inviato in streaming da desktop o server ai servizi AWS attesi

Symptoms

Quando si esaminano i log, gli eventi e i parametri ospitati da vari servizi AWS che vengono configurati per ricevere flussi di dati da Kinesis Agent per Windows, nessun dato viene trasmesso mediante Kinesis Agent per Windows.

Causes

Le cause per questo problema sono diverse:

- Un'origine, un sink o una pipe non sono stati configurati correttamente.
- L'autenticazione per Kinesis Agent per Windows non è stata impostata correttamente.
- L'autorizzazione per Kinesis Agent per Windows non è stata impostata correttamente.
- Si è verificato un errore in un'espressione regolare fornita in una dichiarazione `DirectorySource`.
- Una directory inesistente viene specificata per una dichiarazione `DirectorySource`.

- I valori non validi vengono forniti per i servizi AWS, che quindi respingono le richieste da Kinesis Agent per Windows.
- Un sink è un riferimento a una risorsa che non esiste nella regione AWS specificata o implicita.
- Una query non valida viene specificata per una dichiarazione `WindowsEventLogSource`.
- È stato specificato un valore non valido per la coppia chiave-valore `InitialPosition` per un'origine.
- Il file di configurazione `appsettings.json` non è conforme allo schema JSON per quel file.
- I dati vengono trasmessi in streaming in una regione diversa rispetto a quello selezionata nella console di gestione AWS.
- Kinesis Agent per Windows non è installato correttamente o non è in esecuzione.

Resolutions

Per risolvere i problemi con i dati non in streaming, eseguire i seguenti passaggi:

1. Esaminare i registri di Kinesis Agent per Windows nel`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`Directory. Cercare la stringa `ERROR`.
   a. Se non è stato possibile caricare un'origine o un sink, procedere nel seguente modo:
      i. Esaminare il messaggio di errore e individuare il `Id` del sink o dell'origine.
      ii. Controllare la dichiarazione dell'origine o del sink che corrisponde al `Id` nel file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` per verificare la presenza di eventuali errori correlati al messaggio di errore trovato. Per ulteriori dettagli, consulta Configurazione di Amazon Kinesis Agent per Microsoft Windows.
      iii. Correggere eventuali problematiche relative all'errore del file di configurazione.
      iv. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di configurazione siano stati risolti.
   b. Se il messaggio di errore indica che una `SourceRef` o un `SinkRef` non è stato trovato per una pipe, eseguire le operazioni descritte di seguito:
      i. Prendere nota della pipe `Id`.
      ii. Esaminare la dichiarazione della pipe nel file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` che corrisponde all'`Id` annotato. Verificare che i valori delle coppie chiave-valore `SourceRef` e `SinkRef` abbiano `Id` scritti correttamente per le dichiarazioni di origine e sink a cui si intende fare riferimento. Correggere gli eventuali errori di battitura o di ortografia. Se una dichiarazione di sink oppure origine manca dal

file di configurazione, aggiungere la dichiarazione. Per ulteriori informazioni, consulta Configurazione di Amazon Kinesis Agent per Microsoft Windows.

iii. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di configurazione siano stati risolti.

c. Se il messaggio di errore indica che un determinato utente o ruolo IAM non è autorizzato a eseguire determinate operazioni, procedere nel seguente modo:

i. Verificare che l'utente o il ruolo IAM corretti vengano utilizzati da Kinesis Agent per Windows. Se non lo è, consultaConfigurazione di sicurezza del sinke regolare come autentica Kinesis Agent per Windows per garantire che venga utilizzato l'utente o il ruolo IAM corretto.

ii. Se viene utilizzato l'utente o il ruolo IAM corretto, utilizzando AWS Management Console, esaminare le policy associate all'utente o al ruolo. Verificare che l'utente o il ruolo dispongano di tutte le autorizzazioni menzionate nel messaggio di errore per tutte le risorse AWS a cui Kinesis Agent per Windows accede. Per ulteriori informazioni, consulta Configurazione dell'autorizzazione.

iii. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di sicurezza siano stati risolti.

d. Se il messaggio di errore indica che si è verificato un errore di argomento durante l'analisi di un'espressione regolare contenuta nel file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`, eseguire le operazioni descritte di seguito:

i. Esaminare l'espressione regolare nel file di configurazione.

ii. Verificare la sintassi dell'espressione regolare. Vi sono diversi siti Web che è possibile utilizzare per verificare le espressioni regolari, oppure utilizzare le seguenti righe di comando per verificare la presenza di espressioni regolari per una dichiarazione di origine `DirectorySource`:

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

Sostituisci *`sourceId`* con il valore della coppia chiave-valore `Id` della dichiarazione di origine `DirectorySource` con un'errata espressione regolare.

iii. Apportare le correzioni necessarie all'espressione regolare nel file di configurazione in modo che sia valido.

iv. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di configurazione siano stati risolti.

e. Se il messaggio di errore indica che si è verificato un errore di argomento durante l'analisi di un'espressione regolare che non è contenuta nel file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` ed è relativa a un sink specifico, eseguire le operazioni descritte di seguito:

   i. Individuare la dichiarazione di sink nel file di configurazione.

   ii. Verificare che le coppie chiave-valore relative a un servizio AWS stanno utilizzando nomi che rispettino le regole di convalida per quel servizio. Ad esempio, i nomi dei gruppi CloudWatch Logs devono contenere solo un determinato set di caratteri specificato utilizzando l'espressione regolare.`[\.\-_/#A-Za-z0-9]+`: .

   iii. Correggere gli eventuali nomi non validi nelle coppie chiave-valore per la dichiarazione di sink e verificare che tali risorse siano configurate correttamente in AWS.

   iv. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di configurazione siano stati risolti.

f. Se il messaggio di errore indica che un'origine o un sink non è in grado di caricare a causa di un parametro nullo o mancante, procedere nel seguente modo:

   i. Prendere nota del `Id` dell'origine o del sink.

   ii. Individuare la dichiarazione dell'origine o del sink corrispondente all'`Id` specificato nel file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

   iii. Esaminare le coppie chiave-valore che vengono fornite nella dichiarazione di origine o del sink confrontata con i requisiti del tipo di origine o di sink nella documentazione Configurazione di Amazon Kinesis Agent per Microsoft Windows per il tipo di sink pertinente. Aggiungere qualsiasi coppia chiave-valore obbligatoria mancante alla dichiarazione dell'origine o del sink.

   iv. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di configurazione siano stati risolti.

g. Se il messaggio di errore indica che un nome di directory non è valido, eseguire le operazioni descritte di seguito:

   i. Individuare il nome della directory non valido nel file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

   ii. Verificare che questa directory esista già e contenga i file di log che devono essere oggetto dello streaming.

   iii. Correggere gli eventuali refusi o errori nel nome della directory specificato nel file di configurazione.

iv. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di configurazione siano stati risolti.

h. Se il messaggio di errore indica che una risorsa non esiste:

i. Individuare il riferimento per la risorsa che non esiste in una dichiarazione di sink nel file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

ii. Utilizzare AWS Management Console per individuare la risorsa nella regione AWS corretta che dovrebbe essere utilizzata nella dichiarazione del sink. Confrontarla con ciò che è stato specificato nel file di configurazione.

iii. Cambiare la dichiarazione del sink nel file di configurazione per avere il nome della risorsa e della regione corretti.

iv. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di configurazione siano stati risolti.

i. Se il messaggio di errore indica che una query non è valida per un particolare `WindowsEventLogSource`, eseguire le operazioni descritte di seguito:

i. Nel file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap \appsettings.json`, individuare la dichiarazione `WindowsEventLogSource` con lo stesso `Id` del messaggio di errore.

ii. Verificare che il valore della coppia chiave-valore `Query` nella dichiarazione di origine soddisfi le query di eventi e gli XML di eventi.

iii. Apportare le modifiche alla query per garantire la conformità.

iv. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di configurazione siano stati risolti.

j. Se il messaggio di errore indica che una posizione iniziale non è valida, eseguire le operazioni descritte di seguito:

i. Nel file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap \appsettings.json`, individuare la dichiarazione di origine con lo stesso `Id` del messaggio di errore.

ii. Cambiare il valore della coppia chiave-valore `InitialPosition` nella dichiarazione di origine per essere conformi ai valori consentiti, come descritto in Configurazione del segnalibro.

iii. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di configurazione siano stati risolti.

2. Assicurarsi che il file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json` sia conforme allo schema JSON.

   a. In una finestra di prompt dei comandi, invocare i comandi seguenti:

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

   b. Correggere eventuali problemi rilevati con il file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap\appsettings.json`.

   c. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più attuale per verificare che i problemi di configurazione siano stati risolti.

3. Cambiare il livello di log per cercare di ottenere dati sui log più dettagliati.

   a. Sostituire il file di configurazione `%PROGRAMFILES%\Amazon\AWSKinesisTap\nlog.xml` con il contenuto seguente:

   ```
   <?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. Avviare e arrestare il servizio `AWSKinesisTap`. Quindi verificare il file di log più recente per vedere se ci sono messaggi aggiuntivi nel log che potrebbero contribuire a diagnosticare e risolvere il problema.

4. Verificare che si stanno esaminando le risorse nella regione corretta nella console di gestione AWS.
5. Verificare che l'agente Kinesis per Windows sia installato e in esecuzione.
   a. In Windows, scegliere Start (Avvia), quindi passare a Control Panel (Pannello di controllo), Administrative Tools (Strumenti di amministrazione), Services (Servizi).
   b. Trovare il servizio AWSKinesisTap.
   c. Se il servizio AWSKinesisTap non è visibile, installare Kinesis Agent per Windows seguendo le istruzioni inNozioni di base su Amazon Kinesis Agent per Microsoft Windows: .
   d. Se il servizio è visibile, stabilire se il servizio è in esecuzione. Se non è in esecuzione, aprire il menu contestuale (clic con il pulsante destro del mouse) per il servizio e scegliere Start (Avvia).
   e. Verificare che il servizio sia stato avviato esaminando il file di log più recente nella directory `%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`.

Si applica a

Questa informazione si applica a Kinesis Agent per Windows versione 1.0.0.0.115 e versione successiva.

I dati attesi sono a volte mancanti

Symptoms

Nella maggior parte dei casi, Kinesis Agent per Windows esegue lo streaming dei dati correttamente, ma che occasionalmente alcuni dati sono mancanti.

Causes

Le cause per questo problema sono diverse:

- La funzione di creazione di un segnalibro non viene utilizzata.
- I limiti di velocità dei dati per i servizi AWS vengono superati in base alla configurazione corrente di tali servizi.
- I limiti di velocità per le chiamate API per i servizi AWS vengono superati in base a`appsettings.json`e i limiti dell'account AWS.

Resolutions

Per risolvere i problemi con i dati mancanti, eseguire i seguenti passaggi:

1. Valutare la possibilità di utilizzare la funzione di creazione di un segnalibro documentata in Configurazione del segnalibro. Contribuisce a garantire che tutti i dati vengono infine inviati, anche quando Kinesis Agent per Windows viene interrotto e avviato.
2. Utilizza le metriche integrate di Kinesis Agent per Windows per individuare i problemi:
   a. Abilitare lo streaming delle metriche di Kinesis Agent per Windows come descritto inConfigurazione dell'agente Kinesis per le pipe metriche di Windows: .
   b. Se c'è un notevole numero di errori non recuperabili per uno o più sink, stabilire il numero di byte o record inviati al secondo. Quindi specificare se questo è entro i limiti configurati per quei servizi AWS nella regione e nell'account verso cui i dati vengono trasmessi.
   c. Quando i limiti vengono superati, o si riduce la velocità o la quantità di dati inviati, si richiede un aumento dei limiti oppure si aumenta lo sharding, se applicabile.
   d. Dopo aver apportato gli adeguamenti, continuare a monitorare i parametri integrati di Kinesis Agent per Windows per assicurare che il problema sia stato risolto.

Per ulteriori informazioni sui limiti di Kinesis Data Streams, consultaLimiti di Kinesis Data Streamsnella Kinesis Data Streams: . Per ulteriori informazioni sui limiti di Kinesis Data Firehose, consultaLimiti di Amazon Kinesis Data Firehose: .

Si applica a

Questa informazione si applica a Kinesis Agent per Windows versione 1.0.0.0.115 e versione successiva.

I dati arrivano in un formato non corretto

Symptoms

I dati arrivano al servizio AWS in un formato errato.

Causes

Le cause per questo problema sono diverse:

- Il valore della coppia chiave-valore `Format` per una dichiarazione del sink nel file di configurazione `appsettings.json` non è corretto.
- Il valore per la coppia chiave-valore `RecordParser` in una dichiarazione `DirectorySource` non è corretto.
- Le espressioni regolari in una dichiarazione `DirectorySource` che usa il parser di record `Regex` non sono corrette.

Resolutions

Per risolvere i problemi con una formattazione non corretta, eseguire i seguenti passaggi:

1. Rivedere le dichiarazioni di sink nel file di configurazione `%PROGRAMFILES%\Amazon \AWSKinesisTap\appsettings.json`.
2. Verificare che il valore corretto della coppia chiave-valore `Format` venga specificato per ogni dichiarazione sink. Per ulteriori informazioni, consulta Dichiarazioni dei sink.
3. Se le origini con dichiarazioni `DirectorySource` sono collegate da pipe a sink che specificano i valori `xml` o `json` per la coppia chiave-valore `Format`, assicurarsi che tali sorgenti specifichino uno dei seguenti valori per la coppia chiave-valore `RecordParser`:

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

Altri parser di record sono basati solo su testo e non funzionano correttamente con sink che richiedono la formattazione JSON o XML.

4. Se i record di log non vengono analizzati correttamente dal tipo di origine `DirectorySource`, invocare le seguenti righe in una finestra del prompt dei comandi per verificare il timestamp e le coppie chiave-valore di espressioni regolari specificate nella dichiarazione `DirectorySource`:

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

Sostituisci il *`sourceID`* con il valore della coppia chiave-valore `Id` della dichiarazione di origine `DirectorySource` che sembra non funzionare correttamente. Correggere gli eventuali problemi rilevati da `ktdiag.exe`.

Si applica a

Questa informazione si applica a Kinesis Agent per Windows versione 1.0.0.0.115 e versione successiva.

Problematiche di prestazioni

Symptoms

Le applicazioni e i servizi hanno aumentato le latenze dopo l'installazione e l'avvio di Kinesis Agent per Windows.

Causes

Le cause per questo problema sono diverse:

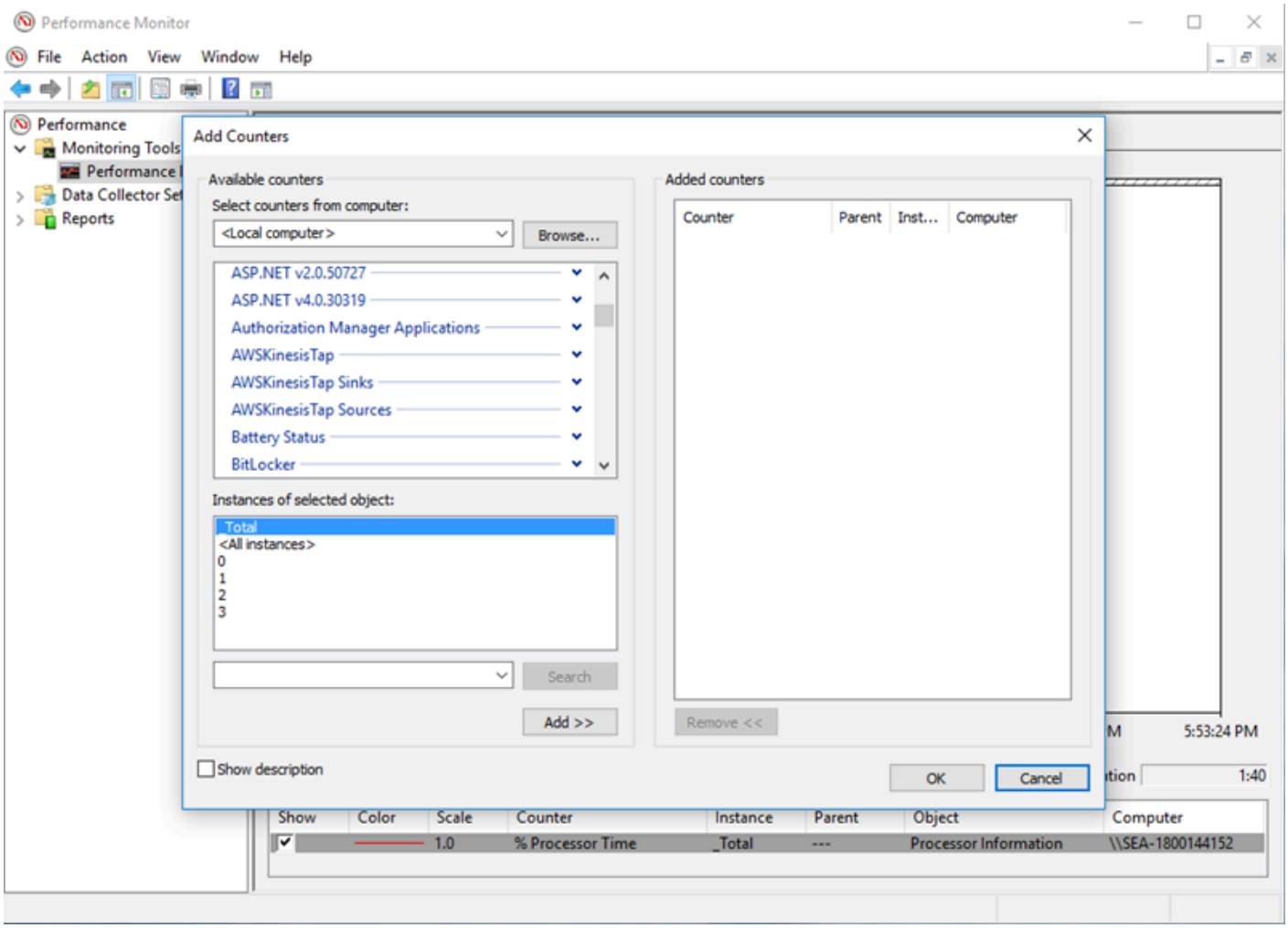
- Il computer in cui viene eseguito Kinesis Agent per Windows non dispone di capacità sufficiente per eseguire lo streaming della quantità di dati desiderata.

- I dati non necessari vengono trasmessi a uno o più servizi AWS.
- Kinesis Agent per Windows sta eseguendo lo streaming ai servizi AWS che non sono configurati per una velocità dei dati così elevata.
- Kinesis Agent per Windows sta chiamando le operazioni sui servizi AWS in un account in cui il limite di velocità delle chiamate API è troppo basso.

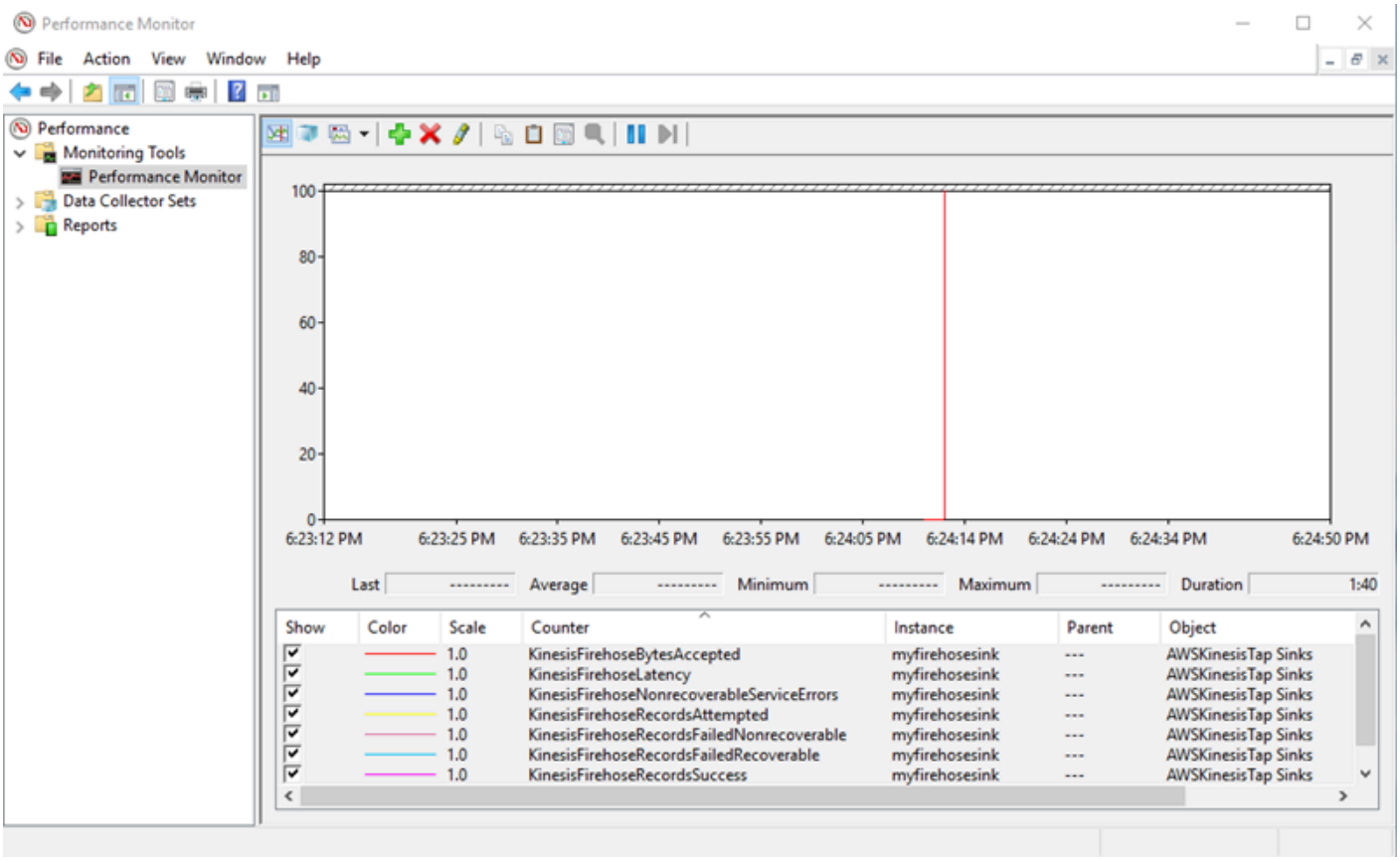
Resolutions

Per risolvere i problemi relativi alle prestazioni, eseguire i seguenti passaggi:

1. Utilizzare l'applicazione di monitoraggio delle risorse di Windows per verificare la memoria, la CPU, il disco e l'utilizzo della rete. Se devi trasmettere grandi quantità di dati con Kinesis Agent per Windows, potrebbe essere necessario effettuare il provisioning di una macchina con capacità superiori in alcune di queste aree, a seconda della configurazione.
2. È possibile ridurre la quantità di dati registrati utilizzando il filtraggio:
   - Consulta la coppia chiave-valore `Query` Configurazione WindowsEventLogSource.
   - Vedi il filtraggio delle pipeline in Configurazione di pipe.
   - Consulta il filtro dei parametri di Amazon CloudWatch inConfigurazione sink CloudWatch).
3. Utilizzare l'applicazione di monitoraggio delle prestazioni di Windows per visualizzare i parametri di Kinesis Agent per Windows o eseguire lo streaming di tali parametri a CloudWatch (vediOrigine dei parametri integrati di Kinesis Agent per Windows). Nell'applicazione di monitoraggio delle prestazioni di Windows, è possibile aggiungere contatori per sink e origini Kinesis Agent per Windows. Sono elencati sotto le categorie di contatori AWSKinesisTap Sinks e AWSKinesisTap Sources.



Ad esempio, per diagnosticare i problemi di prestazioni di Kinesis Data Firehose, aggiungere ilKinesis Firehose LavelloContatori di prestazioni.



In caso di un numero elevato di errori recuperabili, controllare i log di Kinesis Agent per Windows più recenti nella sezione`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`Directory. Se per i sink `KinesisStream` o `KinesisFirehose` si verifica il throttling, procedere come segue:

- Se si verifica il throttling a causa di uno streaming dei dati troppo veloce, prendere in considerazione la possibilità di aumentare il numero di shard per il flusso di dati Kinesis. Per ulteriori informazioni, consultaResharding, dimensionamento ed elaborazione parallelanellaKinesis Data Streams: .
- Prendere in considerazione la possibilità di aumentare il limite delle chiamate API per Kinesis Data Streams o di aumentare le dimensioni del buffer per il sink se le chiamate API sono oggetto di throttled. Per ulteriori informazioni, consultaLimiti di Kinesis Data StreamsnellaKinesis Data Streams: .
- Se lo streaming dei dati viene eseguito troppo rapidamente, è possibile richiedere un aumento del limite concesso per il flusso di distribuzione Kinesis Data Firehose. Oppure, se le chiamate API sono oggetto di throttling, richiedere un aumento del limite delle chiamate API (consulta Limiti di Amazon Kinesis Data Firehose) oppure aumentare le dimensioni del buffer per il sink.
- Dopo l'aumento del numero di shard per un Kinesis Data Streams o l'aumento del limite di velocità per un flusso di distribuzione Kinesis Data Firehose, modificare l'agente Kinesis per

Windows`appsettings.json`File di configurazione per aumentare i record al secondo o i byte al secondo per il sink. In caso contrario, Kinesis Agent per Windows non è in grado di sfruttare i limiti aumentati.

Si applica a

Questa informazione si applica a Kinesis Agent per Windows versione 1.0.0.0.115 e versione successiva.

Spazio su disco esaurito

Symptoms

Kinesis Agent per Windows è in esecuzione su una macchina che ha poco spazio su disco a disposizione in uno o più dischi rigidi.

Causes

Le cause per questo problema sono diverse:

- Il file di configurazione di registrazione di Kinesis Agent per Windows non è corretto.
- La coda persistente di Kinesis Agent per Windows non è stata impostata correttamente.
- Alcune altri applicazione o servizi stanno consumando spazio su disco.

Resolutions

Per risolvere i problemi relativi allo spazio su disco, eseguire i seguenti passaggi:

- Se lo spazio sul disco che contiene i file di log di Kinesis Agent per Windows è ridotto, esaminare la directory dei file di log (in genere`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`). Verificare che un numero ragionevole di file di log venga conservato e che i file di log siano di dimensioni ragionevoli. È possibile controllare il percorso, la conservazione e il livello di dettaglio dei log di Kinesis Agent per Windows modificando il`%PROGRAMFILES%\Amazon\AWSKinesisTap \Nlog.xml`File di configurazione.
- Quando la funzione di accodamento sink è abilitata, esaminare le dichiarazioni dei sink che usano tale funzionalità. Verificare che la coppia chiave-valore `QueuePath` faccia riferimento a una unità

disco con spazio sufficiente per contenere il numero massimo di batch specificato utilizzando la coppia chiave-valore `QueueMaxBatches`. Se questo non è possibile, ridurre il valore della coppia chiave-valore `QueueMaxBatches`, in modo che i dati entrino facilmente nel rimanente spazio su disco per l'unità disco specificato.

- Utilizzare Esplora file di Windows per individuare i file che consumano lo spazio su disco e trasferire o eliminare i file in eccesso. Modificare la configurazione dell'applicazione o del servizio che consumano grandi quantità di spazio su disco.

Si applica a

Questa informazione si applica a Kinesis Agent per Windows versione 1.0.0.0.115 e versione successiva.

Strumenti per la risoluzione dei problemi

Oltre che per verificare il file di configurazione, è possibile utilizzare il`ktdiag.exe`, che offre diverse altre caratteristiche per diagnosticare e risolvere i problemi durante la configurazione e l'utilizzo di Kinesis Agent per Windows. Lo strumento `ktdiag.exe` si trova nella directory `%PROGRAMFILES%\Amazon\AWSKinesisTap`.

- Se si ritiene che i file di log con un determinato modello di file vengono scritti in una directory, ma non sono in corso di elaborazione da parte di Kinesis Agent per Windows, utilizzare la`/w`per verificare che queste modifiche vengano rilevate. Ad esempio, supponiamo che i file di log con il modello di nome di file `*.log` vengano scritti nella directory `c:\foo`. È possibile utilizzare l'opzione `/w` quando si esegue lo strumento `ktdiag.exe`, specificando la directory e il modello del file:

  ```
  cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
  ktdiag /w c:\foo *.log
  ```

  Se vengono scritti i file di log, è possibile visualizzare risultati simili ai seguenti:

  ```
  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 non si verifica l'output, quindi c'è un problema a livello di applicazione o di servizio nello scrivere i log, oppure si è verificato un problema di configurazione di sicurezza piuttosto che un problema con Kinesis Agent per Windows. Se ad esempio si verifica l'output, ma Kinesis Agent per Windows non sta ancora apparentemente elaborando i log, consultaNessun dato viene inviato in streaming da desktop o server ai servizi AWS attesi: .

- A volte i log vengono scritti solo occasionalmente, ma potrebbe essere utile verificare che Kinesis Agent per Windows funzioni correttamente. Utilizzare l'opzione `/log4net`per simulare i log di scrittura di un'applicazione utilizzando la libreria `Log4net`, ad esempio:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
KTDiag.exe /log4net c:\foo\log2.log
```

Questo scrive un file di log di stile `Log4net` per il file di log `c:\foo\log2.log` e mantiene l'aggiunta di nuove voci di log finché il tasto viene selezionato. È possibile configurare diverse opzioni utilizzando opzioni aggiuntive che sono facoltativamente specificate dopo il nome del file:

Blocco: `-lm`, `-li` o `-le`

È possibile specificare una delle seguenti opzioni di blocco che controllano il modo in cui il file di log è bloccato:

`-lm`

La quantità minima di blocco viene utilizzata nel file di log, consentendo l'accesso massimo al file di log.

`-li`

Solo i thread all'interno dello stesso processo possono accedere al log nello stesso momento.

`-le`

Solo un thread alla volta può accedere al log. Questa è l'impostazione predefinita.

`-tn:`*`millisecondi`*

Specifica il numero di *`millisecondi`* tra la scrittura delle voci di registro. Il valore predefinito è 1000 millisecondi (1 secondo).

`-sm:`*`byte`*

Specificare il numero di *`byte`* per ogni voce di log. Il valore predefinito è 1000 byte.

`-bk:`*`numero`*

Specifica il *`numero`* di voci di registro da scrivere alla volta. Il valore predefinito è 1.

- Talvolta è utile simulare un'applicazione che scrive nel registro eventi di Windows. Utilizzare l'opzione `/e` per scrivere voci di registro di un evento di registro di Windows, ad esempio:

```
cd /D %PROGRAMFILES%\Amazon\AWSKinesisTap
KTDiag.exe /e Application
```

Questo scrive voci di registro al log di eventi di Windows Application fino a quando un tasto è premuto. È anche possibile specificare le seguenti opzioni aggiuntive dopo il nome del registro:

`-tn:`*`millisecondi`*

Specifica il numero di *`millisecondi`* tra la scrittura delle voci di registro. Il valore predefinito è 1000 millisecondi (1 secondo).

`-sm:`*`byte`*

Specificare il numero di *`byte`* per ogni voce di log. Il valore predefinito è 1000 byte.

`-bk:`*`numero`*

Specifica il *`numero`* di voci di registro da scrivere alla volta. Il valore predefinito è 1.

Creazione di Kinesis Agent per i plugin di Windows

Per la maggior parte delle situazioni, la creazione di un plugin Amazon Kinesis Agent per Microsoft Windows non è necessaria. Kinesis Agent per Windows è altamente configurabile e contiene origini e sink potenti, come`DirectorySource`e`KinesisStream`, che sono sufficienti per la maggior parte degli scenari. Per ulteriori informazioni sulle origini e i sink esistenti, consulta Configurazione di Amazon Kinesis Agent per Microsoft Windows.

Per gli scenari insoliti, potrebbe essere necessario estendere Kinesis Agent per Windows utilizzando un plug-in personalizzato. Di seguito sono riportati alcuni scenari di esempio:

- La creazione di un pacchetto di una dichiarazione `DirectorySource` complessa utilizzando i parser di record `Regex` o `Delimited` in modo che sia facile da applicare in diversi tipi di file di configurazione.
- Creazione di una nuova origine che non si basa su file o che supera la funzionalità di analisi fornita dal parser di record esistente.
- Creazione di un sink per un servizio AWS che non è attualmente supportato.

Argomenti

Introduzione ai plugin Kinesis Agent per Windows

Non c'è nulla di speciale sui plug-in personalizzati. Tutte le origini e i sink esistenti utilizzano gli stessi meccanismi che i plug-in personalizzati utilizzano per caricare quando viene avviato e creano un'istanza di plug-in rilevanti dopo la lettura del`appsettings.json`File di configurazione.

Quando si avvia Kinesis Agent per Windows, si verifica la sequenza seguente:

1. Kinesis Agent per Windows analizza i gruppi nella directory di installazione (`%PROGRAMFILES%\Amazon\AWSKinesisTap`) per le classi che implementano il`IFactory<T>`definita nella`Amazon.KinesisTap.Core`Assembly. Questa interfaccia è definita

in`Amazon.KinesisTap.Core\Infrastructure\IFactory.cs`nel codice sorgente Kinesis Agent per Windows.

2. Kinesis Agent per Windows carica gli assembly contenenti queste classi e richiama il`RegisterFactory`su queste classi.
3. Kinesis Agent per Windows carica il`appsettings.json`File di configurazione. Per ogni origine e sink nel file di configurazione, vengono esaminate le coppie chiave-valore `SourceType` e `SinkType`. Se non ci sono factory registrate con lo stesso nome dei valori delle coppie chiave-valore `SourceType` e `SinkType`, il metodo `CreateInstance` viene richiamato su tali factory. Il metodo `CreateInstance` viene passato alla configurazione e ad altre informazioni come un oggetto `IPluginContext`. Il metodo `CreateInstance` è responsabile della configurazione e inizializzazione del plug-in.

Affinché un plug-in funzioni correttamente, ci deve essere una factory registrata che crea il plug-in e deve essere definita la classe di plug-in.

Il codice sorgente di Kinesis Agent per Windows si trova inhttps://github.com/awslabs/kinesis-agent-windows: .

Implementazione dell'agente Kinesis per le fabbriche di plugin Windows

Segui questi passaggi per implementare una factory di plug-in di Kinesis per Windows.

Per creare una fabbrica di plugin Kinesis Agent per Windows

1. Creazione di un progetto di libreria C# indirizzato a .NET Framework 4.6.
2. Aggiungi un riferimento all'assembly `Amazon.KinesisTap.Core`. Questo assembly si trova nella`%PROGRAMFILES%\Amazon\AWSKinesisTap`dopo l'installazione di Kinesis Agent per Windows.
3. Utilizzare `NuGet` per installare il pacchetto `Microsoft.Extensions.Configuration.Abstractions`.
4. Utilizzare `NuGet` per installare il pacchetto `System.Reactive`.
5. Utilizzare `NuGet` per installare il pacchetto `Microsoft.Extensions.Logging`.

6. Creare una classe factory che implementa sia `IFactory<IEventSource>` per origini o `IFactory<IEventSink>` per sink. Aggiungere i metodi `RegisterFactory` e `CreateInstance`.

   Ad esempio, il codice seguente crea una factory di plug-in di Kinesis Agent per Windows che crea un'origine che genera dati casuali:

```
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);
            }
        }
    }
}
```

La dichiarazione `switch` viene utilizzata nel metodo `CreateInstance` nel caso in cui si desidera migliorare la factory per creare diversi tipi di istanze.

Per creare una factory di sink che crea un sink che non fa nulla, utilizzare una classe simile alla seguente:

```
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}.");
            }
        }
    }
}
```

Implementazione di Kinesis Agent per origini plugin Windows

Segui questi passaggi per implementare un'origine di plug-in Kinesis per Windows.

Per creare un plugin Kinesis Agent per Windows

1. Aggiungi una classe che implementa l'interfaccia `IEventSource<out T>` al progetto creato in precedenza per l'origine.

   Ad esempio, utilizzare il codice seguente per definire una sorgente che genera dati casuali:

```
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();
        }
    }
}
```

In questo esempio, la classe `RandomSource` eredita dalla classe `EventSource<T>` perché fornisce la proprietà `Id`. Anche se questo esempio non supporta i segnalibri, questa classe di base è utile anche per l'implementazione di tale funzionalità. Le envelope forniscono un modo per memorizzare i metadati e avvolgere i dati arbitrari per lo streaming ai sink. La classe `RandomData` viene definita nella fase successiva e rappresenta il tipo di oggetto di output da questa origine.

2. Aggiungi una classe al progetto definito in precedenza che contiene i dati oggetto di streaming dall'origine.

Ad esempio, un contenitore di dati casuali potrebbe essere definito come segue:

```
namespace MyCompany.MySources
{
    public class RandomData
    {
        public int RandomValue { get; set; }
    }
}
```

3. Compila il progetto definito in precedenza.
4. Copia l'assembly nella directory di installazione per Kinesis Agent per Windows.
5. Crea o aggiorna un`appsettings.json`che utilizza la nuova origine e inserirlo nella directory di installazione per Kinesis Agent per Windows.
6. Arresto e avvio di Kinesis Agent per Windows.
7. Controllare il file di registro corrente di Kinesis Agent per Windows (in genere si trova nella`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`) per garantire che non ci siano problemi con il plug-in sorgente personalizzato.
8. Verificare che i dati siano pervenuti al servizio AWS desiderato.

Per un esempio di come estendere il`DirectorySource`Per implementare l'analisi di un determinato formato di log, consulta`Amazon.KinesisTap.Uls\UlsSourceFactory.cs`e`Amazon.KinesisTap.Uls\UlsLogParser.cs`nel codice sorgente Kinesis Agent per Windows.

Per un esempio di come creare un'origine che fornisce la funzionalità di segnalibro, consulta`Amazon.KinesisTap.Windows\WindowsSourceFactory.cs`e`Amazon.KinesisTap.Windows\EventLogSource.cs`nel codice sorgente Kinesis Agent per Windows.

Implementazione di Kinesis Agent per i sink plugin di Windows

Segui questi passaggi per implementare il sink di plug-in Kinesis per Windows.

Per creare un sink di plugin Kinesis Agent per Windows

1. Aggiungi una classe al progetto definito in precedenza che implementa l'interfaccia `IEventSink`.

   Ad esempio, il codice seguente implementa un sink che non fa altro che registrare l'arrivo dei record, in seguito rimossi.

```
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.");
        }
    }
}
```

   In questo esempio, la classe del sink `NullSink` eredita dalla classe `EventSink` perché offre la possibilità di trasformare i record in diversi formati di serializzazione, ad esempio JSON e XML.

2. Compila il progetto definito in precedenza.
3. Copia l'assembly nella directory di installazione per Kinesis Agent per Windows.

4. Crea o aggiorna un`appsettings.json`che utilizza il nuovo sink e inserirlo nella directory di installazione per Kinesis Agent per Windows. Ad esempio, per utilizzare i plug-in personalizzati `RandomSource` e `NullSink`, è possibile utilizzare il seguente file di configurazione `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"
 }
  ]
}
```

   Questa configurazione crea un'origine che invia un'istanza di `RandomData` con un set `RandomValue` a un numero casuale compreso tra 0 e 50 ogni 10 secondi. Crea un sink che trasforma le istanze `RandomData` in entrata per JSON, registra quel JSON, quindi elimina le istanze. Assicurati di includere entrambe le factory di esempio, la classe di origine `RandomSource` e la classe del sink `NullSink` nel progetto definito in precedenza per l'utilizzo di file di configurazione di esempio.

5. Arresto e avvio di Kinesis Agent per Windows.

6. Controllare il file di registro corrente di Kinesis Agent per Windows (in genere si trova nella`%PROGRAMDATA%\Amazon\AWSKinesisTap\logs`) per garantire che non ci siano problemi con il plug-in sink personalizzato.

7. Verificare che i dati siano pervenuti al servizio AWS desiderato. Poiché l'esempio `NullSink` non viene trasmesso a un servizio AWS, è possibile verificare il corretto funzionamento del sink ricercando i messaggi di log che indicano che i record sono stati ricevuti.

   Ad esempio, il file di log dovrebbe essere simile a quanto segue:

```
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 si crea un sink che accede ai servizi AWS, vi sono classi di base che possono risultare utili. Per un sink che utilizza il`AWSBufferedEventSink`classe base, vedere`Amazon.KinesisTap.AWS \CloudWatchLogsSink.cs`nel codice sorgente di Kinesis Agent per Windows.

Cronologia documenti per Amazon Kinesis Agent per Microsoft Windows Guida per l'utente

Versione API: 15/10/2018

Nella tabella seguente vengono descritte le modifiche apportate alleGuida per l'utente di Amazon Kinesis Agent per Microsoft Windows(presente documento).

update-history-change	update-history-description	update-history-date
Aggiornamento della documentazione principale	Sono state aggiunte istruzioni per l'installazione di MSI. Configurazione Directory Source aggiornata e aggiunta WindowsEventLogPollingSource. Per la configurazione del sink, è stata aggiunta la configurazione di sincronizzazione del file system locale; ProfileRefreshingAWSCredentialProvider; informazioni sulle decorazioni di testo, risoluzione delle variabili negli attributi sink, configurazione degli endpoint regionali STS per i sink, configurazione degli endpoint VPC e configurazione di server proxy alternativi. Per le pipe, attributi di configurazione aggiunti.	23 febbraio 2021
Aggiornamento alla documentazione	Argomento aggiornato per comunicare che le specifiche delle posizioni di Amazon S3 sono sensibili alle maiuscole.	7 Novembre 2018

Rilascio iniziale, versione 1.0.115	Prima versione della Guida dell'utente di Kinesis Agent per Windows.	5 Novembre 2018

Glossario AWS

Per la terminologia AWS più recente, consulta il glossario AWS in Riferimenti generali AWS.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.